2022

Beyond Statistical Fairness

Christopher Sangyeon Jung
University of Pennsylvania

Beyond Statistical Fairness

Abstract

In recent years, a great deal of fairness notions has been proposed. Yet, most of them take a reductionist approach by indirectly viewing fairness as equalizing some error statistic across pre-defined groups. This thesis aims to explore some ideas as to how to go beyond such statistical fairness frameworks.

First, we consider settings in which the right notion of fairness may not be captured by simple mathematical definitions but might be more complex and nuanced and thus require elicitation from individual or collective stakeholders. By asking stakeholders to make pairwise comparisons to learn which pair of individuals should be treated similarly, we show how to approximately learn the most accurate classifier or converge to such one subject to the elicited fairness constraints. We consider an offline setting where the pairwise comparisons must be made prior to training a model and an online setting where one can continually provide fairness feedback to the deployed model in each round. We also report preliminary findings of a behavioral study of our framework using human-subject fairness constraints elicited on the COMPAS criminal recidivism dataset.

Second, unlike most of the statistical fairness framework that promises fairness for pre-defined and often coarse groups, we provide fairness guarantees for finer subgroups, such as all possible intersections of the pre-defined groups, in the context of uncertainty estimation in both offline and online setting. Our framework gives uncertainty guarantees that are more locally sensible than the ones given by conformal prediction techniques; our uncertainty estimates are valid even when averaged over any subgroup, but uncertainty estimates in conformal predictions are usually only valid when averaged over the entire population.

Degree Type

Dissertation

Degree Name

Doctor of Philosophy (PhD)

Graduate Group

Computer and Information Science

First Advisor

Aaron Roth

Second Advisor

Michael Kearns

Subject Categories

Computer Sciences

BEYOND STATISTICAL FAIRNESS

Christopher Sangyeon Jung

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy

2022

Co-Supervisor of Dissertation

Michael Kearns, Professor and National Center Chair, Computer and Information Science

Co-Supervisor of Dissertation

Aaron Roth, Henry Salvatori Professor of Computer & Cognitive Science, Computer and Information Science

Graduate Group Chairperson

Mayur Naik, Professor, Computer and Information Science,

Dissertation Committee
Edgar Dobriban, Assistant Professor, Statistics and Data Science
Sampath Kannan, Henry Salvatori Professor, Computer and Information Science
Rakesh Vohra, George A. Weiss and Lydia Bravo Weiss University Professor, Computer and Information Science
Steven Wu, Assistant Professor, School of Computer Science at Carnegie Mellon University

To my grandfather, 유 창 상

ACKNOWLEDGEMENTS

First of all, I want to thank my advisors, Aaron Roth and Michael Kearns. It wouldn't be an exaggeration to say that any brilliance you may find in this thesis is a mere reflection of their brilliance. It has been an honor to see firsthand how they formulate research questions that just feel right and offer answers that bring thought-provoking insights to the problems at hand; their art of research that brings order and simplicity to chaos and complexity is awe-inspiring. I feel genuinely so fortunate to have had the chance to learn from them, and I am deeply grateful for their guidance and encouragement over the years. I could not have had better advisors.

I also have other mentors that I must thank. I thank Sampath Kannan for acting as my third unofficial advisor, especially in the beginning of my graduate school years. Sampath brought me to the fairness program at the Simons institute twice where I was able to get my foot in the door of the algorithmic fairness community and make new friends and collaborators. I thank Rakesh Vohra who always willingly made time to discuss projects that I was working on and provided helpful feedback and encouragement: through those discussions, he helped me develop clearer and simpler intuition for my research projects. I thank Rajiv Gandhi for first introducing me to the world of theoretical computer science through PACT (Program in Algorithmic and Combinatorial Thinking) and instilling in me the simple joy of learning. I thank James Davis and David Williamson for working patiently with me when I was just a naive undergraduate student at Cornell. I thank Eva Tardos for encouraging me to go to graduate school. I thank Edgar Doriban for serving on my thesis committee even on such a short notice.

I'm also indebted to my collaborators and friends that I made during graduate school: Matthew Joseph, Hadi Elzayn, Shahin Jabbari, Saeed Sharifi-Malvajerdi, Seth Neel, Zachary

Schutzman, Marcella Hastings, Emily Diana, Yahav Bechavod, Neil Lutz, Steven Wu, Changhwa Lee, Mallesh Pai, Katrina Ligett, Logan Stapleton, Chris Waites, Moshe Shenfeld, Varun Gupta, Juba Ziani, Pranjal Awasthi, and Jamie Morgenstern. I have genuinely enjoyed spending time together and have learned so much from them.

I also need to thank my close friends outside of graduate school: Charles Kim, Kevin Jiseup and Jen Kim, David Sang Jun and Sarah Kim, Cornell and Faye Bang, Chanwook and Steph Park, Jon and Jeniffer Kim, Kevin and Heejung Lee, Joseph An, Harold Shin and Christie Kim, Joon Jang, Sam Lee, Andrew Qian and Esther Yoon, Eric and Joslyn Cheung, and Harry Kim. They kept me sane and reminded me countless times that there's more to this life through their rich friendship with me.

I thank Risen Christ Fellowship and Metro Church for faithfully preaching the gospel to me and helping me remember my hope lies not in my own works but in the finished work of Jesus Christ.

Finally, I thank my family. I thank my 엄마 and 아빠 for unconditionally loving me and firmly believing in me. Thank you for always just being there for me. As I am getting older, I am learning more and more how much you have given up for me and how deeply you love me. I also thank my younger brothers, 상은 and 상민. Honestly, you guys were more like my older brothers to me: I'm sure that I have learned important things in life more from you than you guys have from me.

I am grateful that God has blessed me with all these wonderful people in my life — I won't be who I am without them. God has revealed much of His love to me through these people. But more importantly, I am grateful that God has faithfully reminded me of His love in the person of Jesus Christ in the past few years — that He became human to die for my sin so that I may become His righteousness. It was this love that kept me during what felt like a never-ending journey. I can confidently say that this love is far more meaningful and goes much deeper than anything this thesis has to offer.

ABSTRACT

BEYOND STATISTICAL FAIRNESS

Christopher Jung

Michael Kearns

Aaron Roth

In recent years, a great deal of fairness notions has been proposed. Yet, most of them take a reductionist approach by indirectly viewing fairness as equalizing some error statistic across pre-defined groups. This thesis aims to explore some ideas as to how to go beyond such statistical fairness frameworks.

First, we consider settings in which the right notion of fairness may not be captured by simple mathematical definitions but might be more complex and nuanced and thus require elicitation from individual or collective stakeholders. By asking stakeholders to make pairwise comparisons to learn which pair of individuals should be treated similarly, we show how to approximately learn the most accurate classifier or converge to such one subject to the elicited fairness constraints. We consider an offline setting where the pairwise comparisons must be made prior to training a model and an online setting where one can continually provide fairness feedback to the deployed model in each round. We also report preliminary findings of a behavioral study of our framework using human-subject fairness constraints elicited on the COMPAS criminal recidivism dataset.

Second, unlike most of the statistical fairness framework that promises fairness for pre-defined and often coarse groups, we provide fairness guarantees for finer subgroups, such as all possible intersections of the pre-defined groups, in the context of uncertainty estimation in both offline and online setting. Our framework gives uncertainty guarantees that are more locally sensible than the ones given by conformal prediction techniques; our uncertainty estimates are valid even when averaged over any subgroup, but uncertainty estimates in conformal predictions are usually only valid when averaged over the entire population.

Table of Contents

List of Illustrations

Chapter 1

Introduction

As the amount of data being collected has increased exponentially and the speed at which we can sift through such data has sped up dramatically in the past few decades, machine learning now pervades almost every facet of our lives, including consequential decision making processes such as hiring, lending, criminal sentencing, policing, and so on. And as a result of machine learning's growing role in these high-stakes settings, people have started to worry about its potential unfair behaviors — rightfully so. This worry about unfairness in machine learning algorithms is no longer hypothetical, as it has been in fact documented in several real world scenarios that machine learning algorithms even without any explicit nefarious intentions encoded in them can display some unfair behaviors.

So far, most of these complaints about machine learning algorithms' unfair behaviors have been against their discrepancy in some statistical error between groups defined by protected attributes, such as race and gender. Let us take COMPAS, a recidivism risk assessment tool used in many jurisdictions in the U.S., as an example, as it has received one of the first and well known complaints regarding machine learning algorithm's potentially unfair behavior: Angwin et al. [4] argued that COMPAS may be unfair as it has non-trivial differences in its false positive and negative rates between white and black defendants [63].

In an effort to combat against these documented and potential statistical discrepancies, researchers have proposed various fairness notions and how to impose such technical definitions of fairness [2, 3, 16, 19, 41, 50, 56, 59, 73, 92, 99]. However, maybe because many of these notions have been born possibly as an reaction to statistical discrepancies, most of them have the form of equalizing simple error statistics across groups. As a starting point of this thesis, we observe the following:

1. The above process cannot result in notions of fairness that do not have any simple,

analytic description. Moreover, this process overlooks a more precursory problem: namely, *who gets to define what is fair?* There are many statistical measures that one can choose, but they cannot be equalized all simultaneously [13, 63]. So in the statistical fairness framework, one is tied to having to decide only a select few compatible statistical measures and across which groups the measure should be equalized. Yet there haven't been many guidelines in terms of how to reason about this process of choosing which statistical measure, which group, and more importantly who makes this decision.

2. Statistical measures by definition are only meaningful when averaged over a group of points, and the groups for which fairness guarantees are made are often quite coarse — e.g. most of the time, the intersections across multiple sensitive attributes are not considered [59]. Hence, there is usually no fairness guarantee for such finer and intersecting subgroups, let alone individuals.

We divide the thesis into two parts to address the points raised above.

Fairness in the Hands of the People: In the first part, we argue that researchers may not be able to propose a concise technical definition, e.g. statistical parity, to capture the nuances of fairness in any given context. Instead, many philosophers hold that *stakeholders* who are affected by moral decisions and *domain experts* who understand the context in which moral decisions are made will have the best judgment about which decisions are fair in that context [64, 95]; this is also aligned with recent work on *virtual democracy* which propose and enact participatory methods to automate moral decision-making [15, 54, 67, 75]. One non-statistical fairness notion that can possibly take people's conceptions of fairness into account is individual fairness originally proposed by Dwork et al. [19]. Individual fairness asks that similar people should be treated similarly, and one's conception of fairness can be distilled into the fairness metric that defines this similarity between two individuals. Furthermore, unlike statistical fairness guarantees that are aggregate in nature, individual fairness gives guarantees at the individual level.

Despite its stronger fairness semantic and ability to encode one's conceptions of fairness, an entity (e.g. a group of stakeholders and/or a single domain expert) whose conception of fairness we want to instill into the algorithm or want to study may have a hard time enunciating the quantitative fairness metric exactly and/or form a consistent fairness metric. Nevertheless, we hold that one can easily identify specific scenarios where fairness and/or unfairness occurs. Therefore, in this part of the thesis, we investigate how to incorporate individual conceptions of fairness into a system that only receives access to examples where these conceptions are violated and/or met. We broadly consider two settings.

In Chapter 2 and Chapter 3, we consider an online setting where a batch of k individuals shows up in each round for whom we need to make predictions. There is an auditor who has some unknown fairness metric that determines which individuals are similar. But the auditor cannnot directly enunciate this fairness metric, so the auditor contintually engages with the deploy model and complains in each round whenever any pair of similar individuals is not treated similarly. In Chapter 2, we consider when the underlying reward associated with each individual is linear with respect to some unknown vector in expectation, and the unknown fairness metric of the auditor is Mahalanobis. We show how to achieve no-regret with respect to all policies that are fair with respect to the auditor's fairness metric and also guarantee the the number of rounds that the auditor complains about unfair treatments to be sublinear. In Chapter 3, we consider a slightly different setting and loosen some assumptions. We make no structural assumption about the auditor's fairness metric and how the label of each agent is determined. By considering a new loss that combines the classification and fairness loss together, we reduce the problem of achieving no-regret with respect to fair policies and sublinear fairness loss to the standard online classification problem with no fairness constraints.

In Chapter 4, we consider an offline setting where we are endowed with a labeled dataset that consists of agents' features and their labels. Here we aim to elicit stakeholders and experts' conceptions of fairness by asking them to compare pairs of individuals in specific scenarios *prior to* training and deploying a predictor. To each subject from whom we want to elicit fairness, we present randomly drawn pairs of individuals from the dataset and ask whether each pair of individuals should be treated similarly or not. Given the original dataset and elicited pairs of individuals who should be treated similarly according to the subjects, we provide a provably convergent and oracle-efficient algorithm that minimizes the empirical risk over the dataset and satisfies the elicited fairness constraints. Then, by making some modifications to the standard generalization argument via VC-dimension, we show that fairness loss, which measures how well we respect the fairness constraints of the subjects, generalizes. Finally, we report preliminary findings of a behavioral study of our framework using human-subject fairness constraints elicited on the COMPAS criminal recidivism dataset.

Uncertainty Estimation for Subgroups: In the statistical fairness framework, the groups for which the fairness guarantees are made are usually pre-defined according to some sensitive attributes and hence very coarse — e.g. whites and non-whites, males and females, and so on. However, as noted by Kearns et al. [59], such process may be susceptible to inadvertent "fairness gerrymandering" in which the classifier may be still unfair to one or more subgroups defined by the intersections of the original groups (e.g. non-white females). With regards to this, Hébert-Johnson et al. [43] show how to guarantee calibration, a fairness criterion that many have considered [9, 79], against all groups that are computationally identifiable — for instance, all the intersecting subgroups across sensitive attributes as described above, which may be exponential in the number of sensitive attributes.

In the second half of the thesis, we consider how we can extend the idea of "multicalibration" introduced by Hébert-Johnson et al. [43] in an offline and online setting. In chapter 5, we study an offline setting where we show how to multicalibrate not only means as originally done in Hébert-Johnson et al. [43] but also higher moments, such as variance, allowing us to compute uncertainty estimates via Chebyshev's inequality that are calibrated for all subgroups. The uncertainty estimates that we compute provide have more locally sensible guarantees than conformal prediction techniques. More specifically, our uncertainty estimate for each point is valid averaged over randomness of any subgroup that the point may belong to, but conformal prediction techniques' uncertainty estimates almost always average over the randomness over the entire population. In Chapter 6, we show how to compute multicalibrated predictions for means, higher moments, and prediction intervals in an online setting where there is essentially no assumption made regarding the data generation process, thereby removing the exchangeability assumption that is common in the conformal prediction literature.

I

Fairness

In the Hands of the People

“It is an axiom in my mind that our liberty can never be safe but in the hands of the people themselves.”

Thomas Jefferson

Chapter 2

Individual Fairness via Auditing: Mahalanobis Fairness Metric

2.1. Introduction

Most of the work in algorithmic fairness literature has taken the *statistical fairness* approach which first fixes a small collection of high-level groups defined by protected attributes (e.g., race or gender) and then asks for approximate parity of some statistic of the predictor, such as positive classification rate or false positive rate, across these groups (see e.g., [13, 39, 41, 56, 63, 99]). While such notions of group fairness are easy to operationalize, they are aggregate in nature without fairness guarantees for finer subgroups or individuals [19, 43, 59] with no clear guidance on how to choose which statistic and across which groups to equalize the statistic. On the other hand, individual fairness definitions ask for some constraint that binds on the individual level, rather than only over averages of people. Often, these constraints have the semantics that "similar people should be treated similarly [19]. Moreover, one's individual notion of fairness can be distilled into this specific fairness metric that determines who is considered similar for the given context.

Individual fairness definitions indeed have substantially stronger semantics and demands than group definitions of fairness, as Dwork et al. [19] lay out a compendium of ways in which group fairness definitions are unsatisfying. However, the statistical group fairness approach is more prevalent in large part because notions of individual fairness require making stronger assumptions on the setting under consideration. In particular, the definition from Dwork et al. [19] requires that "task-specific fairness metric" is readily available to the algorithmic designer.

Learning problems over individuals are also often implicitly accompanied by some notion of *merit*, embedded in the objective function of the learning problem. For example, in a lending setting we might posit that each loan applicant is either "creditworthy" and will repay a loan, or is not creditworthy and will default — which is what we are trying to predict. Joseph et al. [50] take the approach that this measure of merit — already present in the model, although initially unknown to the learner — can be taken to be the similarity metric in the definition of Dwork et al. [19], requiring informally that creditworthy individuals have at least the same probability of being accepted for loans as defaulting individuals. The implicit and coarse fairness metric here assigns distance zero between pairs of creditworthy individuals and pairs of defaulting individuals, and some non-zero distance between a creditworthy and a defaulting individual. This resolves the problem of how one should discover the "fairness metric" but results in a notion of fairness that is necessarily aligned with the notion of "merit" (creditworthiness) that we are trying to predict.

However, there are many settings in which the notion of merit we wish to predict may be different or even at odds with the notion of fairness people have in their mind. For example, notions of fairness aimed at rectifying societal inequities that result from historical discrimination can aim to favor the disadvantaged population (say, in college admissions), even if the performance of the admitted members of that population can be expected to be lower than that of the advantaged population. Similarly, we may have a fairness notion in mind that try to incorporate only those attributes that individuals can change in principle (and thus excluding ones like race, age and gender) and that further express what are and are not meaningful differences between individuals, outside the context of any particular prediction problem. These kinds of fairness desiderata can still be expressed as an instantiation of the definition from Dwork et al. [19], but with a task-specific fairness metric separate from the notion of merit we are trying to predict.

In this chapter, we revisit the individual fairness definition from Dwork et al. [19]. This definition requires that pairs of individuals who are close in the fairness metric must be treated "similarly" (e.g. in an allocation problem such as lending, served with similar probability).

We investigate the extent to which it is possible to satisfy this fairness constraint while simultaneously solving an online learning problem. Most importantly, one main conceptual problem with metric-based definitions, that we seek to address, is that it may be difficult for anyone to actually precisely express a quantitative metric over individuals — but they nevertheless might "know unfairness when they see it." We therefore assume an auditor that knows intuitively what it means to be fair but cannot explicitly enunciate the fairness metric. And instead of exactly writing down the fairness metric exactly, the auditor can point out pairs of similar individuals who have not received similar predictions, if there exists any such pairs. Then, the goal is to obtain low regret in the online learning problem — measured with respect to the best *fair* policy — while also limiting the total number of rounds where there is any significant fairness violation (i.e. the amount by which the difference in the treatments is more than the distance between two individuals is non-trivial).

2.1.1. Overview of Model and Results

Here we study a setting where we make a structural assumption about the data generation process and the fairness metric. In Chapter 3, we consider a slightly different online learning setting but without these structural assumptions.

Here we study the standard linear contextual bandit setting. In rounds $t = 1, \ldots, T$, a learner observes arbitrary and possibly adversarially selected d-dimensional contexts, each corresponding to one of k actions. The reward for each action is (in expectation) an unknown linear function of the contexts. The learner seeks to minimize its regret.

The learner also wishes to satisfy *fairness constraints*, defined with respect to an unknown distance function defined over contexts. The constraint requires that the difference between the probabilities that any two actions are taken is bounded by the distance between their contexts. The learner has no initial knowledge of the distance function. Instead, after the learner makes its decisions according to some probability distribution π^t at round t, it receives feedback specifying for which pairs of contexts the fairness constraint were violated. Also, the learner sees the reward of the action that is chosen in that round but not the reward of other actions.

Our goal in designing a learner is to simultaneously guarantee near-optimal regret in the contextual bandit problem (with respect to the best *fair* policy), while violating the fairness constraints as infrequently as possible. Our main result is a computationally efficient algorithm that guarantees this for a large class of distance functions known as *Mahalanobis distances* (these can be expressed as $d(x_1, x_2) = ||Ax_1 - Ax_2||_2$ for some matrix A).

Theorem (Informal): There is a computationally efficient learning algorithm in our setting that guarantees that for any Mahalanobis distance, any time horizon T:

1. (Learning) With high probability, $\boldsymbol{L}$ obtains regret $\tilde{O}\left(k^2 d^2 \log(T) + d\sqrt{T}\right)$ to the best fair policy (See Theorem 4 for a precise statement.)

2. (Fairness) For any ϵ, the number of rounds where the unknown fairness constraints are violated by more than ϵ is at most $O\left(k^2 d^2 \log(d/\epsilon)\right)$ with probability 1. (Theorem 5.)

We note that the quoted regret bound requires setting $\epsilon = O(1/T)$, and so this implies a number of fairness violations of magnitude more than $1/T$ that is bounded by a function growing logarithmically in T. Other tradeoffs between regret and fairness violations are possible.

These two goals of obtaining low regret and violating the unknown constraint a small number of times are seemingly in tension. A standard technique for obtaining a mistake bound with respect to fairness violations would be to play a "halving algorithm", which would always act as if the unknown metric is at the center of the current version space (the set of metrics consistent with the feedback observed thus far) — so that mistakes necessarily remove a non-trivial fraction of the version space, making progress. On the other hand, a standard technique for obtaining a diminishing regret bound is to play "optimistically" – i.e. to act as if the unknown metric is the point in the version space that would allow for the largest possible reward. But “optimistic” points are necessarily at the boundary of the version space, and when they are falsified, the corresponding mistakes do not necessarily reduce the version space by a constant fraction.

We prove our theorem in two steps. First, in Section 2.4, we consider the simpler problem in which the linear objective of the contextual bandit problem is known, and the distance function is all that is unknown. In this simpler case, we show how to obtain a bound on the number of fairness violations using a linear-programming based reduction to a recent algorithm which has a mistake bound for learning a linear function with a particularly weak form of feedback [71]. A complication is that our algorithm does not receive all of the feedback that the algorithm of Lobel et al. [71] expects. We need to use the structure of our linear program to argue that this is ok. Then, in Section 2.5, we give our algorithm for the complete problem, using large portions of the machinery we develop in Section 2.4.

We note that in a non-adversarial setting, in which contexts are drawn from a distribution, the algorithm of Lobel et al. [71] could be more simply applied along with standard techniques for contextual bandit learning to give an explore-then-exploit style algorithm. This algorithm would obtain bounded (but suboptimal) regret and the number of fairness violations that grows as a root of T. The principal advantages of our approach are that we are able to give the number of fairness violations that has only *logarithmic* dependence on T, while tolerating contexts that are chosen adversarially, all while obtaining an optimal $\tilde{O}(\sqrt{T})$ regret bound to the best fair policy.

2.2. Related Work

There are a couple of papers that tackle orthogonal issues in metric-fair learning. Rothblum and Yona [81] consider the problem of *generalization* when performing learning subject to a known metric constraint. They show that it is possible to prove relaxed PAC-style generalization bounds without any assumptions on the metric, and that for worst-case metrics, learning subject to a metric constraint can be computationally hard, even when the unconstrained learning problem is easy. In contrast, our work focuses on online learning with an *unknown* metric constraint. Gupta and Kamble [37] also studies online learning subject to individual fairness but with a known metric. They formulate a one-sided fairness constraint across time, called fairness in hindsight and provide an algorithm with regret $O(T^{M/(M+1)})$ for some distribution-dependent constant M.

Kim et al. [60] considers a group-fairness like relaxation of metric-fairness, asking that on average, individuals in pre-specified groups are classified with probabilities proportional to the average distance between individuals in those groups. They show how to learn such classifiers in the offline setting, given access to an oracle which can evaluate the distance between two individuals according to the metric (allowing for unbiased noise). The similarity to our work is that we also consider access to the fairness metric via an oracle, but our oracle is substantially weaker and does not provide numeric valued output. Similarly, Ilvento [45] studies the problem of metric learning by asking human arbiters distance queries. Unlike Ilvento [45], our algorithm does not explicitly learn the underlying similarity measure and does not require asking auditors numeric queries.

There are also several papers in the algorithmic fairness literature that are thematically related to ours, in that they both aim to bridge the gap between group notions of fairness (which can be semantically unsatisfying) and individual notions of fairness (which require very strong assumptions). Zemel et al. [100] attempt to automatically learn a representation for the data in a batch learning problem (and hence, implicitly, a similarity metric) that causes a classifier to label an equal proportion of two protected groups as positive. They provide a heuristic approach and an experimental evaluation.

Two papers (Kearns et al. [59] and Hébert-Johnson et al. [43]) take the approach of asking for a group notion of fairness but over exponentially many implicitly defined protected groups, thus mitigating what Kearns et al. [59] call the “fairness gerrymandering” problem, which is one of the principal weaknesses of group fairness definitions. Both papers give polynomial time reductions which yield efficient algorithms whenever a corresponding agnostic learning problem is solvable. In contrast, we take a different approach: we attempt to directly satisfy the original definition of individual fairness from Dwork et al. [19], but with substantially less information about the underlying similarity metric.

Starting with Joseph et al. [50], several papers have studied notions of fairness in classic and contextual bandit problems. Joseph et al. [50] study a notion of "meritocratic" fairness in the contextual bandit setting, and prove upper and lower bounds on the regret achievable by algorithms that must be "fair" at every round. This can be viewed as a variant of the Dwork et al. [19] notion of fairness, in which the expected reward of each action is used to define the "fairness metric". The algorithm does not originally know this metric, but must discover it through experimentation. Joseph et al. [49] extend the work of Joseph et al. [50] to the setting in which the algorithm is faced with a continuum of options at each time step, and give improved bounds for the *linear* contextual bandit case. Jabbari et al. [46] extend this line of work to the reinforcement learning setting in which the actions of the algorithm can impact its environment. Finally, Liu et al. [70] consider a notion of fairness based on calibration in the simple stochastic bandit setting.

There is a large literature that focuses on learning Mahalanobis distances — see Kulis et al. [65] for a survey. In this chapter, we particularly rely heavily on Lobel et al. [71] which we describe in further detail later. In this literature, the closest paper to our work focuses on *online* learning of Mahalanobis distances (Jain et al. [47]). However, this result is in a very different setting from the one we consider here. In Jain et al. [47], the algorithm is repeatedly given pairs of points, and needs to predict their distance. It then learns their true distance, and aims to minimize its squared loss. In contrast, our main objective of the learning algorithm is orthogonal to the metric learning problem — i.e. to minimize regret in the linear contextual bandit problem, but while simultaneously learning and obeying a fairness constraint, and only from weak feedback noting violations of fairness.

2.3. Preliminaries

We study algorithms that operate in the *linear contextual bandits* setting. A linear contextual bandit problem is parameterized by an unknown vector of linear coefficients $\theta \in \mathbb{R}^d$, with $||\theta||_2 \leq 1$. Algorithms in this setting operate in *rounds* $t = 1, \ldots, T$. In each round t, an algorithm $\boldsymbol{L}$ observes k *contexts* $x_1^t, \ldots, x_k^t \in \mathbb{R}^d$, scaled such that $||x_i^t||_2 \leq 1$. We write $x^t = (x_1^t, \ldots, x_k^t)$ to denote the entire set of contexts observed at round t. After observing the contexts, the algorithm chooses an action i^t. After choosing an action, the algorithm obtains some stochastic *reward* $r_{i^t}^t$ such that $r_{i^t}^t$ is subgaussian[1] and $\mathbb{E}[r_{i^t}^t] = \langle x_{i^t}^t, \theta \rangle$. The algorithm does not observe the reward for the actions not chosen. When the action i^t is clear from context, and write r^t instead of $r_{i^t}^t$.

Remark 1. *For simplicity, we consider algorithms that select only a* single *action at every round. However, this assumption is not necessary. In the appendix of the original paper [35], we show how our results extend to the case in which the algorithm can choose any number of actions at each round. This assumption is sometimes more natural: for example, in a lending scenario, a bank may wish to make loans to as many individuals as will be profitable, without a budget constraint.*

In this chapter, we will be discussing algorithms $\boldsymbol{L}$ that are necessarily randomized. To formalize this, we denote a history including everything observed by the algorithm up through but not including round t as

$$h^t = ((x^1, i^1, r^1), \ldots, (x^{t-1}, i^{t-1}, r^{t-1})).$$

The space of such histories is denoted by $\mathcal{H}^t = (\mathbb{R}^{d \times k} \times [k] \times \mathbb{R})^{t-1}$. An algorithm $\boldsymbol{L}$ is defined by a sequence of functions $f^1, \ldots, f^T$ each mapping histories and observed contexts to probability distributions over actions:

$$f^t : \mathcal{H}^t \times \mathbb{R}^{d \times k} \to \Delta[k].$$

We write π^t to denote the probability distribution over actions that $\boldsymbol{L}$ plays at round t: $\pi^t = f^t(h^t, x^t)$. We view π^t as a vector over $[0,1]^k$, and so π_i^t denotes the probability that

[1] A random variable X with $\mu = \mathbb{E}[X]$ is sub-gaussian, if for all $t \in \mathbb{R}$, $\mathbb{E}[e^{t(X-\mu)}] \leq e^{\frac{t^2}{2}}$.

$\boldsymbol{L}$ plays action i at round t. We denote the expected reward of the algorithm at day t as $\mathbb{E}[r^t] = \mathbb{E}_{i \sim \pi^t}[r_i^t]$. It will sometimes also be useful to refer to the vector of expected rewards across all actions on day t. We denote it as

$$\bar{r}^t = (\langle x_1^t, \theta \rangle, \ldots, \langle x_k^t, \theta \rangle).$$

Note that this vector is of course unknown to the algorithm.

2.3.1. Fairness Constraints and Feedback

We study algorithms that are constrained to behave *fairly* in some manner. We adapt the definition of fairness from Dwork et al. [19] that asserts, informally, that "similar individuals should be treated similarly". We imagine that the decisions that our contextual bandit algorithm $\boldsymbol{L}$ makes correspond to individuals, and that the contexts x_i^t correspond to features pertaining to individuals. We adopt the following (specialization of) the fairness definition from Dwork et al. [19], which is parameterized by a distance function $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$.

Definition 1 (Dwork et al. [19])**.** *Algorithm $\boldsymbol{L}$ is Lipschitz-fair on round t with respect to distance function d if for all pairs of individuals i, j:*

$$|\pi_i^t - \pi_j^t| \leq d(x_i^t, x_j^t).$$

For brevity, we will often just say that the algorithm is fair *at round t, with the understanding that we are always talking about this one particular kind of fairness.*

Remark 2. *Note that this definition requires a fairness constraint that binds between individuals at a single round t, but not between rounds t. This is for several reasons. First, at a philosophical level, we want our algorithms to be able to improve with time, without being bound by choices they made long ago before they had any information about the fairness metric. At a (related) technical level, it is easy to construct lower bound instances that certify that it is impossible to simultaneously guarantee that an algorithm has diminishing regret to the best fair policy, while violating fairness constraints (now defined as binding* *across rounds) a sublinear number of times. See Gupta and Kamble [37] for more discussion regarding this issue.*

One of the main difficulties in working with Lipschitz fairness (as discussed in [19]) is that the distance function d plays a central role, but it is not clear how it should be specified. In this paper, we concern ourselves with learning d from feedback. In particular, algorithms $\boldsymbol{L}$ will have access to an auditor.

Informally, the auditor will take as input:

1. the set of choices available to $\boldsymbol{L}$ at each round t,

2. the probability distribution π^t that $\boldsymbol{L}$ uses to make its choices at round t,

and returns the set of all pairs of individuals for which $\boldsymbol{L}$ violates the fairness constraint.

Definition 2 (Auditor)**.** *Given a distance function d, a fairness oracle O_d is a function $O_d : \mathbb{R}^{d\times k} \times \Delta[k] \to 2^{[k]\times[k]}$ defined such that:*

$$O_d(x^t, \pi^t) = \{(i,j) : |\pi_i^t - \pi_j^t| > d(x_i^t, x_j^t)\}$$

Formally, algorithms $\boldsymbol{L}$ in our setting will operate in the following environment:

1. An adversary fixes a linear reward function $\theta \in \mathbb{R}^d$ with $||\theta|| \leq 1$ and a distance function d. $\boldsymbol{L}$ is given access to the fairness oracle O_d.

2. In rounds $t = 1$ to T:

   (a) The adversary chooses contexts $x^t \in \mathbb{R}^{d\times k}$ with $||x_i^t|| \leq 1$ and gives them to $\boldsymbol{L}$.

   (b) $\boldsymbol{L}$ chooses a probability distribution π^t over actions and chooses action $i^t \sim \pi^t$.

   (c) $\boldsymbol{L}$ receives reward $r_{i^t}^t$ and observes feedback $O_d(\pi^t)$ from the fairness oracle.

Because of the power of the adversary in this setting, we cannot expect algorithms that can avoid arbitrarily small violations of the fairness constraint. Instead, we will aim to limit *significant* violations.

Definition 3. *Algorithm $\boldsymbol{L}$ is ϵ-unfair on pair (i, j) at round t with respect to distance function d if*

$$|\pi_i^t - \pi_j^t| > d(x_i^t, x_j^t) + \epsilon.$$

Given a sequence of contexts and a history h^t (which fixes the distribution on actions at day t), we write

$$\textbf{Unfair}(\boldsymbol{L}, \epsilon, h^t) = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \mathbb{1}(|\pi_i^t - \pi_j^t| > d(x_i^t, x_j^t) + \epsilon)$$

to denote the number of pairs on which $\boldsymbol{L}$ is ϵ-unfair at round t.

Given a distance function d and a history h^{T+1}, the ϵ-*fairness loss* of an algorithm $\boldsymbol{L}$ is the total number of pairs on which it is ϵ-unfair:

$$\textbf{FairnessLoss}(\boldsymbol{L}, h^{T+1}, \epsilon) = \sum_{t=1}^{T} \textbf{Unfair}(\boldsymbol{L}, \epsilon, h^t)$$

For a shorthand, we'll write $\textbf{FairnessLoss}(\boldsymbol{L}, T, \epsilon)$.

We will aim to design algorithms $\boldsymbol{L}$ that guarantee that their fairness loss is bounded with probability 1 in the worst case over the instance: i.e. in the worst case over both θ and $x^1, \ldots, x^T$, and in the worst case over the distance function d (within some allowable class of distance functions – see Section 2.3.3).

2.3.2. Regret to the Best Fair Policy

In addition to minimizing fairness loss, we wish to design algorithms that exhibit diminishing *regret* to the best *fair* policy. We first define a linear program that we will make use of throughout the paper. Given a vector $a \in \mathbb{R}^d$ and a vector $c \in \mathbb{R}^{k^2}$, we denote by $LP(a, c)$ the following linear program:

$$
\begin{aligned}
\underset{\pi=\{p_1,\ldots,p_k\}}{\text{maximize}} \quad & \sum_{i=1}^{k} p_i a_i \\
\text{subject to} \quad & |p_i - p_j| \leq c_{i,j}, \forall (i,j) \\
& \sum_{i=1}^{k} p_i \leq 1
\end{aligned}
$$

We write $\pi(a, c) \in \Delta[k]$ to denote an optimal solution to $LP(a, c)$. Given a set of contexts x^t, recall that $\bar{r}^t$ is the vector representing the expected reward corresponding to each context (according to the true, unknown linear reward function θ). Similarly, we write $\bar{d}^t$ to denote the vector representing the set of distances between each pair of contexts i, j (according to the true, unknown distance function d): $\bar{d}^t_{i,j} = d(x^t_i, x^t_j)$.

Observe that $\pi(\bar{r}^t, \bar{d}^t)$ corresponds to the distribution over actions that maximizes expected reward at round t, subject to satisfying the fairness constraints — i.e. the distribution that an optimal player, with advance knowledge of θ would play, if he were not allowed to violate the fairness constraints at all. This is the benchmark with respect to which we define regret:

Definition 4. *Given an algorithm $\boldsymbol{L}$ $(f_1, \ldots, f_T)$, a distance function d, a linear parameter vector θ, and a history h^{T+1} (which includes a set of contexts $x^1, \ldots, x^T$), its regret is defined to be:*

$$\textbf{Regret}(\boldsymbol{L}, \theta, d, h^{T+1}) = \sum_{t=1}^{T} \mathop{\mathbb{E}}_{i \sim \pi(\bar{r}^t, \bar{d}^t)} [\bar{r}^t_i] - \sum_{t=1}^{T} \mathop{\mathbb{E}}_{i \sim f^t(h^t, x^t)} [\bar{r}^t_i]$$

For shorthand, we'll write $\textbf{Regret}(\boldsymbol{L}, T)$.

Our goal will be to design algorithms for which we can bound regret with high probability over the randomness of h^{T+1} [2] in the worst case over θ, d, and $(x^1, \ldots, x^T)$.

[2] We assume that h^{T+1} is generated by algorithm A, meaning randomness only comes from the stochastic reward and the way in which each arm is selected according to the probability distribution calculated by the algorithm. We don't assume any distributional assumption over $x^1, \ldots, x^T$

2.3.3. Mahalanobis Distance

In this part of the chapter, we will restrict our attention to a special family of distance functions which are parameterized by a matrix A:

Definition 5 (Mahalanobis distances)**.** *A function $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is a Mahalanobis distance function if there exists a matrix A such that for all $x_1, x_2 \in \mathbb{R}^d$:*

$$d(x_1, x_2) = ||Ax_1 - Ax_2||_2$$

where $||\cdot||_2$ denotes Euclidean distance. Note that if A is not full rank, then this does not define a metric — but we will allow this case (and be able to handle it in our algorithmic results).

Mahalanobis distances will be convenient for us to work with, because *squared* Mahalanobis distances can be expressed as follows:

$$\begin{aligned} d(x_1, x_2)^2 &= ||Ax_1 - Ax_2||_2^2 \\ &= \langle A(x_1 - x_2), A(x_1 - x_2) \rangle \\ &= (x_1 - x_2)^\top A^\top A(x_1 - x_2) \\ &= \sum_{i,j=1}^{d} G_{i,j}(x_1 - x_2)_i (x_1 - x_2)_j \end{aligned}$$

where $G = A^\top A$. Observe that when x_1 and x_2 are fixed, this is a linear function in the entries of the matrix G. We will use this property to reason about *learning* G, and thereby learning d.

2.4. Warmup: The Known Objective Case

In this section, we consider an easier case of the problem in which the linear objective function θ is known to the algorithm, and the distance function d is all that is unknown. In this case, we show via a reduction to an online learning algorithm of Lobel et al. [71], how to simultaneously obtain a logarithmic regret bound and a logarithmic (in T) number of fairness violations. The analysis we do here will be useful when we solve the full version of our problem (in which θ is unknown) in Section 2.5.

2.4.1. Outline of the Solution

Recall that since we know θ, at every round t after seeing the contexts, we know the vector of expected rewards $\bar{r}^t$ that we would obtain for selecting each action. Our algorithm will play at each round t the distribution $\pi(\bar{r}^t, \hat{d}^t)$ that results from solving the linear program $LP(\bar{r}^t, \hat{d}^t)$, where $\hat{d}^t$ is a "guess" for the pairwise distances between each context $\bar{d}^t$. (Recall that the optimal distribution to play at each round is $\pi(\bar{r}^t, \bar{d}^t)$.)

The main engine of our reduction is an efficient online learning algorithm for linear functions recently given by Lobel et al. [71] which is further described in Section 2.4.2. Their algorithm, which we refer to as **DistanceEstimator**, works in the following setting. There is an unknown vector of linear parameters $\alpha \in \mathbb{R}^m$. In rounds t, the algorithm observes a vector of features $u^t \in \mathbb{R}^m$ and produces a prediction $g^t \in \mathbb{R}$ for the value $\langle \alpha, u^t \rangle$. After it makes its prediction, the algorithm learns whether its guess was *too large* or not, but does not learn anything else about the value of $\langle \alpha, u^t \rangle$. The guarantee of the algorithm is that the number of rounds in which its prediction is off by more than ϵ is bounded by $O(m \log(m/\epsilon))$[3].

Our strategy will be to instantiate $\binom{k}{2}$ copies of this distance estimator — one for each pair of actions — to produce guesses $(\hat{d}^t_{i,j})^2$ intended to approximate the *squared* pairwise distances $d(x^t_i, x^t_j)^2$. From this, we derive estimates $\hat{d}^t_{i,j}$ of the pairwise distances $d(x^t_i, x^t_j)$. Note that this is a linear estimation problem for any Mahalanobis distance, because by our observation in Section 2.3.3, a squared Mahalanobis distance can be written as a linear function of the $m = d^2$ unknown entries of the matrix $G = A^\top A$ which defines the Mahalanobis distance.

The complication is that the **DistanceEstimator** algorithms expect feedback at every round, which we cannot always provide. This is because the auditor O_d provides feedback about the distribution $\pi(\bar{r}^t, \hat{d}^t)$ used by the algorithm, *not* directly about the guesses $\hat{d}^t$.

[3] If the algorithm also learned whether or not its guess was in error by more than ϵ at each round, variants of the classical halving algorithm could obtain this guarantee. But the algorithm does not receive this feedback, which is why the more sophisticated algorithm of Lobel et al. [71] is needed.

These are not the same, because not all of the constraints in the linear program $LP(\bar{r}^t, \hat{d}^t)$ are necessarily tight — it may be that $|\pi(\bar{r}^t, \hat{d}^t)_i - \pi(\bar{r}^t, \hat{d}^t)_j| < \hat{d}^t_{i,j}$. For any copy of **DistanceEstimator** that does not receive feedback, we can simply "roll back" its state and continue to the next round. But we need to argue that we make progress — that whenever we are ϵ-unfair, or whenever we experience large per-round regret, then there is at least one copy of **DistanceEstimator** that we can give feedback to such that the corresponding copy of **DistanceEstimator** has made a large prediction error, and we can thus charge either our fairness loss or our regret to the mistake bound of that copy of **DistanceEstimator**.

As we show, there are three relevant cases.

1. In any round in which we are ϵ-unfair for some pair of contexts x^t_i and x^t_j, then it must be that $\hat{d}^t_{i,j} \geq d(x^t_i, x^t_j) + \epsilon$, and so we can always update the (i, j)th copy of **DistanceEstimator** and charge our fairness loss to its mistake bound. We formalize this in Lemma 1.

2. For any pair of arms (i, j) such that we have not violated the fairness constraint *and* the (i, j)th constraint in the linear program is tight, we can provide feedback to the (i, j)th copy of **DistanceEstimator** (its guess was not too large). There are two cases. Although the algorithm never knows which case it is in, we handle each case separately in the analysis.

   (a) For every constraint (i, j) in $LP(\bar{r}^t, \hat{d}^t)$ that is *tight* in the optimal solution, $|\hat{d}^t_{i,j} - d(x^t_i, x^t_j)| \leq \epsilon$. In this case, we show that our algorithm does not incur very much per round regret. We formalize this in Lemma 4.

   (b) Otherwise, there is a tight constraint (i, j) such that $|\hat{d}^t_{i,j} - d(x^t_i, x^t_j)| > \epsilon$. In this case, we may incur high per-round regret — but we can charge such rounds to the mistake bound of the (i, j)th copy of **DistanceEstimator** using Lemma 1.

2.4.2. The Distance Estimator

First, we fix some notation for the **DistanceEstimator** algorithm. We write **DistanceEstimator**(ϵ) to instantiate a copy of **DistanceEstimator** with a mistake bound for ϵ-misestimations. The mistake bound we state for **DistanceEstimator** is predicated on the assumption that the norm of the unknown linear parameter vector $\alpha \in \mathbb{R}^m$ is bounded by $||\alpha|| \leq B_1$, and the norms of the arriving vectors $u^t \in \mathbb{R}^m$ are bounded by $||u^t|| \leq B_2$. Given an instantiation of **DistanceEstimator** and a new vector u^t for which we would like a prediction, we write: $g^t =$ **DistanceEstimator**.$guess(u^t)$ for its guess of the value of $\langle \alpha, u^t \rangle$. We use the following notation to refer to the feedback we provide to **DistanceEstimator**: If $g^t > \langle \alpha, u^t \rangle$ and we provide feedback, we write **DistanceEstimator**.$feedback(\top)$. Otherwise, if $g^t \leq \langle \alpha, u^t \rangle$ and we give feedback, we write **DistanceEstimator**.$feedback(\bot)$. In some rounds, we may be unable to provide the feedback that **DistanceEstimator** is expecting: in these rounds, we simply "roll-back" its internal state. We can do this because the mistake bound for **DistanceEstimator** holds for *every* sequence of arriving vectors u^t. If we give feedback to **DistanceEstimator** in a given round t, we write $v^t = 1$ and $v^t = 0$ otherwise.

Definition 6. *Given an accuracy parameter ϵ, a linear parameter vector α, a sequence of vectors $u^1, \ldots, u^T$, a sequence of guesses $g^1, \ldots, g^T$ and a sequence of feedback indicators, $v^1, \ldots, v^T$, the number of valid ϵ-mistakes made by* **DistanceEstimator** *is:*

$$\mathbf{Mistakes}(\epsilon) = \sum_{t=1}^{T} \mathbb{1}(v^t = 1 \wedge |g^t - \langle u^t, \alpha \rangle| > \epsilon)$$

In other words, it is the number of ϵ-mistakes made by **DistanceEstimator** *in rounds for which we provided the algorithm feedback.*

We now state a version of the main theorem from Lobel et al. [71], adapted to our setting[4]:

[4] In [71], the algorithm receives feedback in every round, and the scale parameters B_1 and B_2 are normalized to be 1. But the version we state is an immediate consequence.

Lemma 1 (Lobel et al. [71])**.** *For any $\epsilon > 0$ and any sequence of vectors $u^1, \ldots, u^T$,* **DistanceEstimator**(ϵ) *makes a bounded number of valid ϵ-mistakes.*

$$\textbf{Mistakes}(\epsilon) = O\left(m \log\left(\frac{m \cdot B_1 \cdot B_2}{\epsilon}\right)\right)$$

2.4.3. The Algorithm

Algorithm 1: $\boldsymbol{L}_{\text{known}-\theta}$

```
for $i, j = 1, \ldots, k$ do
    $\textbf{DistanceEstimator}_{i,j} = \textbf{DistanceEstimator}(\epsilon^2)$
end
for $t = 1, \ldots, T$ do
    receive the contexts $x^t = (x^t_1, \ldots, x^t_k)$
    for $i, j = 1, \ldots, k$ do
        $u^t_{i,j} = flatten((x^t_i - x^t_j)(x^t_i - x^t_j)^\top)$
        $g^t_{i,j} = \textbf{DistanceEstimator}_{ij}.guess(u^t_{i,j})$
        $\hat{d}^t_{i,j} = \sqrt{g^t_{i,j}}$
    end
    $\pi^t = \pi(\bar{r}^t, \hat{d}^t)$
    Pull an arm $i^t$ according to $\pi^t$ and receive a reward $r^t_{i^t}$
    $S = \boldsymbol{O}_d(x^t, \pi^t)$
    $R = \{(i,j) | (i,j) \notin S \wedge |p^t_i - p^t_j| = \hat{d}^t_{ij}\}$
    for $(i,j) \in S$ do
        $\textbf{DistanceEstimator}_{ij}.feedback(\bot)$
        $v^t_{ij} = 1$
    end
    for $(i,j) \in R$ do
        $\textbf{DistanceEstimator}_{ij}.feedback(\top)$
        $v^t_{ij} = 1$
    end
end
```

For each pair of arms $i, j \in [k]$, our algorithm instantiates a copy of **DistanceEstimator**(ϵ^2), which we denote by **DistanceEstimator**$_{i,j}$: we also subscript all variables relevant to **DistanceEstimator**$_{i,j}$ with i, j (e.g. $u^t_{i,j}$). The underlying linear parameter vector we want to learn $\alpha = flatten(G) \in \mathbb{R}^{d^2}$, where $flatten : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \cdot n}$ maps a $m \times n$ matrix to a vector of size mn by concatenating its rows into a vector. Similarly, given a pair of contexts x^t_i, x^t_j, we will define $u^t_{i,j} = flatten((x^t_i - x^t_j)(x^t_i - x^t_j)^\top)$. **DistanceEstimator**$_{i,j}.guess(u^t_{i,j})$

will output guess $g^t_{i,j}$ for the value $\langle \alpha, u^t_{i,j} \rangle = (\bar{d}^t_{i,j})^2$, as

$$\langle flatten(G), flatten((x^t_i - x^t_j)(x^t_i - x^t_j)^\top) \rangle = \sum_{a,b=1}^{d} G_{a,b}(x^t_i - x^t_j)_a (x^t_i - x^t_j)_b = (\bar{d}^t_{i,j})^2$$

We take $\hat{d}^t_{i,j} = \sqrt{g^t_{i,j}}$ as our estimate for the distance between x^t_i and x^t_j.

The algorithm then chooses an arm to pull according to the distribution $\pi(\bar{r}^t, \hat{d}^t)$, where $\bar{r}^t_i = \langle \theta, x_i \rangle$. The auditor O_d returns all pairs of arms that violate the fairness constraints. For these pairs (i, j) we provide feedback to **DistanceEstimator**$_{i,j}$: the guess was too large. For the remaining pairs of arms (i, j), there are two cases. If the (i, j)th constraint in $LP(\bar{r}^t, \hat{d}^t)$ was not tight, then we provide no feedback ($v^t_{i,j} = 0$). Otherwise, we provide feedback: the guess was not too large. The pseudocode appears as Algorithm 1.

First we derive the valid mistake bound that the **DistanceEstimator**$_{i,j}$ algorithms incur in our parameterization.

Lemma 2. *For pair (i, j), the total number of valid ϵ^2 mistakes made by* **DistanceEstimator**$_{i,j}$ *is bounded as:*

$$\mathbf{Mistakes}(\epsilon^2) = O\left(d^2 \log\left(\frac{d \cdot ||A^\top A||_F}{\epsilon}\right)\right)$$

where the distance function is defined as $d(x_i, x_j) = ||Ax_i - Ax_j||_2$ and $|| \cdot ||_F$ denotes the Frobenius norm.

Proof. This follows directly from Lemma 1, and the observations that in our setting, $m = d^2$, $B_1 = ||\alpha|| = ||A^\top A||_F$, and

$$B_2 \leq \max_t ||u^t_{i,j}||_2 \leq \max_t ||x^t_i - x^t_j||^2 \leq 4.$$

□

We next observe that since we only instantiate k^2 copies of **DistanceEstimator** in total, Lemma 2 immediately implies the following bound on the total number of rounds in which *any* distance estimator that receives feedback provides us with a distance estimate that differs by more than ϵ from the correct value:

Corollary 1. *The number of rounds where there exists a pair (i, j) such that feedback is provided ($v^t_{i,j} = 1$) and its estimate is off by more than ϵ is bounded:*

$$\left|\{t : \exists(i,j) : v^t_{ij} = 1 \wedge |\hat{d}^t_{i,j} - \bar{d}^t_{i,j}| > \epsilon\}\right| \leq O\left(k^2 d^2 \log\left(\frac{d \cdot ||A^\top A||_F}{\epsilon}\right)\right)$$

Proof. This follows from summing the k^2 valid ϵ^2 mistake bounds for each copy of $\textbf{DistanceEstimator}_{i,j}$, and noting that an ϵ mistake in predicting the value of $\bar{d}^t_{i,j}$ implies an ϵ^2 mistake in predicting the value of $(\bar{d}^t_{i,j})^2$. □

We now have the pieces to bound the ϵ-unfairness loss of our algorithm:

Theorem 1. *For any sequence of contexts and any Mahalanobis distance $d(x_1, x_2) = ||Ax_1 - Ax_2||_2$:*

$$\textbf{FairnessLoss}(\boldsymbol{L}_{known-\theta}, T, \epsilon) \leq O\left(k^2 d^2 \log\left(\frac{d \cdot ||A^T A||_F}{\epsilon}\right)\right)$$

Proof.

$$\begin{aligned}
\textbf{FairnessLoss}(\boldsymbol{L}_{\text{known}-\theta}, T, \epsilon) &= \sum_{t=1}^{T} \textbf{Unfair}(\boldsymbol{L}_{\text{known}-\theta}, \epsilon) \\
&\leq \sum_{t=1}^{T} \sum_{i,j} \mathbb{1}(|\pi^t_i - \pi^t_j| > \bar{d}^t_{ij} + \epsilon) \\
&= \sum_{i,j} \sum_{t=1}^{T} \mathbb{1}(\{v^t_{ij} = 1 \wedge \hat{d}^t_{ij} > d^t_{ij} + \epsilon\}) \\
&\leq \sum_{i,j} \sum_{t=1}^{T} \mathbb{1}(\{v^t_{ij} = 1 \wedge |\hat{d}^t_{ij} - d^t_{ij}| > \epsilon\}) \\
&= O\left(k^2 d^2 \log\left(\frac{d \cdot ||A^\top A||_F}{\epsilon}\right)\right) \qquad \text{Corollary 1}
\end{aligned}$$

□

We now turn our attention to bounding the regret of the algorithm. Recall from the overview in Section 2.4.1, that our plan will be to divide rounds into two types. In rounds of the first type, our distance estimates corresponding to every *tight constraint* in the linear program have only small error. We cannot bound the number of such rounds, but we can bound the regret incurred in any such rounds. In rounds of the second type, we have at least one significant error in the distance estimate corresponding to a tight constraint. We might incur significant regret in such rounds, but we can bound the number of such rounds.

The following lemma bounds the *decrease* in expected per-round reward that results from under-estimating a *single* distance constraint in our linear programming formulation.

Lemma 3. *Fix any vector of distance estimates d and any vector of rewards r. Fix a constant ϵ and any pair of coordinates $(a, b) \in [k] \times [k]$. Let d' be the vector such that $d'_{ab} = d_{ab} - \epsilon$ and $d'_{ij} = d_{ij}$ for $(i, j) \neq (a, b)$, then $\langle r, \pi(r, d)\rangle - \langle r, \pi(r, d')\rangle \leq \epsilon \sum_{i=1}^{k} r_i$*

Proof. The plan of the proof is to start with $\pi(r, d)$ and perform surgery on it to arrive at a new probability distribution $p' \in \Delta k$ that satisfies the constraints of $LP(r, d')$ and obtains objective value at least $\langle r, p'\rangle \geq \langle r, \pi(r, d)\rangle - \epsilon \sum_{i=1}^{k} r_i$. Because p' is feasible, it lower bounds the objective value of the optimal solution $\pi(r, d')$, which yields the theorem.

To reduce notational clutter, for the rest of the argument we write p to denote $\pi(r, d)$. Without loss of generality, we assume that $p_a \geq p_b$. If $p_a - p_b \leq d_{ab} - \epsilon$, then p_i is still a feasible solution to $LP(r, d')$, and we are done. Thus, for the rest of the argument, we can assume that $p_a - p_b > d_{ab} - \epsilon$. We write $\Delta = (p_a - p_b) - (d_{ab} - \epsilon) > 0$

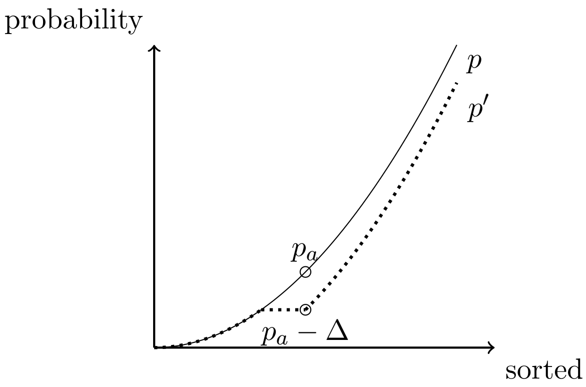


Figure 1: A visual interpretation of the surgery performed on p in the proof of Lemma 3 to obtain P'. Note that the surgery manages to shrink the distance between p_a and p_b without increasing the distance between any other pair of points.

We now define our modified distribution p':

$$p'_i = \begin{cases} p_i - \Delta & p_a \leq p_i \\ p_a - \Delta & p_a - \Delta \leq p_i < p_a \\ p_i & \text{otherwise} \end{cases}$$

We'll partition the coordinates of p_i into which of the three cases they fall into in our definition of p' above. $S_1 = \{i | p_a \leq p_i\}$, $S_2 = \{i | p_a - \epsilon \leq p_i < p_a\}$, and $S_3 = \{i | i < p_b + (d_{ab} - \epsilon)\}$. It remains to verify that p' is a feasible solution to $LP(r, d')$, and that it obtains the claimed objective value.

Feasibility: First, observe that $\sum_i p'_i \leq 1$. This follows because p' is coordinate-wise smaller than p, and by assumption, p was feasible. Thus, $\sum_i p'_i \leq \sum_i p_i \leq 1$.

Next, observe that by construction, $p'_i \geq 0$ for all i. To see this, first observe that $p_a - \Delta = p_b + (d_{ab} - \epsilon) \geq 0$ where the last inequality follows because $d_{ab} \geq \epsilon$. We then consider the three cases:

1. For $i \in S_1$, $p'_i = p_i - \Delta \geq p_a - \Delta \geq 0$ because $p_i \geq p_a$.

2. For $i \in S_2$, $p'_i = p_a - \Delta \geq 0$.

3. For $i \in S_3$, $p'_i = p_i \geq 0$.

Finally, we verify that for all (i, j), $|p'_i - p'_j| \leq d'_{ij}$. First, observe that $p'_a - p'_b = (p_b + (d_{ab} - \epsilon)) - p'_b = d_{ab} - \epsilon = d'_{ab}$, and so the inequality is satisfied for index pair (a, b). For all the other pairs $(i, j) \neq (a, b)$, we have $d'_{ij} = d_{ij}$, so it is enough to show that $|p'_i - p'_j| \leq d_{ij}$. Note that for all $x, y \in \{1, 2, 3\}$ with $x < y$, if $i \in S_x$ and $j \in S_y$, we have that $x \leq y$. Therefore, it is sufficient to verify the following six cases:

1. $i \in S_1, j \in S_1$: $|p'_i - p'_j| = (p_i - \Delta) - (p_j - \Delta) = p_i - p_j \leq d_{ij}$

2. $i \in S_1, j \in S_2$: $|p'_i - p'_j| = (p_i - \Delta) - (p_a - \Delta) = p_i - p_a < p_i - p_j \leq d_{ij}$

3. $i \in S_1, j \in S_3$: $|p'_i - p'_j| = (p_i - \Delta) - p_j = (p_i - p_j) - \Delta \leq (p_i - p_j) \leq d_{ij}$

4. $i \in S_2, j \in S_2$: $|p'_i - p'_j| = (p_a - \Delta) - (p_a - \Delta) = 0 \leq d_{ij}$

5. $i \in S_2, j \in S_3$: $|p'_i - p'_j| = (p_a - \Delta) - p_j \leq p_i - p_j \leq d_{ij}$

6. $i \in S_3, j \in S_3$: $|p'_i - p'_j| = p_i - p_j \leq d_{ij}$

Thus, we have shown that p' is a feasible solution to $LP(r, d')$.

Objective Value: Note that for each index i, $p_i - p'_i \leq \Delta \leq \epsilon$. Therefore we have:

$$\begin{aligned} \langle r, \pi(r,d)\rangle - \langle r, \pi(r,d')\rangle &\leq \langle r, \pi(r,d)\rangle - \langle r, p'\rangle \\ &= \langle r, p - p'\rangle \\ &\leq \epsilon \sum_{i=1}^{k} r_i \end{aligned}$$

which completes the proof. □

We now prove the main technical lemma of this section. It states that in any round in which the error of our distance estimates for *tight constraints* is small (even if we have high error in the distance estimates for slack constraints), then we will have low per-round regret.

Lemma 4. *At round t, if for all pairs of indices (i,j), we have either:*

1. $|\hat{d}^t_{i,j} - \bar{d}^t_{i,j}| \leq \epsilon$ *or*

2. $v^t_{i,j} = 0$ *(corresponding to an LP constraint that is not tight)*

then:

$$\langle r^t, \pi(r^t, \bar{d}^t)\rangle - \langle r^t, \pi(r^t, \hat{d}^t)\rangle \leq \epsilon k^3$$

for any vector r^t with $||r^t||_\infty \leq 1$.

Proof. First, define $\tilde{d}^t$ to be the coordinate-wise maximum of $\hat{d}^t$ and $\bar{d}^t$: i.e. the vector such that for every pair of coordinates i,j, $\tilde{d}_{ij} = \max(\bar{d}_{ij}, \hat{d}_{ij})$. To simplify notation, we will write $\hat{p} = \pi(r^t, \hat{d}^t)$, $\bar{p} = \pi(r^t, \bar{d}^t)$, and $\tilde{p} = \pi(r^t, \tilde{d}^t)$.

We make three relevant observations:

1. First, because $LP(r^t, \tilde{d}^t)$ is a relaxation of $LP(r^t, \bar{d}^t)$, it has only larger objective value. In other words, we have that $\langle r^t, \tilde{p}\rangle \geq \langle r^t, \bar{p}\rangle$. Thus, it suffices to prove that $\langle r^t, \hat{p}\rangle \geq \langle r^t, \tilde{p}\rangle - \epsilon k^3$.

2. Second, for all pairs i,j, $|\hat{d}^t_{i,j} - \tilde{d}^t_{i,j}| \leq |\hat{d}^t_{i,j} - \bar{d}^t_{i,j}|$. Thus, if we had $|\hat{d}^t_{i,j} - \bar{d}^t_{i,j}| \leq \epsilon$, we also have $|\hat{d}^t_{i,j} - \tilde{d}^t_{i,j}| \leq \epsilon$.

3. Finally, by construction, for every pair (i,j), we have $\tilde{d}_{ij} \geq \hat{d}_{ij}$

Let S_1 be the set of indices (i,j) such that $|\hat{d}^t_{i,j} - \tilde{d}^t_{i,j}| \leq \epsilon$, and let S_2 be the set of indices $(i,j) \notin S_1$ such that $v^t_{i,j} = 0$. Note that by assumption, these partition the space, and that by construction, for every $(i,j) \in S_2$, the corresponding constraint in $LP(r^t, \hat{d}^t)$ is not tight: i.e. $|\hat{p}_i - \hat{p}_j| < \hat{d}^t_{i,j}$. Let d^* be the vector such that for all $(i,j) \in S_1$, $d^*_{ij} = \hat{d}_{ij}$, and for all $(i,j) \in S_2$, $d^*_{ij} = \tilde{d}_{ij}$. Observe that $LP(r^t, d^*)$ corresponds to a relaxation of $LP(r^t, \hat{d})$ in which *only constraints that were already slack were relaxed.* As a result, $\hat{p}$ is also an optimal solution to $LP(r^t, d^*)$. Note also that by construction, we now have that for *every* pair (i, j): $|\tilde{d}_{ij} - d^*_{ij}| \leq \epsilon$

Our argument will proceed by describing a sequence of $n+1 = k^2+1$ vectors $p^0, p^1, \ldots, p^n$ such that $p^0 = \tilde{p}$, p^n is a feasible solution to $LP(r^t, d^*)$, and for all adjacent pairs $p^\ell, p^{\ell+1}$, we have: $\langle r^t, p^{\ell+1}\rangle \geq \langle r^t, p^\ell\rangle - \epsilon k$. Telescoping these inequalities yields:

$$\langle r^t, \hat{p}\rangle \geq \langle r^t, p^n\rangle \geq \langle r^t, \tilde{p}\rangle - k^3\epsilon$$

which will complete the proof.

To finish the argument, fix an arbitrary ordering on the indices $(i, j) \in [k] \times [k]$, which we denote by $(i_1, j_1), \ldots, (i_n, j_n)$. Define the distance vector d^ℓ such that:

$$d^\ell_{i_a, j_a} = \begin{cases} \tilde{d}_{i_a, j_a}, & \text{If } a > \ell; \\ d^*_{i_a, j_a}, & \text{If } a \leq \ell. \end{cases}$$

Note that the sequence of distance vectors $d^1, \ldots, d^n$ “walks between” $\tilde{d}$ and d^* one coordinate at a time. Now let $p^\ell = \pi(r^t, d^\ell)$. By construction, we have that every pair $(d^\ell, d^{\ell+1})$ differ in only a single coordinate, and that the difference has magnitude at most ϵ. Therefore, we can apply Lemma 3 to conclude that:

$$\langle r^t, p^{\ell+1}\rangle \geq \langle r^t, p^\ell\rangle - \epsilon \sum_{i=1}^{k} r^t_i \geq \langle r^t, p^\ell\rangle - \epsilon k$$

as desired. □

Finally, we have all the pieces we need to prove a regret bound for $\boldsymbol{L}_{\text{known}-\theta}$.

Theorem 2. *For any time horizon T:*

$$\textbf{Regret}(\boldsymbol{L}_{known-\theta}, T) \le O\left(k^2 d^2 \log\left(\frac{d \cdot ||A^\top A||_F}{\epsilon}\right) + k^3 \epsilon T\right)$$

Setting $\epsilon = O(1/(k^3 T))$ yields a regret bound of $O(d^2 \log(||A^\top A||_F \cdot dkT))$.

Proof. We partition the rounds t into two types. Let S_1 denote the rounds such that there is at least one pair of indices (i, j) such that one instance $\textbf{DistanceEstimator}_{ij}$ produced an estimate that had error more than ϵ, and it was provided feedback. We let S_2 denote the remaining rounds, for which for *every* pair of indices (i, j), *either* $\textbf{DistanceEstimator}_{ij}$ produced an estimate that had error at most ϵ, or $\textbf{DistanceEstimator}_{ij}$ was not given feedback.

$$S_1 = \{t : \exists (i,j) : |\hat{d}^t_{ij} - \bar{d}^t_{ij}| > \epsilon \text{ and } v^t_{ij} = 1\} \quad S_2 = \{t : \forall (i,j) : |\hat{d}^t_{ij} - \bar{d}^t_{ij}| \le \epsilon \text{ or } v^t_{ij} = 0\}$$

Observe that S_1 and S_2 partition the set of all rounds. Next, observe that Corollary 1 tells us that:

$$|S_1| \le O\left(k^2 d^2 \log\left(\frac{d \cdot ||A^\top A||_F}{\epsilon}\right)\right)$$

and Lemma 4 tells us that for every round $t \in S_2$, the per-round regret is at most ϵk^3. Together with the facts that $|S_2| \le T$ and that the per-round regret for any $t \in S_1$ is at most 1, we obtain:

$$\textbf{Regret}(\boldsymbol{L}_{\text{known}-\theta}, T) \le O\left(k^2 d^2 \log\left(\frac{d \cdot ||A^\top A||_F}{\epsilon}\right) + k^3 \epsilon T\right)$$

□

2.5. The Full Algorithm

In this section, we present our final algorithm, which has no knowledge of either the distance function d or the linear objective θ. The resulting algorithm shares many similarities with the algorithm we developed in Section 2.4, and so much of the analysis can be reused.

2.5.1. Outline of the Solution

At a high level, our plan will be to combine the techniques we developed in Section 2.4 with a standard "optimism in the face of uncertainty" strategy for learning the parameter vector θ. Our algorithm will maintain a ridge-regression estimate $\tilde{\theta}$ together with confidence regions derived by Abbasi-Yadkori et al. [1]. After it observes the contexts x_i^t at round t, it uses these to derive upper confidence bounds on the expected rewards, corresponding to each context — represented as a vector $\hat{r}^t$. The algorithm continues to maintain distance estimates $\hat{d}^t$ using the **DistanceEstimator** subroutines, identically to how they were used in Section 2.4. At ever round, the algorithm then chooses its action according to the distribution $\pi^t = \pi(\hat{r}^t, \hat{d}^t)$.

The regret analysis of the algorithm follows by decomposing the per-round regret into two pieces. The first can be bounded by the sum of the *expected widths* of the confidence intervals corresponding to each context x_i^t that might be chosen at each round t, where the expectation is over the randomness of the algorithm's distribution π^t. A theorem of Abbasi-Yadkori et al. [1] bounds the sum of the widths of the confidence intervals corresponding to arms *actually chosen* by the algorithm (Lemma 6). Using a martingale concentration inequality, we are able to relate these two quantities (Lemma 7). We show that the second piece of the regret bound can be manipulated into a form that can be bounded using Lemmas 1 and 4 from Section 2.4 (Theorem 4).

2.5.2. Confidence Intervals from Abbasi-Yadkori et al. [1]

We would like to be able to construct confidence intervals at each round t around each arm's expected reward such that for each arm i, with probability $1-\delta$, $\bar{r}_i^t \in [\tilde{r}_i^t + w_i^t, \tilde{r}_i^t + w_i^t]$, where $\tilde{r}_i^t$ is our ridge-regression estimate of $\bar{r}_i^t$ and w_i^t is the confidence interval width around the estimate. Our algorithm will make use of such confidence intervals for the ridge regression estimator derived and analyzed in [1], which we recount here.

Let $\tilde{V}^t = {X^t}^\top X^t + \lambda I$ be a regularized design matrix, where $X^t = [x_{i_1}^1, \ldots, x_{i_{t-1}}^{t-1}]$ represents all the contexts whose rewards we have observed up to but not including time t. Let $Y^t = [r^1_{i_1}, \ldots, r^{t-1}_{i_{t-1}}]$ be the corresponding vector of observed rewards. $\tilde{\theta} = (V^t)^{-1} {X^t}^\top Y^t$ is the (ridge regression) regularized least squares estimator we use at time t. We write $\tilde{r}^t_i = \langle \tilde{\theta}, x^t_i \rangle$ for the reward point prediction that this estimator makes at time t for arm i.

We can construct the following confidence intervals around $\tilde{r}^t$:

Lemma 5 (Abbasi-Yadkori et al. [1])**.** *With probability* $1 - \delta$*,*

$$|\bar{r}^t_i - \tilde{r}^t_i| = |\langle x^t_i, (\theta - \tilde{\theta}) \rangle| \leq \|x^t_i\|_{(\bar{V^t})^{-1}} \left(\sqrt{2d \log \left(\frac{1 + t/\lambda}{\delta} \right)} + \sqrt{\lambda} \right)$$

for all $i \in [k]$ *where* $||x||_A = \sqrt{x^\top A x}$

Therefore, the confidence interval widths we use in our algorithm will be

$$w^t_i = \min \left(\|x^t_i\|_{(\bar{V^t})^{-1}} \left(\sqrt{2d \log \left(\frac{1 + t/\lambda}{\delta} \right)} + \sqrt{\lambda} \right), 1 \right)$$

(expected rewards are bounded by 1 in our setting, and so the minimum maintains the validity of the confidence intervals). The upper confidence bounds we use to compute our distribution over arms will be $\hat{r}^t_i = \tilde{r}^t_i + w^t_i$. We will write $w^t = [w^t_1, \ldots, w^t_k]$ to denote the vector of confidence interval widths at round t.

Little can be said about the widths of these confidence intervals in isolation. However, the following theorem bounds the *sum* (over time) of the widths of the confidence intervals around the contexts actually selected.

Lemma 6 (Abbasi-Yadkori et al. [1])**.**

$$\sum_{t=1}^{T} w^t_{i^t} \leq \sqrt{2d \log \left(1 + \frac{T}{d\lambda}\right)} \left(\sqrt{2dT \log(\frac{1 + T/\lambda}{\delta})} + \sqrt{T\lambda} \right)$$

2.5.3. The Algorithm

The pseudocode for the full algorithm is given in Algorithm 2.

In our proof of Theorem 4, we will connect the regret of $\boldsymbol{L}_{full}$ to the sum of the *expected* widths of the confidence intervals pulled at each round. In contrast, what is bounded by Lemma 6 is the sum of the *realized* widths. Using the Azuma Hoeffding inequality, we can relate these two quantities.

Theorem 3 (Azuma-Hoeffding inequality ([44]))**.** *Suppose* $\{X_k : k = 0, 1, 2, 3, \ldots\}$ *is a martingale and*

$$|X_k - X_{k-1}| < c_k.$$

Then, for all positive integers N *and all positive reals* t*,*

$$\Pr(X_N - X_0 \geq t) \leq \exp(\frac{t^2}{2\sum_{k=1}^{N} c_k^2})$$

Lemma 7.

$$\Pr\left(\sum_{t=1}^{T} \mathbb{E}_{i\sim\pi^t}[w_i^t] - \sum_{t=1}^{T} w_{i^t}^t \geq \sqrt{2T\log\frac{1}{\delta}}\right) \leq \delta$$

Proof. Once $x^1, \ldots, x^{t-1}, r_{i^t}^1, \ldots, r_{i^{t-1}}^{t-1}$ and x^t are fixed, π^t is fixed. In other words, for the filtration $\mathscr{F}^t = \sigma(x^1, \ldots, x^{t-1}, r_{i^t}^1, \ldots, r_{i^{t-1}}^{t-1}, x^t)$, $w_{i^t}^t$ is $\mathscr{F}^t$ measurable. Now, define

$$D^t = \sum_{s=1}^{t} \mathbb{E}_{i\sim\pi^s}[w_i^s] - \sum_{s=1}^{t} w_{i^s}^s$$

with respect to $\mathscr{F}^t$. One can think of D^t as the accumulated difference between the confidence width of the arm that was actually pulled and the expected confidence width. It's easy to see that $\{D^t\}$ is a martingale, as $\mathbb{E}[D^1] = 0$, and $\mathbb{E}[D^{t+1}|\mathscr{F}^t] = D^t$.

Also, $D_t - D_{t-1} = w_{i^t}^t - \mathbb{E}_{i\sim\pi^t}[w_i^t] \leq 1$, since the confidence interval widths are bounded by 1.

Algorithm 2: $\boldsymbol{L}_{\text{full}}$

```
for $i, j = 1, \ldots, k$ do
    $\textbf{DistanceEstimator}_{ij} = \textbf{DistanceEstimator}(\epsilon^2)$
end
for $t = 1, \ldots, T$ do
    receive the contexts $x^t = (x_1^t, \ldots, x_k^t)$
    $X^t = [x^1, \ldots, x^{t-1}]$
    $Y^t = [r^t, \ldots, r^{t-1}]$
    $\tilde{V}^t = {X^t}^\top X^t + \lambda I$
    $\tilde{\theta} = (V^t)^{-1} {X^t}^\top Y^t$
    for $i = 1, \ldots, k$ do
        $\tilde{r}_i^t = \langle \tilde{\theta}, x_i^t \rangle$
        $w_i^t = \min\left( \|x_i^t\|_{(\bar{V^t})^{-1}} \left( \sqrt{2d \log(\frac{1+t/\lambda}{\delta})} + \sqrt{\lambda} \right), 1 \right)$
        $\hat{r}_i^t = \tilde{r}_i^t + w_i^t$
    end
    for $i, j = 1, \ldots, k$ do
        $u_{i,j}^t = flatten((x_i^t - x_j^t)(x_i^t - x_j^t)^T))$
        $g_{i,j}^t = \textbf{DistanceEstimator}_{i,j}.guess(u_{i,j}^t)$
        $\hat{d}_{ij}^t = \sqrt{g_{i,j}^t}$
    end
    $\pi^t = \pi(\hat{r}^t, \hat{d}^t)$
    Pull an arm $i^t$ according to $\pi^t$ and receive a reward $r_{i^t}^t$
    $S = \boldsymbol{O}_d(x^t, \pi^t)$
    $R = \{(i,j) | (i,j) \notin S \wedge |\pi_i^t - \pi_j^t| = \hat{d}_{i,j}^t\}$
    for $(i, j) \in S$ do
        $\textbf{DistanceEstimator}_{i,j}.feedback(\perp)$
        $v_{i,j}^t = 1$
    end
    for $(i, j) \in R$ do
        $\textbf{DistanceEstimator}_{i,j}.feedback(\top)$
        $v_{i,j}^t = 1$
    end
end
```

Applying the Azuma-Hoeffding inequality gives us the following:

$$\Pr(\sum_{t=1}^{T} \mathbb{E}_{i\sim\pi^t}[w_i^t] - \sum_{t=1}^{T} w_{i^t}^t \geq \epsilon) = \Pr(D^T \geq \epsilon) \leq \exp(\frac{-\epsilon^2}{2T})$$

Now, setting $\epsilon = \sqrt{2T \ln \frac{1}{\delta}}$ yields:

$$\Pr(\sum_{t=1}^{T} \mathbb{E}_{i\sim\pi^t}[w_i^t] - \sum_{t=1}^{T} w_{i^t}^t \geq \sqrt{2T \log \frac{1}{\delta}}) \leq \delta$$

□

Theorem 4. *For any time horizon T, with probability $1-\delta$:*

$$\textbf{Regret}(\boldsymbol{L}_{full}, T) \leq O\left(k^2 d^2 \log\left(\frac{d \cdot ||A^\top A||_F}{\epsilon}\right) + k^3 \epsilon T + d\sqrt{T} \log\left(\frac{T}{\delta}\right)\right)$$

If $\epsilon = 1/k^3 T$, this is a regret bound of $O\left(k^2 d^2 \log\left(kdT \cdot ||A^\top A||_F\right) + d\sqrt{T}\log(\frac{T}{\delta})\right)$

Proof. We can compute:

$$\begin{aligned}
\textbf{Regret}(\boldsymbol{L}_{full}, T) &= \sum_{t=1}^{T} \mathop{\mathbb{E}}_{i\sim\pi(\bar{r}^t,\bar{d}^t)}[\bar{r}_i^t] - \sum_{t=1}^{T} \mathop{\mathbb{E}}_{i\sim\pi(\hat{r}^t,\hat{d}^t)}[\bar{r}_i^t] \\
&= \sum_{t=1}^{T} \langle \bar{r}^t, \pi(\bar{r}^t, \bar{d}^t)\rangle - \langle \bar{r}^t, \pi(\hat{r}^t, \hat{d}^t)\rangle \\
&= \sum_{t=1}^{T} \langle \bar{r}^t, \pi(\bar{r}^t, \bar{d}^t)\rangle - \langle \bar{r}^t, \pi(\hat{r}^t, \bar{d}^t)\rangle + \langle \bar{r}^t, \pi(\hat{r}^t, \bar{d}^t)\rangle - \langle \bar{r}^t, \pi(\hat{r}^t, \hat{d}^t)\rangle \\
&\leq \sum_{t=1}^{T} \langle \hat{r}^t, \pi(\hat{r}^t, \bar{d}^t)\rangle - \langle \bar{r}^t, \pi(\hat{r}^t, \bar{d}^t)\rangle + \langle \bar{r}^t, \pi(\hat{r}^t, \bar{d}^t)\rangle - \langle \bar{r}^t, \pi(\hat{r}^t, \hat{d}^t)\rangle \\
&\leq \sum_{t=1}^{T} \langle 2w^t, \pi(\hat{r}^t, \bar{d}^t)\rangle + \langle \bar{r}^t, \pi(\hat{r}^t, \bar{d}^t)\rangle - \langle \bar{r}^t, \pi(\hat{r}^t, \hat{d}^t)\rangle
\end{aligned}$$

Here, the first inequality follows from the fact that $\hat{r}^t$ is coordinate-wise larger than $\bar{r}^t$ with probability 1-δ, and that $\pi(\hat{r}^t, \bar{d}^t)$ is the optimal solution to $LP(\hat{r}^t, \bar{d}^t)$. The second inequality follows from $\bar{r} \in [\tilde{r} - w, \tilde{r} + w] = [\hat{r} - 2w, \hat{r}]$.

Just as in the proof of Theorem 2, we now partition time into two sets:

$$S_1 = \{t : \exists (i,j) : |\hat{d}_{ij}^t - \bar{d}_{ij}^t| > \epsilon \text{ and } v_{ij}^t = 1\} \quad S_2 = \{t : \forall (i,j) : |\hat{d}_{ij}^t - \bar{d}_{ij}^t| \leq \epsilon \text{ or } v_{ij}^t = 0\}$$

Recall that corollary 1 bounds $|S_1| \leq O\left(k^2 d^2 \log\left(\frac{d \cdot ||A^\top A||_F}{\epsilon}\right)\right)$. Since the per-step regret of our algorithm can be at most 1, this means that rounds $t \in S_1$ can contribute in total at most $C \doteq O\left(k^2 d^2 \log\left(\frac{d \cdot ||A^\top A||_F}{\epsilon}\right)\right)$ regret. Thus, for the rest of our analysis, we can focus on rounds $t \in S_2$.

Fix any round $t \in S_2$. From Lemma 4 we have:.

$$\langle \hat{r}, \pi(\hat{r}, \bar{d})\rangle - \langle \hat{r}, \pi(\hat{r}, \hat{d})\rangle \leq k^3 \epsilon$$

Further manipulations give:

$$\begin{aligned}
&\big(\langle \hat{r}, \pi(\hat{r}, \bar{d})\rangle - \langle \bar{r}, \pi(\hat{r}, \bar{d})\rangle\big) - \big(\langle \hat{r}, \pi(\hat{r}, \hat{d})\rangle - \langle \bar{r}, \pi(\hat{r}, \hat{d})\rangle\big) \leq k^3\epsilon - \langle \bar{r}, \pi(\hat{r}, \bar{d})\rangle + \langle \bar{r}, \pi(\hat{r}, \hat{d})\rangle \\
\Leftrightarrow &\langle 2w, \pi(\hat{r}, \bar{d})\rangle - \langle 2w, \pi(\hat{r}, \hat{d})\rangle \leq k^3\epsilon - \langle \bar{r}, \pi(\hat{r}, \bar{d})\rangle + \langle \bar{r}, \pi(\hat{r}, \hat{d})\rangle \\
\Leftrightarrow &\langle 2w, \pi(\hat{r}, \bar{d})\rangle \leq \langle 2w, \pi(\hat{r}, \hat{d})\rangle + k^3\epsilon - \langle \bar{r}, \pi(\hat{r}, \bar{d})\rangle + \langle \bar{r}, \pi(\hat{r}, \hat{d})\rangle
\end{aligned}$$

Now, substituting the above expressions back into our expression for regret:

$$\begin{aligned}
&\textbf{Regret}(\boldsymbol{L}_{full}, T) \\
&\leq C + \sum_{t \in S_2} \langle 2w^t, \pi(\hat{r}^t, \bar{d}^t)\rangle + \langle \bar{r}^t, \pi(\hat{r}^t, \bar{d}^t)\rangle - \langle \bar{r}^t_i, \pi(\hat{r}^t, \hat{d}^t)\rangle \\
&\leq C + \sum_{t \in S_2} \langle 2w^t, \pi(\hat{r}^t, \hat{d}^t)\rangle + k^3\epsilon - \langle \bar{r}^t, \pi(\hat{r}^t, \bar{d}^t)\rangle \\
&+ \langle \bar{r}^t, \pi(\hat{r}^t, \hat{d}^t)\rangle + \langle \bar{r}^t, \pi(\hat{r}^t, \bar{d}^t)\rangle - \langle \bar{r}^t_i, \pi(\hat{r}^t, \hat{d}^t)\rangle \\
&\leq C + \sum_{t \in S_2} \langle 2w^t, \pi(\hat{r}^t, \hat{d}^t)\rangle + k^3\epsilon \\
&\leq C + 2\sum_{t \in S_2} \mathop{\mathbb{E}}_{i \in \pi(\hat{r}^t, \hat{d}^t)} [w^t_i] + k^3\epsilon \\
&\leq C + k^3\epsilon T + 2\left(\sqrt{2d \log\big(1 + \frac{T}{d\lambda}}\big)\left(\sqrt{2dT \log(\frac{1 + T/\lambda}{\delta}}) + \sqrt{T\lambda}\right) + \sqrt{2T \log\frac{1}{\delta}}\right) \\
&= O\left(k^2 d^2 \log\left(\frac{d \cdot ||A^\top A||_F}{\epsilon}\right)\right) + k^3\epsilon T + O\left(d\sqrt{T} \log\left(\frac{T}{\delta}\right)\right)
\end{aligned}$$

The last inequality holds with probability $1 - \delta$ and uses Lemmas 6 and 7, and sets $\lambda = 1$.

□

Finally, the bound on the fairness loss is identical to the bound we proved in Theorem 1 (because our algorithm for constructing distance estimates $\hat{d}$ is unchanged). We have:

Theorem 5. *For any sequence of contexts and any Mahalanobis distance* $d(x_1, x_2) =$

$||Ax_1 - Ax_2||_2$:

$$\mathbf{FairnessLoss}(\boldsymbol{L}_{full}, T, \epsilon) \leq O\left(k^2 d^2 \log\left(\frac{d \cdot ||A^\top A||_F}{\epsilon}\right)\right)$$

2.6. Discussion

We have initiated the study of fair sequential decision making in settings where the notions of payoff and fairness are separate and may be in tension with each other, and have shown that in a stylized setting, optimal fair decisions can be efficiently learned even without direct knowledge of the fairness metric. A number of extensions of our framework and results would be interesting to examine. At a high level, the interesting question is: how much can we further relax the information about the fairness metric available to the algorithm and the structure of the fairness metric itself?

For instance,

1. what if the fairness feedback is only partial, identifying some but not all fairness violations?
2. What if the fairness metric doesn't have a nice parametric form?
3. What if the feedback is not guaranteed to be exactly consistent with any metric?

In the next chapter, we in fact answer the above questions affirmatively in that we show how to approximately satisfy individual fairness even when the available information about the fairness metric is limited and there isn't much structure to the fairness metric as describe above.

Chapter 3

Individual Fairness via Auditing: No Assumption on the Fairness Metric

3.1. Introduction

In the previous chapter, we have provided an online learning algorithm that can eventually learn a Mahalanobis metric based on identified fairness violations, while achieving no-regret against the optimal fair policy. While the framework allows one to bypass the fact that it might be difficult for humans to enunciate a precise quantitative similarity measure between individuals, it still faces some limitations. In particular, people may not be able to distill their conception of fairness into a Mahalanobis metric function, let alone any metric (e.g. it may not satisfy the triangle inequality).

To tackle these issues, we study *metric-free* online learning algorithms for individual fairness that rely on a weaker form of interactive human feedback and minimal assumptions on the fairness metric across individuals. Similar to what we have shown in Chapter 2, we do not assume a pre-specified metric but instead assume access to an *auditor* who observes the learner's decisions over a group of individuals that show up in each round and attempts to identify a fairness violation—a pair of individuals in the group that should have been treated more similarly by the learner. Since the auditor only needs to identify such unfairly treated pairs, there is no need for them to enunciate a quantitative measure — to specify the distance between the identified pairs. But more importantly, we do not impose *any parametric assumption* on the underlying fairness measure, nor do we assume that it is actually a metric since we do not require that fairness measure to satisfy the triangle inequality. Furthermore, we do not require the auditor to identify all pairs where there was a fairness violation but rather one arbitrary pair with violation if there exists one.

Under this model, we provide a general reduction framework that can take any online classification algorithm (without fairness constraint) as a black-box and obtain a learning algorithm that can bound the sum of the regret with respect to classification loss and the total number of rounds with fairness violations. In fact, by setting some trade-off parameter that balances the fairness loss and misclassification loss, we provide an oracle-efficient algorithm that can in fact guarantee that both of them will be sublinear simultaneously.

Essentially, we resolve main questions left open from Chapter 2. First, we assume a weaker auditor who only identifies a single fairness violation (as opposed to all of the fairness violations in their setting). Second, we remove the strong parametric assumption on the Mahalanobis metric and work with a broad class of functions that need not be metric.

3.1.1. Overview of Model and Results

In each round $t = 1, \ldots, T$, a batch of k individuals arrives along with their contexts and binary labels as opposed to rewards. Unlike before, we make no assumption regarding how this batch of contexts and labels are determined — they can arrive in an adaptive and adversarial fashion. At every round, the learner gets to deploy some policy π that outputs a soft-prediction $\pi(x) \in [0,1]$ for each context x. For each context and its label (x, y), there is an associated misclassification loss for predicting $\pi(x)$.

Similarly as before, we wish to make these soft predictions for k individuals such that for any two individuals, the difference between their soft-predictions is at most the distance of their distance. Although the learner has no knowledge of this distance initially, the learner has access to an auditor who upon seeing these predictions made for k individuals will spot a pair of individuals where the difference in the predictions is more than than distance. However, we make no structural assumption about the distance function that the auditor is working off of and also have the auditor to output only one such pair even if there exists many such pairs. Similarly as before, our goal is to design an algorithm such that the cumulative misclassification regret is competitive with respect to any fair policy and the total number of rounds on which the auditor finds a pair with fairness violation.

To do so, we design a hybrid loss that combines the misclassification loss and the fairness violation into account and show that playing no-regret with respect to this new hybrid loss is sufficient to have the sum of the misclassification regret and fairness loss will be sublinear.

Then, we provide a reduction-based algorithm that can take any no-regret online classification learner as a black-box and guarantee the sum of the regret and the fairness loss (i.e. the number of rounds with any fairness violation) will be sublinear by re-expressing the hybrid loss solely as misclassification loss on some artificially created batch of contexts and labels.

Our reduction-based algorithm can take any no-regret online (batch) classification learner as a black-box and achieve sublinear cumulative fairness loss and sublinear regret on misclassification loss compared to the most accurate policy that is fair on every round. In particular, our framework can leverage the generic exponential weights method [5, 11, 32] and also oracle-efficient methods, including variants of Follow-the-Perturbed-Leader (FTPL) (e.g., [89, 90]), that further reduces online learning to standard supervised learning or optimization problems. We instantiate our framework using two online learning algorithms (exponential weights and CONTEXT-FTPL), both of which obtain a $O(\sqrt{T})$ regret. By setting the parameter that balances the misclassification loss and the fairness violation in the hybrid loss just right so that without any additional argument, we show how the overall misclassification regret with respect to fair policies and fairness loss can be bounded to be sublinear simultaneously.

With some modifications to the batch-to-online conversion techniques, it can be shown that fairness loss and misclassification loss also generalizes to the averaged policy; for more interested readers, see Bechavod et al. [8], which this chapter is based off of.

3.2. Preliminaries

Here we try to stay close to the notations used in Chapter 2 but use some slightly different notations in some cases. We define the instance space to be $\mathcal{X}$ and its label space to be $\mathcal{Y}$. Throughout this chapter, we will restrict our attention to binary labels, that is $\mathcal{Y} = \{0, 1\}$. We write $\mathcal{F} : \mathcal{X} \to \mathcal{Y}$ to denote the hypothesis class and assume that $\mathcal{F}$ contains a constant hypothesis — i.e. there exists f such that $f(x) = 0$ (and/or 1) for all $x \in \mathcal{X}$. Also, we allow a convex combination of hypotheses for the purpose of randomizing the prediction and denote the simplex of hypotheses by $\Delta\mathcal{F}$. For consistency, we say $f \in \mathcal{F}$, which maps to $\mathcal{Y} = \{0, 1\}$, is a *hypothesis* and $\pi \in \Delta\mathcal{F}$, which is a mixture of some hypotheses and hence maps to $[0, 1]$, a *policy*. For each prediction $\hat{y} \in \mathcal{Y}$ and its true label $y \in \mathcal{Y}$, there is an associated misclassification loss, $\ell(\hat{y}, y) = \mathbb{1}[\hat{y} \neq y]$. For simplicity, we overload the notation and write

$$\ell(\pi(x), y) = (1 - \pi(x)) \cdot y + \pi(x) \cdot (1 - y) = \mathop{\mathbb{E}}_{f \sim \pi}[\ell(f(x), y)].$$

Note that the loss is linear in π. Given π_1 and π_2 and some mixing probability p, define $\pi_3(x) = p\pi_1(x) + (1 - p)(\pi_2(x))$. Then, it is immediate that

$$\ell(\pi_3(x), y) = p\ell(\pi_1(x), y) + (1 - p)\ell(\pi_2(x), y).$$

3.2.1. Online Batch Classification

Here, we describe the vanilla online batch classification setting, as we will try to reduce the problem we are interested into this setting. In each round $t = 1, \dots, T$, a learner deploys some policy $\pi^t \in \Delta\mathcal{F}$. Upon seeing the deployed policy π^t, the environment chooses a batch of k individuals, $(x_i^t, y_i^t)_{i=1}^k$. Because the environment can adversarially and adaptively choose this batch of individuals, we can think of this as the environment strategically choosing $z^t_{\text{BATCH}} = (x^t, y^t)$ where we write $x^t = (x_i^t)_{i=1}^k$ and $y^t = (y_i^t)_{i=1}^k$ for simplicity. Note that the strategy space for the environment is

$$\mathcal{Z}^t_{\text{BATCH}} = \mathcal{X}^k \times \mathcal{Y}^k.$$

The history in each round then will consist of everything observed by the learner up through but not including round t:

$$h^t = ((\pi^1, z^1), \dots, (\pi^{t-1}, z^{t-1})).$$

The space of such histories is denoted by $\mathcal{H}^t_{\text{BATCH}} = (\Delta\mathcal{F} \times \mathcal{Z}_{\text{BATCH}})^{t-1}$. A learner $\boldsymbol{A} : \mathcal{H}^*_{\text{BATCH}} \to \Delta\mathcal{F}$ is then defined to be a mapping from history to a policy:

$$\pi^t = \boldsymbol{A}(h^{t-1}).$$

Given some loss that be calculated in each round t, the learner cannot hope to upper-bound the overall loss by itself over all the rounds $t = 1, \ldots, T$ will be small due to the adversarial nature of the environment. Therefore, the learner can only hope to minimize its regret with respect to some fixed policy it could have used.

Definition 7. *Fix the adversary's strategy space $\mathcal{Z}$. With respect to some baseline $Q \subseteq \Delta\mathcal{F}$ and some loss $L : \Delta\mathcal{F} \times \mathcal{Z} \to \mathbb{R}$, we say learner $\boldsymbol{A}$'s regret with respect to adaptively and adversarially chosen sequence of $(z^t)_{t=1}^T$ is*

$$\sum_{t=1}^{T} L\left(\pi^t, z^t\right) - \min_{\pi^* \in Q} \sum_{t=1}^{T} L\left(\pi^*, z^t\right).$$

In this vanilla online batch classification setting, the only loss that we care about is the misclassification loss:

Definition 8 (Misclassification Loss)**.** *The (batch) misclassification loss* **Err** *is*

$$\mathbf{Err}(\pi, z^t) = \sum_{i=1}^{k} \ell(\pi(x_i^t), y_i^t).$$

In other words, the goal in this setting is to come up with an learner $\boldsymbol{A}$ such that against any adaptively and adversarially chosen $(z^t)_{t=1}^T$, we can achieve

$$\sum_{t=1}^{T} \mathbf{Err}\left(\pi^t, z^t\right) - \min_{\pi^* \in Q} \sum_{t=1}^{T} \mathbf{Err}\left(\pi^*, z^t\right) = o(T).$$

Often, when learner $\boldsymbol{A}$ can guarantee that the regret is sublienar as above, it is said to be a no-regret learner. Examples of such no-regret learners include exponential weights [5, 11, 32] and Follow-the-Perturbed-Leader strategies [89, 90].

3.2.2. Online Fair Batch Classification

As opposed to only trying to minimize its misclassification regret, we also want to make sure that the deployed policies $(\pi^t)_{t=1}^T$ satisfy individual fairness constraints (i.e. each policy π^t treats similar individuals similarly according to some fairness metric $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$) as in Chapter 2.

Definition 9 (α-fairness)**.** *Assume $\alpha > 0$. A policy $\pi \in \Delta\mathcal{F}$ is said to be α-fair on pair (x, x'), if*

$$|\pi(x) - \pi(x')| \leq d(x, x') + \alpha.$$

We say policy π's α-fairness violation on pair (x, x') is

$$\textbf{Violation}_\alpha(\pi, (x, x')) = \max(0, |\pi(x) - \pi(x')| - d(x, x') - \alpha).$$

In order to find such fair policies, we once again rely on an auditor who can give feedback by pointing out when a pair of two similar individuals are not treated similarly according to metric d. However, in this chapter, we make no parametric assumption on the metric d nor do we require it to be a proper metric (i.e. it doesn't need to satisfy the triangle inequality). The only requirement on d is that it is always non-negative. In addition to removing the parametric assumption on the metric d, we further relax the assumption on how the auditor operates: the auditor only need output one arbitrary pair where fairness violation has occurred as opposed to reporting all violations as in Chapter 2.

Definition 10 (Auditor)**.** *An auditor $O_{d,\alpha}$, which can have its own internal state, takes in a* reference set *$S \subseteq \mathcal{X}$ and a policy π. Then, it outputs ρ which is either null or a pair of indices from the provided reference set to denote that there is some positive α-fairness* *violation for that pair. For some $x = (x_1, \dots x_k)$,*

$$O_{d,\alpha}(x, \pi) = \begin{cases} \rho = (\rho_1, \rho_2) & \textit{if} \quad \exists \rho_1, \rho_2 \in [k]. \pi(x_{\rho_1}) - \pi(x_{\rho_2}) > d(x_{\rho_1}, x_{\rho_2}) + \alpha \\ \textit{null} & \textit{otherwise} \end{cases}$$

If there exists multiple pairs with some α-violation, the auditor can choose one arbitrarily. We will elide d and write O_α, as we will only focus on the case where d is fixed.

Remark 3. *We emphasize that the assumptions on the auditor here are much more relaxed than those of Chapter 2, which require that the auditor outputs whether the policy is 0-fair (i.e. with no slack) on all pairs exactly. Furthermore, the auditor in Chapter 2 can only handle Mahalanobis distances. In our setting, because of the internal state of the auditor, the auditor does not have to be a fixed function but rather can be adaptively changing in each round. Our argument actually never relies on the fact the distance function d stays the same throughout rounds, meaning all our results extend to the case where the distance function governing the fairness constraints is changing every round. For simplicity, we focus on the case where d is fixed.*

The order in which the learner, the environment, and the auditor interact is as follows. In each round $t = 1, \dots, T$, a learner deploys a policy $\pi^t \in \Delta\mathcal{F}$. Then, a batch of k individuals $(x^t, y^t) = ((x_i^t)_{i\in[k]}, (y_i^t)_{i\in[k]})$ arrives. The auditor upon inspecting (π^t, x^t) provides a fairness feedback $\rho^t \in [k]^2 \cup \{null\}$ which may be a pair of indices for which the deployed policy π^t is treating unfairly or null if there doesn't exist any such pair. Because the auditor can essentially choose the pair arbitrarily among all such pairs, we can actually fold the auditor into the environment. That is – the environment choose $z^t_{\text{FAIR-BATCH}} = (x^t, y^t, \rho^t)$ simultaneously, meaning the strategy space of the environment here is

$$\mathcal{Z}_{\text{FAIR-BATCH}} = (\mathcal{X}^k \times \mathcal{Y}^k \times ([k]^2 \cup \{null\})).$$

Similarly as in the vanilla online batch classification setting, the history is now described as

$$h^t = ((\pi^1, z^1), \dots, (\pi^{t-1}, z^{t-1})).$$

where $z^t \in \mathcal{Z}_{\text{FAIR-BATCH}}$ and the space of such histories is then $\mathcal{H}^t_{\text{FAIR-BATCH}} = (\Delta\mathcal{F} \times \mathcal{Z}_{\text{FAIR-BATCH}})^{t-1}$. A learner $\boldsymbol{A}$ is still defined to be a mapping from history to a policy:

$$\pi^t = \boldsymbol{A}(h^{t-1}).$$

In addition to the misclassification loss **Err**, the learner also has to worry about is the fairness loss.

Definition 11 (Fairness Loss)**.** *The α-fairness loss* $\mathbf{Unfair}_\alpha$ *is*

$$\mathbf{Unfair}_\alpha(\pi, z^t) = \begin{cases} \mathbb{1}\left[\mathbf{Violation}_\alpha(\pi, (x^t_{\rho^t_1}, x^t_{\rho^t_2})) > 0\right] & \text{if } \rho^t = (\rho^t_1, \rho^t_2) \\ 0 & \text{otherwise} \end{cases}$$

In other words, the learner will incur misclassification loss $\mathbf{Err}(\pi^t, z^t)$[5] and fairness loss $\mathbf{Unfair}(\pi^t, z^t)$ in each round t. Note that unlike the fairness loss defined in Chapter 2 which counts the total number of pairs with fairness violations, the fairness loss is either 0 or 1 depending on the existence of such pair in this setting. We try to compare this online batch classification setting with fairness constraints to the vanilla online batch classification setting in Figure 2.

Finally, the baseline Q_α that we compete against will be all policies that are α-fair on x^t for all $t \in [T]$:

$$Q_\alpha = \{\pi \in \Delta\mathcal{F} : \pi \text{ is } \alpha\text{-fair on } x^t \text{ for all } t \in [T]\}.$$

Given some fixed trade-off slack $\epsilon \in (0, \alpha)$ an auditor O_α, our goal is to provide a learner

[5] We overload the notation for **Err** and write $\mathbf{Err}(\pi, ((x, y), \rho)) = \mathbf{Err}(\pi, (x, y))$.

Algorithm 1 Online Fair Batch Classification FAIR-BATCH

```
for $t = 1, \dots, T$ do
    Learner deploys $\pi^t$
    Environment chooses $(x^t, y^t)$
    Environment chooses the pair $\rho^t$
    $z^t = ((x^t, y^t), \rho^t)$
    Learner incurs misclassfication loss $\mathbf{Err}(\pi^t, z^t)$
    Learner incurs fairness loss $\mathbf{Unfair}(\pi^t, z^t)$
end
```

Algorithm 2 Online Batch Classification BATCH

```
for $t = 1, \dots, T$ do
    Learner deploys $\pi^t$
    Environment chooses $z^t = (x^t, y^t)$
    Learner incurs misclassification loss $\mathbf{Err}(\pi^t, z^t)$
end
```

Figure 2: Comparison between Online Fair Batch Classification and Online Batch Classification: each is summarized by the interaction between the learner and the environment: $(\Delta\mathcal{F} \times \mathcal{Z}_{\text{FAIR-BATCH}})^T$ and $(\Delta\mathcal{F} \times \mathcal{Z}_{\text{BATCH}})^T$ where $\mathcal{Z}_{\text{FAIR-BATCH}} = \mathcal{X}^k \times \mathcal{Y}^k \times ([k]^2 \cup \{null\})$ and $\mathcal{Z}_{\text{BATCH}} = \mathcal{X}^k \times \mathcal{Y}^k$.

$\boldsymbol{A}$ such that for any adversarially and adaptively chosen $((x^t, y^t))_{t=1}^T$, regret with respect to misclassification and fairness loss is sublinear against $Q_{\alpha-\epsilon}$:

$$\sum_{t=1}^T \mathbf{Err}\left(\pi^t, z^t\right) - \min_{\pi^* \in Q_{\alpha-\epsilon}} \sum_{t=1}^T \mathbf{Err}\left(\pi^*, z^t\right) = o(T)$$

$$\sum_{t=1}^T \mathbf{Unfair}_\alpha(\pi^t, z^t) - \min_{\pi^* \in Q_{\alpha-\epsilon}} \sum_{t=1}^T \mathbf{Unfair}_\alpha(\pi^*, z^t) = \sum_{t=1}^T \mathbf{Unfair}_\alpha(\pi^t, z^t) = o(T)$$

where $z^t = (x^t, y^t, O_\alpha(x^t, \pi^t))$. Because $\pi^* \in Q_{\alpha-\epsilon}$ must be α-fair, guaranteeing sublinear fairness regret is equivalent to guaranteeing the overall fairness loss is sublinear.

3.3. Related Work

We refer the readers to the related work section of Chapter 2 (Section 2.2) in terms of related work in the algorithmic fairness literature to the work presented in this chapter: this chapter is essentially a generalization of the setting consider in Chapter 2.

Our technique of combining the loss and the constraint violation into a Lagrangian loss is very related to the technique used in the online convex optimization with long term constraints [10, 48, 72, 97, 98]. Similarly as in our setting, they are interested in choosing some point x^t in each round and incur some loss $f^t(x^t)$ where f^t is chosen by the adversary.

Given some collection of constraints $\{g_i\}_{i=1}^m$, the goal is to simultaneously ensure the total violation of the constraints $\sum_{t=1}^t \sum_{i=1}^m g_i(x^t)$ is sublinear and to achieve sublinear regret against any fixed point x^* in hindsight that satisfies all the constraints — $\sum_{i=1}^m g_i(x^*) \leq 0$. However, in most of these settings, the constraints are initially known to the learner, and the algorithm requires a projection into the feasible region. This is different than our setting where we have no idea what the space of fair policies look like, as the auditor cannot enunciate what the fairness metric is. The only exception among the works cited above is the work of Cao and Liu [10] and Mahdavi et al. [72]. They consider a bandit feedback setting where the constraints are not known initially and only the max violation of the constraints with respect to the chosen point is revealed in each round — i.e. $\max_{i\in[m]} g_i(x^t)$. Our setting is still different in that we do not receive the amount of violation in each round but only the point's feasibility (i.e. $\mathbb{1}(\max_{i\in[m]} g_i(x^t) \leq 0)$) and one of the violated constraints chosen arbitrarily. Furthermore, in their setting, the the point chosen in each round x^t is a d-dimesnional vector (i.e. $x^t \in \mathbb{R}^d$), whereas we imagine the policy chosen in each round comes from the simplex of some hypothesis class $\Delta\mathcal{F}$, which is often much larger than $\mathbb{R}^d$. In that sense, our use of the Follow-The-Perturbed-Leader approach to make the overall algorithm oracle-efficient is novel.

3.4. Lagrangian Regret

Because we wish to achieve no-regret with respect to both the misclassification and fairness loss, it is natural to consider a hybrid loss that combines them together. In fact, we define a round-based Lagrangian loss and show that the regret with respect to our Lagrangian loss also serves as an upperbound for the misclassification and the fairness complaint regret multipled by some parameter that balances the misclassification and fairness loss in the Lagrangian loss.

Then, we show how to achieve no regret with respect to the Lagrangian loss by reducing the problem to an online batch classification where there is no fairness constraint. For concreteness, we show how to leverage exponential weights in order to achieve sublinear misclassification regret and fairness loss.

3.4.1. Lagrangian Formulation

Here we present a hybrid loss that we call *Lagrangian loss* that combines the misclassification loss and the fairness violation in round t.

Definition 12 (Lagrangian Loss)**.** *C-Lagrangian loss of π is*

$$\mathcal{L}_C\left(\pi, \left((x^t, y^t), \rho^t\right)\right) = \sum_{i=1}^{k} \ell\left(\pi\left(x_i^t\right), y_i^t\right) + \begin{cases} C\left(\pi(x_{\rho_1}^t) - \pi(x_{\rho_2}^t)\right) & \rho^t = (\rho_1, \rho_2) \\ 0 & \rho^t = null \end{cases}$$

Given an auditor O_α that can detect any α-fairness violation, we can simulate the online fair batch classification setting with an auditor O_α by setting the pair $\rho^t_{O_\alpha} = O_\alpha(x^t, \pi^t)$.

Now, we show that the Lagrangian regret serves as an upper bound for the sum of the α-fairness loss (with some multiplicative factor that depends on C and ϵ) and the misclassification loss regret with respect to $Q_{\alpha-\epsilon}$.

Theorem 6. *Given an auditor O_α, fix any sequence $((x^t, y^t))_{t=1}^T$, $(\pi^t)_{t=1}^T$, and $\rho^t_{O_\alpha} = O_\alpha(x^t, \pi^t)$ for each $t \in [T]$. Then, the following holds for any $\epsilon \in (0, \alpha]$ and $\pi^* \in Q_{\alpha-\epsilon}$ simultaneously:*

$$\begin{aligned} &C\epsilon \sum_{t=1}^{T} \mathbf{Unfair}_\alpha(\pi^t, z^t) + \left(\sum_{t=1}^{T} \mathbf{Err}(\pi^t, z^t) - \sum_{t=1}^{T} \mathbf{Err}(\pi^*, z^t)\right) \\ &\leq \sum_{t=1}^{T} \mathcal{L}_C(\pi^t, z^t) - \sum_{t=1}^{T} \mathcal{L}_C(\pi^*, z^t) \end{aligned}$$

where $z^t = ((x^t, y^t), \rho^t_{O_\alpha})$ for each $t \in [T]$.

Proof. Fix $\epsilon \in [0, \alpha]$. Fix any $(\alpha - \epsilon)$-fair policy $\pi^* \in Q_{\alpha-\epsilon}$. Note that for any round t where

$\rho^t_{O_\alpha} \neq null$, we have

$$\begin{aligned}
\pi^*(x^t_{\rho^t_1}) - \pi^*(x^t_{\rho^t_2}) \leq d(x^t_{\rho^t_1}, x^t_{\rho^t_2}) + \alpha - \epsilon \quad &\Rightarrow \quad -\left(\pi^*(x^t_{\rho^t_1}) - \pi^*(x^t_{\rho^t_2}) - d(x^t_{\rho^t_1}, x^t_{\rho^t_2}) - \alpha\right) \geq \epsilon \\
\pi^t(x^t_{\rho^t_1}) - \pi^t(x^t_{\rho^t_2}) > d(x^t_{\rho^t_1}, x^t_{\rho^t_2}) + \alpha \quad &\Rightarrow \quad \pi^t(x^t_{\rho^t_1}) - \pi^t(x^t_{\rho^t_2}) - d(x^t_{\rho^t_1}, x^t_{\rho^t_2}) - \alpha > 0
\end{aligned}$$

because π^* is $(\alpha - \epsilon)$-fair on this pair and π^t wasn't α-fair on $(x^t_{\rho^t_1}, x^t_{\rho^t_2})$.

Then we can show

$$\begin{aligned}
&\sum_{t=1}^{T} \mathcal{L}_C(\pi^t, z^t) - \mathcal{L}_C(\pi^*, z^t) \\
&= \sum_{t=1}^{T}\sum_{i=1}^{k} \ell\left(\pi^t\left(x^t_i\right), y^t_i\right) - \ell\left(\pi^*\left(x^t_i\right), y^t_i\right) + \sum_{t\in[T]:\rho^t_{O_\alpha}\neq null} C\left(\pi^t(x^t_{\rho^t_1}) - \pi^t(x^t_{\rho^t_2})\right) \\
&- C\left(\pi^*(x^t_{\rho^t_1}) - \pi^*(x^t_{\rho^t_2})\right) \\
&= \sum_{t=1}^{T}\sum_{i=1}^{k} \ell\left(\pi^t\left(x^t_i\right), y^t_i\right) - \ell\left(\pi^*\left(x^t_i\right), y^t_i\right) \\
&+ \sum_{t\in[T]:\rho^t_{O_\alpha}\neq null} C\left(\pi^t(x^t_{\rho^t_1}) - \pi^t(x^t_{\rho^t_2}) - d(x^t_{\rho^t_1}, x^t_{\rho^t_2}) - \alpha\right) \\
&- C\left(\pi^*(x^t_{\rho^t_1}) - \pi^*(x^t_{\rho^t_2}) - d(x^t_{\rho^t_1}, x^t_{\rho^t_2}) - \alpha\right) \\
&\geq \sum_{t=1}^{T}\sum_{i=1}^{k} \ell\left(\pi^t\left(x^t_i\right), y^t_i\right) - \ell\left(\pi^*\left(x^t_i\right), y^t_i\right) + \sum_{t\in[T]:\rho^t_{O_\alpha}\neq null} C\epsilon \\
&= \sum_{t=1}^{T}\sum_{i=1}^{k} \ell\left(\pi^t\left(x^t_i\right), y^t_i\right) - \ell\left(\pi^*\left(x^t_i\right), y^t_i\right) + C\epsilon\sum_{t=1}^{T} \mathbf{Unfair}_\alpha(\pi^t, ((x^t, y^t), \rho^t_{O_\alpha}))
\end{aligned}$$

□

By considering the above theorem statement with $\epsilon = 0$, we can always bound the misclassification regret with the Lagrangian regret:

$$\begin{aligned}\max_{\pi^* \in Q_\alpha} \left(\sum_{t=1}^{T} \mathbf{Err}(\pi^t, z^t) - \sum_{t=1}^{T} \mathbf{Err}(\pi^*, z^t) \right) &\leq \max_{\pi^* \in Q_\alpha} \sum_{t=1}^{T} \mathcal{L}_C(\pi^t, z^t) - \sum_{t=1}^{T} \mathcal{L}_C(\pi^*, z^t) \\ &\leq \max_{f^* \in \mathcal{F}} \sum_{t=1}^{T} \mathcal{L}_C(\pi^t, z^t) - \sum_{t=1}^{T} \mathcal{L}_C(\pi^*, z^t)\end{aligned}$$

However, note that we cannot always hope to bound the fairness loss with the Lagrangian regret because the misclassification regret may be negative. It alludes to the fact that it is not sufficient to simply come up with a way to bound the Lagrangian regret to be sublinear in T. In fact, we have to tune C accordingly to what kind of Lagrangian regret guarantee we can get: we want to want to set C high enough so that we give more emphasis to the fairness loss, but also we cannot set it too high because C controls the bound of the Lagrangian loss and the regret guarantees usually has a linear dependence on the bound of the loss. In the next section, we will show exactly how to set C so that we can achieve no regret with respect to fairness and misclassification loss simultaneously.

3.5. Achieving No Regret Simultaneously

In Section 3.5.1, we show an efficient reduction from the setting of online batch classification *with* fairness constraints to that of online batch classification *without* any constraints. This reduction to the online batch classification without the fairness constraints is not necessary, as the Lagrangian loss is already linear in the first argument, π. However, we go through this reduction, as it's more obvious that no-regret algorithms can be directly used for this setting without fairness constraints.

Then, combining our reduction to the online batch classification without fairness constraints, exponential weights approach, and C that is appropriately set with respect to the regret guarantee of exponential weights, we show how to bound the misclassification regret and fairness loss with $O(T^{\frac{3}{4}})$ in Section 3.5.2.

Finally, in Section 3.5.3, we show how we can use the Follow-The-Perturbed-Leader approach, specifically that of Syrgkanis et al. [90], can be used to make the algorithm run in an oracle-efficient manner.

3.5.1. Reduction to Online Batch Classification without Fairness Constraints

Here we show how to reduce the online batch fair classification problem to the online batch classification problem in an efficient manner. Once we can do this reduction, then we can any online batch algorithm $\boldsymbol{A}_{\text{BATCH}}((\pi^\tau, (x'^\tau, y'^\tau))_{\tau=1}^t)$ as a black box in order to achieve sublinear Lagrangian regret. At a high level, our reduction involves just carefully transforming our online fair batch classification history up to t, $(\pi^\tau, ((x^\tau, y^\tau), \rho_O^\tau))_{\tau=1}^t \in (\Delta\mathcal{F} \times \mathcal{Z}_{\text{FAIR-BATCH}})^t$ into some fake online batch classification history $(\pi^\tau, (x'^\tau, y'^\tau))_{\tau=1}^t \in (\Delta\mathcal{F} \times \mathcal{Z}_{\text{BATCH}})^t$ and then feeding the artificially created history to a no-regret learner $\boldsymbol{A}_{\text{BATCH}}$ for the online batch classification setting.

Without loss of generality, we assume that C is an integer; if it's not, then take the ceiling. Now, we describe how the transformation of the history works. For each round t, whenever $\rho^t = (\rho_1^t, \rho_2^t)$, we add C copies of each of $(x_{\rho_1^t}^t, 0)$ and $(x_{\rho_2^t}^t, 1)$ to the original pairs to form x'^t and y'^t. Just to keep the batch size the same across each round, even if $\rho^t = null$, we add C copies of each of $(v, 0)$ and $(v, 1)$ where v is some arbitrary instance in $\mathcal{X}$. We describe this process in more detail in Algorithm 3.

This reduction essentially preserves the regret, which we formally state in Lemma 8.

Lemma 8. *For any sequence of $(\pi^t)_{t=1}^T$, $((x^t, y^t))_{t=1}^T$, $(\rho^t)_{t=1}^T$, and $\pi^* \in \Delta\mathcal{F}$, we have*

$$\sum_{t=1}^T \mathcal{L}_C(\pi^t, z^t) - \sum_{t=1}^T \mathcal{L}_C(\pi^*, z^t) = \sum_{t=1}^T \mathbf{Err}(\pi^t, (x'^t, y'^t)) - \sum_{t=1}^T \mathbf{Err}(\pi^*, (x'^t, y'^t))$$

where $z^t = ((x^t, y^t), \rho^t)$ and $(x'^t, y'^t) = R_C((x^t, y^t), \rho^t)$.

Proof. It is sufficient to show that in each round $t \in [T]$,

$$\mathcal{L}_{C,\alpha}(\pi^t, z^t) - \mathcal{L}_{C,\alpha}(\pi^*, z^t) = \sum_{i=1}^{k+2C} \ell(\pi^t(x_i^t), y_i^t) - \sum_{i=1}^{k+2C} \ell(\pi^*(x_i^t), y_i^t)$$

Algorithm 3 Reduction from Online Fair Batch Classification to Online Batch Classification, $R_C((x^t, y^t), \rho^t)$

```
Parameters: $C$
Input: $(x^t, y^t), \rho^t$
if $\rho^t = (\rho_1^t, \rho_2^t)$ then
    for $i = 1, \dots, C$ do
        $x_{k+i}^t = x_{\rho_1^t}^t$ and $y_{k+i}^t = 0$;
        $x_{k+C+i}^t = x_{\rho_2^t}^t$ and $y_{k+C+i}^t = 1$;
    end
end
else
    for $i = 1, \dots, C$ do
        $x_{k+i}^t = v$ and $y_{k+i}^t = 0$;
        $x_{k+C+i}^t = v$ and $y_{k+C+i}^t = 1$;
    end
end
$x'^t = (x_i^t)_{i=1}^{k+2C}$ and $y'^t = (y_i^t)_{i=1}^{k+2C}$
Output: $(x'^t, y'^t)$
```

First, assume $\rho^t = (\rho_1^t, \rho_2^t)$.

$$
\begin{aligned}
&\mathcal{L}_C(\pi^t, z^t) - \mathcal{L}_C(\pi^*, z^t) \\
&= \left(\sum_{i=1}^{k} \ell(\pi^t(x_i^t), y_i^t) + C(\pi^t(x_{\rho_1^t}^t) - \pi^t(x_{\rho_2^t}^t)) \right) - \left(\sum_{i=1}^{k} \ell(\pi^*(x_i^t), y_i^t) + C(\pi^*(x_{\rho_1^t}^t) - \pi^*(x_{\rho_2^t}^t)) \right) \\
&= \left(\sum_{i=1}^{k} \ell(\pi^t(x_i^t), y_i^t) + \left(\sum_{i=1}^{C} \ell(\pi^t(x_{\rho_1^t}^t), 0) + \sum_{i=1}^{C} \ell(\pi^t(x_{\rho_2^t}^t), 1) - C \right) \right) \\
&- \left(\sum_{i=1}^{k} \ell(\pi^*(x_i^t), y_i^t) + \left(\sum_{i=1}^{C} \ell(\pi^*(x_{\rho_1^t}^t), 0) + \sum_{i=1}^{C} \ell(\pi^*(x_{\rho_2^t}^t), 1) - C \right) \right) \\
&= \sum_{i=1}^{k+2C} \ell(\pi^t(x_i'^t), y_i'^t) - \sum_{i=1}^{k+2C} \ell(\pi^*({x'}^t_{\,i}), y_i'^t),
\end{aligned}
$$

The second equality follows from the fact that for any π and x,

$$
\ell(\pi(x), 0) = \pi(x) \quad \text{and} \quad \ell(\pi(x), 1) = 1 - \pi(x).
$$

If $\rho^t = null$, then the same argument applies as above; the only difference is that all the $\pi^t(v)$ cancel with each other because the number of copies with label 0 is exactly the same as that of label 1. □

3.5.2. Exponential Weights

It is well known that for linear loss, exponential weights with appropriately tuned learning rate γ can achieve no regret [5, 11, 32]. Let us first describe the setting of the exponential weights so that we can how that setting contains the online batch classification that we are interested in.

For each round $t = 1, \ldots, T$

1. The learner chooses a distribution $p^t = (p^t_1, \ldots, p^t_N)$ over N experts
2. The adversary, *with the full knowledge of* p^t, chooses $m^t = (m^t_1, \ldots, m^t_N)$ where $m^t_i \in [-B, B]$ for each $i \in [N]$.
3. The learner suffers $p^t \cdot m^t$.

We emphasize that because adversary gets to choose m^t with the full knowledge of p^t, m^t can be chosen as a function of p^t.

Exponential weights is defined as $p^t = \frac{1}{N}$ for $t = 1$ and for any $t \geq 2$

$$\hat{p}^{t+1}_i = (1 - \gamma m^t_i)p^t_i$$

and

$$p^{t+1} = \frac{\hat{p}^{t+1}_i}{\sum_{i \in [N]} \hat{p}^{t+1}_i}.$$

Then, we have the following guarantee on the regret of exponential weights:

Theorem 7 (Arora et al. [5]). *Exponential weights with learning rate γ has the following* *guarantee: for any sequence of* $(m^t)_{t=1}^T$ *and any* $i \in [N]$,

$$\sum_{t=1}^{T} p^t \cdot m^t \leq \sum_{t=1} m_i^t + B\left(\gamma T + \frac{\ln(|\mathcal{F}|)}{\gamma}\right).$$

In other words, with learning rate $\gamma = \sqrt{\frac{\ln(N)}{T}}$, *the regret is* $2B\sqrt{\ln(N)T}$.

We can easily reduce the online classification setting to that of exponential weights method. Each π^t that we deploy can be represented as a probability distribution $p^t = (p_f^t)_{f \in \mathcal{F}}$ over each $f \in \mathcal{F}$: for any $x \in \mathcal{X}$,

$$\pi^t(x) = \sum_{f \in \mathcal{F}} p_f^t f(x).$$

If we use $m_f^t = \mathbf{Err}(f, (x'^t, y'^t))$ for every $f \in \mathcal{F}$. Then, we have for any $(x'^t, y'^t) \in (\mathcal{X}^{k+2C} \times \mathcal{Y}^{k+2C})$,

$$\mathbf{Err}(\pi^t(x), x'^t, y'^t) = \sum_{f \in \mathcal{F}} p_f^t \mathbf{Err}(f, (x'^t, y'^t)) = p^t \cdot m^t.$$

Putting everything together, we have

1. The learner deploys π^t where the associated probability distribution over $\mathcal{F}$ is $p^t = (p_f^t)_{f \in \mathcal{F}}$.

2. The adversary, with the knowledge of p^t, comes up with some (x'^t, y'^t) which determines $m^t = (m_f^t)_{f \in \mathcal{F}}$ where $m_f^t = \mathbf{Err}(f, (x'^t, y'^t))$. Remember that $(x'^t, y'^t) = R_C(x^t, y^t, O_\alpha(x^t, \pi^t))$ is a function of π^t.

3. Learner suffers

$$p^t \cdot m^t = \mathbf{Err}(\pi^t(x), x'^t, y'^t).$$

Note that $m^t_f \in [-(k+2C), k+2C]$ for any $f \in \mathcal{F}$. Although (x'^t, y'^t) is formed as a function of π^t, the exponential weights approach still allows the adversary to form m^t as a function of p^t or equivalently π^t. Therefore, we can use the regret guarantee of the exponential weights and appropriately set C to achieve sublinear fairness loss and misclassification regret.

Theorem 8. *If $C = \max(T^{\frac{1}{4}}, k)$, exponential weights guarantees the following: for any adaptively and adversarially chosen $((x^t, y^t))_{t=1}^T$, we have,*

$$\sum_{t=1}^{T} \mathbf{Unfair}_\alpha(\pi^t, z^t) \leq \frac{6}{\alpha}\sqrt{\ln(|\mathcal{F}|)T} + \frac{k}{\alpha}T^{\frac{3}{4}}$$

$$\sum_{t=1}^{T} \mathbf{Err}(\pi^t, z^t) - \min_{\pi^* \in Q_\alpha} \sum_{t=1}^{T} \mathbf{Err}(\pi^*, z^t) \leq 6\sqrt{\ln(|\mathcal{F}|)}T^{\frac{3}{4}}$$

where $z^t = ((x^t, y^t), \rho^t_{O_\alpha})$ and $\rho^t_{O_\alpha} = O(x^t, \pi^t)$. In other words, misclassification regret and fairness loss is both bounded by $O(T^{\frac{3}{4}})$.

Proof. First, we apply the regret guarantee of exponential weights. Theorem 7 gives us that

$$\sum_{t=1}^{T} \mathbf{Err}(\pi^t, (x'^t, y'^t)) - \min_{f^* \in \mathcal{F}} \sum_{t=1}^{T} \mathbf{Err}(f^*, (x'^t, y'^t)) \leq 2(k + 2C)\sqrt{\ln(|\mathcal{F}|)T}.$$

because $\mathbf{Err}(\pi^t, (x'^t, y'^t)) \in [-(k+2C), k+2C]$. Note that $(x'^t, y'^t) = R_C(x^t, y^t, O_\alpha(x^t, \pi^t))$ is a function of π^t, but as we emphasized before, the regret guarantee still holds.

Combining our previous lemmas and theorems, we have for any $((x^t, y^t))_{t=1}^T$ and hence

$z^t = (x^t, y^t, O_\alpha(x^t, \pi^t))$ for $t \in [T]$, we have

$$\begin{aligned}
&C\epsilon \sum_{t=1}^{T} \mathbf{Unfair}_\alpha(\pi^t, z^t) + \left(\sum_{t=1}^{T} \mathbf{Err}(\pi^t, z^t) - \min_{\pi^* \in Q_{\alpha-\epsilon}} \sum_{t=1}^{T} \mathbf{Err}(\pi^*, z^t) \right) \\
&\leq \sum_{t=1}^{T} \mathcal{L}_{C,\alpha}(\pi^t, z^t) - \min_{\pi^* \in Q_{\alpha-\epsilon}} \sum_{t=1}^{T} \mathcal{L}_{C,\alpha}(\pi^*, z^t) \\
&\leq \sum_{t=1}^{T} \mathcal{L}_{C,\alpha}(\pi^t, z^t) - \min_{\pi^* \in \Delta\mathcal{F}} \sum_{t=1}^{T} \mathcal{L}_{C,\alpha}(\pi^*, z^t) \\
&= \sum_{t=1}^{T} \mathbf{Err}(\pi^t, (x'^t, y'^t)) - \min_{\pi^* \in \Delta\mathcal{F}} \sum_{t=1}^{T} \mathbf{Err}(\pi^*, (x'^t, y'^t)) \\
&= \sum_{t=1}^{T} \mathbf{Err}(\pi^t, (x'^t, y'^t)) - \min_{f^* \in \mathcal{F}} \sum_{t=1}^{T} \mathbf{Err}(f^*, (x'^t, y'^t)) \\
&\leq 2(k + 2C)\sqrt{\ln(|\mathcal{F}|)T}.
\end{aligned}$$

The second line follows from Theorem 6, the third from the fact that $Q_{\alpha-\epsilon} \subseteq \Delta\mathcal{F}$, the fourth from Lemma 8, and the last line follows from the linearity of $\sum_{t=1}^{T} \mathbf{Err}(\cdot, (x'^t, y'^t))$. The above inequality holds simultaneously for all $\epsilon \in [0, \alpha]$.

For the fairness loss, consider $\epsilon = \alpha$, and fix any $\pi^* \in Q_{\alpha-\epsilon}$. We then have

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbf{Err}(\pi^t, (x^t, y^t)) - \sum_{t=1}^{T} \mathbf{Err}(\pi^*, (x^t, y^t)) + C\alpha \sum_{t=1}^{T} \mathbf{Unfair}_\alpha(\pi^t, z^t) \leq 2(k + 2C)\sqrt{\ln(|\mathcal{F}|)T} \\
&\Rightarrow C\alpha \sum_{t=1}^{T} \mathbf{Unfair}_\alpha(\pi^t, z^t) \leq 2(k + 2C)\sqrt{\ln(|\mathcal{F}|)T} + kT \\
&\Rightarrow C\alpha \sum_{t=1}^{T} \mathbf{Unfair}_\alpha(\pi^t, z^t) \leq 6C\sqrt{\ln(|\mathcal{F}|)T} + kT \\
&\Rightarrow \sum_{t=1}^{T} \mathbf{Unfair}_\alpha(\pi^t, z^t) \leq \frac{6}{\alpha}\sqrt{\ln(|\mathcal{F}|)T} + \frac{kT}{\alpha C} \\
&\Rightarrow \sum_{t=1}^{T} \mathbf{Unfair}_\alpha(\pi^t, z^t) \leq \frac{6}{\alpha}\sqrt{\ln(|\mathcal{F}|)T} + \frac{k}{\alpha}T^{\frac{3}{4}}
\end{aligned}$$

where the first implication follows from the fact that

$$\sum_{t=1}^{T}\sum_{i=1}^{k}\ell\left(\pi^{t}\left(x_i^t\right),y_i^t\right)-\ell\left(\pi^{*}\left(x_i^t\right),y_i^t\right)\geq -kT.$$

As for the misclassification regret, consider $\epsilon = 0$.

$$\begin{aligned}
&\sum_{t=1}^{T}\mathbf{Err}(\pi^t, z^t) - \min_{\pi^*\in Q_\alpha}\sum_{t=1}^{T}\mathbf{Err}(\pi^*, z^t)\\
&\leq \sum_{t=1}^{T}\mathcal{L}_C(\pi^t, z^t) - \min_{\pi^*\in\Delta\mathcal{F}}\mathcal{L}_C(\pi^*, z^t)\\
&\leq 2(k+2C)\sqrt{\ln(|\mathcal{F}|)T} \leq 6\sqrt{\ln(|\mathcal{F}|)}T^{\frac{3}{4}}.
\end{aligned}$$

□

We remark that C is set to be at least $\max(k, T^{\frac{1}{4}})$ so as to bound both the misclassification regret and the fairness loss with $O(T^{\frac{3}{4}})$, but other trade-off between the two is still possible.

3.5.3. Follow-The-Perturbed-Leader

Running exponential weights as in Section 3.5.2 in general cannot be done in an efficient manner, as such methods need to calculate the loss for each hypothesis $f \in \mathcal{F}$ or for each possible labeling in the case $\mathcal{F}$ has bounded VC-dimension. Here we intend to come up with an algorithm that is oracle-efficient.

Specifically, we show how the algorithm proposed by Syrgkanis et al. [90] can be used as an $\boldsymbol{A}_{\text{BATCH}}$ to achieve sublinear regret in the online batch classification setting in an oracle efficient manner. However, we remark that our approach of leveraging CONTEXT-FTPL requires us to relax how adaptive the environment can be in terms of choosing (x^t, y^t). Previously, we allowed the environment to choose (x^t, y^t) with the full knowledge of the deployed policy π^t. Here, we make an assumption that (x^t, y^t) can be formed as a function of $h^t = ((\pi^1, z^1), \ldots, (\pi^{t-1}, z^{t-1}))$ but *not* π^t.

Let us now consider the setting that Syrgkanis et al. [90] study. They consider an adversarial contextual learning setting where in each round t, the learner randomly deploys some hypothesis[6] $\psi^t \in \Psi$ where $\Psi : \Xi \to \{0,1\}^k$, and the environment chooses $(\xi^t, w^t) \in \Xi \times \mathbb{R}^k$, where k indicates the number of possible actions that can be taken for the context ξ^t whose associated loss vector is w^k. The only knowledge at round t *not* available to the environment is the randomness that the learner uses to choose ψ^t, but the environment may know the actual distribution over ψ^t that the learner has in round t just not the realization of it. And at the end of the round, the learner suffers some loss $L(\psi^t, (\xi^t, w^t))$.

They show that in the small separator setting, they can achieve sublinear regret given that they can compute a separator set prior to learning. We first give the definition of a separator set and then state their regret guarantee.

Definition 13. *A set $S = (\xi_1, \ldots, \xi_n)$ is called a* seperator set *for $\Psi : \Xi \to \{0,1\}^k$ if for any different ψ and ψ' in Ψ, there exists $\xi \in S$ such that $\psi(\xi) \neq \psi'(\xi)$.*

Theorem 9 (Syrgkanis et al. [90])**.** *For any adversarially and adaptively chosen sequence of $(\xi^t, w^t)_{t=1}^T$,* CONTEXT-FTPL *initialized with a separator set S and parameter ω deploys $(\psi^t)_{t=1}^T$ with the following regret: for any $\psi^* \in \Psi$,*

$$\begin{aligned}
&\sum_{t=1}^T \mathop{\mathbb{E}}_{\psi^t \sim D^t} \left[L(\psi^t, (\xi^t, w^t)) \right] - \sum_{t=1}^T L(\psi^*, (\xi^t, w^t)) \\
&\leq 4\omega k n \sum_{t=1}^T \mathop{\mathbb{E}}_{\psi^t \sim D^t} [\left\| L(\cdot, (\cdot, w^t)) \right\|_*^2] + \frac{10}{\omega} \sqrt{nk} \log(|\Psi|),
\end{aligned}$$

where $n = |S|$, $\left\| L(\cdot, (\cdot, w)) \right\|_* = \max_{\psi, \xi} |L(\psi, (\xi, w))|$ and D^t is the implicit distribution over ψ^t that CONTEXT-FTPL *has in each round t.*

Our online batch classification setting can be easily reduced to their setting by simply considering the batch of instances by setting $\xi^t = x^t = (x_1^t, \ldots, x_k^t)$, meaning we set $\Xi = \mathcal{X}^k$

[6] They refer to this as a policy, but we say hypothesis just to be consistent with our terminology where a function that maps to $\{0,1\}$ is called hypothesis and policy is reserved for a mixture of hypotheses that maps to $[0,1]$.

and form the associated loss vector as $w_i^t = \frac{1-2y_i^t}{2k}$ for each $i \in [k]$. And we view each hypothesis as $\psi_f(x^t) = (f(x_1^t), \dots, f(x_k^t))$. In other words, we can define the hypothesis class induced by $\mathcal{F}$ as

$$\Psi_{\mathcal{F},k} = \left\{ \forall f \in \mathcal{F} : (x_i)_{i=1}^k \mapsto (f(x_i))_{i=1}^k \right\}.$$

Note that $|\mathcal{F}| = |\Psi_{\mathcal{F},k}|$. And we can use the following linear loss

$$L_{\text{BATCH},k}\left(\psi_f, (\xi^t, w^t)\right) = \left\langle \psi_f(\xi^t), w^t \right\rangle.$$

Note that by construction, the difference in $L_{\text{BATCH},k}$ over (ξ^t, w^t) between ψ_f and $\psi_{f'}$ preserves the difference in misclassification loss over (x^t, y^t) between f and f':

Lemma 9. *Write $k' = k + 2C$.*

$$2k'\left(L_{\text{BATCH},k'}\left(\psi_f, (\xi^t, w^t)\right) - L_{\text{BATCH},k'}\left(\psi_{f'}, (\xi^t, w^t)\right)\right) = \sum_{i=1}^{k'} \ell(f(x_i), y_i) - \sum_{i=1}^{k'} \ell(f'({x'}_i^t), {y'}_i^t)$$

Proof.

$$\begin{aligned}
&2k'\left(L_{\text{BATCH},k'}\left(\psi_f, (\xi^t, w^t)\right) - L_{\text{BATCH},k'}\left(\psi_f, (\xi^t, w^t)\right)\right)\\
&= \left\langle (f({x'}_i^t))_{i=1}^k, 1 - 2{y'}^t \right\rangle - \left\langle (f'({x'}_i))_{i=1}^k, 1 - 2{y'}^t \right\rangle\\
&= \left(\sum_{i=1}^{k'} (1 - f({x'}_i^t)) \cdot {y'}_i^t + f({x'}_i^t) \cdot (1 - {y'}_i^t)\right) - \left(\sum_{i=1}^{k'} (1 - f'({x'}_i)) \cdot {y'}_i^t + \pi({x'}_i^t) \cdot (1 - y_i^t)\right)\\
&= \sum_{i=1}^{k'} \ell(f({x'}_i^t), {y'}_i^t) - \sum_{i=1}^{k'} \ell(f'({x'}_i^t), {y'}_i^t)
\end{aligned}$$

□

Syrgkanis et al. [90] assume an optimization oracle with respect to L

$$M_L(\{(\xi^j, y^j)\}_{j=1}^P) = \arg\min_{\psi \in \Psi} \sum_{j=1}^{P} L(\psi, (\xi^j, w^j))$$

which in our case corresponds to the following oracle:

$$M_{L_{\text{BATCH},k}}(\{(\xi^j, w^j)\}_{j=1}^D) = \psi_f \quad \text{where } f = \arg\min_{f \in \mathcal{F}} \sum_{j=1}^{D} \sum_{i=1}^{k} f(x_i^j) w_i^j.$$

Note that this is equivalent to a weighted empirical risk minimization oracle:

$$\begin{aligned}
&\arg\min_{f \in \mathcal{F}} \sum_{j=1}^{D} \sum_{i=1}^{k} f(x_i^j) w_i^j \\
&= \arg\min_{f \in \mathcal{F}} \sum_{j=1}^{D} \sum_{i=1}^{k} f(x_i^j) p_i^j \left(\frac{1 - 2y_i^j}{2k} \right) \\
&= \arg\min_{f \in \mathcal{F}} \sum_{j=1}^{D} \sum_{i=1}^{k} p_i^j \ell(f, (x_i^j, y_i^j))
\end{aligned}$$

where $y_i^j = -\text{sign}(w_i^j)$ and $p_i^j = \frac{w_i^t}{y_i^t}$ for each $j \in [D], i \in [k]$. We remark that not all w^j that we feed to the oracle will be of the form $\{\pm \frac{1}{2k}\}$ and $p_i^j = 1$ because CONTEXT-FTPL requires calling the oracle not just on the set of ξ^t, w^t that we create from x^t and y^t — for stability reasons, it also adds in contexts from the separator set and associate each of those contexts with a random vector where each coordinate is drawn from the Laplace distribution.

Furthermore, we can turn any separator set $S \subseteq \mathcal{X}$ for $\mathcal{F}$ into an equal size separator set $S' \subseteq \Xi$ for Ψ. In fact, the construction is as follows:

$$S' = \{\forall x \in S : \xi_x = (x, v, \dots, v)\},$$

where v is some arbitrary instance in $\mathcal{X}$.

Lemma 10. *If S is the separator set for $\mathcal{F}$, then S' is a separator set for $\Psi_{\mathcal{F}}$.*

Proof. Fix any f and f' where $f \neq f'$. Note that by definition of S, there exists $x \in S$ such that $f(x) \neq f'(x)$. As a result, $\psi_f(\xi_x) \neq \psi_{f'}(\xi_x)$ as $(f(x), q, \ldots, q) \neq (f(x'), q, \ldots, q)$. □

Because the loss we use is linear, we take a slightly different view on the interaction between the learner and the environment. Instead of the learner sampling a hypothesis ψ_f^t and having the no-regret guarantee in expectation over the randomness of sampling the hypothesis, we imagine the learner playing the actual distribution over ψ_f^t it has at round t. We note this distribution over ψ_f^t as D^t and write the loss experienced by deploying a *policy* D^t as

$$L_{\text{BATCH},k'}(D^t, (\xi^t, w^t)) = \mathop{\mathbb{E}}_{\psi_f^t \sim D^t}[\langle \psi_f^t(\xi), w^t \rangle] = \langle \mathop{\mathbb{E}}_{\psi_f^t \sim D^t}[\psi_f^t(\xi)], w^t \rangle = \mathop{\mathbb{E}}_{\psi_f^t \sim D^t}[L_{\text{BATCH},k'}(\psi_f^t, (\xi^t, w^t)].$$

However, CONTEXT-FTPL never explicitly keeps track of the distribution D^t but only allows a way to sample from this distribution. Therefore, we form an empirical distribution $\tilde{D}^t$ by calling into CONTEXT-FTPL E many times to approximate D^t — i.e. we write $\tilde{D}^t$ to denote the uniform distribution over $\{\psi_{f_1^t}^t, \ldots, \psi_{f_1^t}^t\}$ where $\psi_{f_j^t}^t$ is the result of our jth call to CONTEXT-FTPL in round t. We describe the overall reduction to CONTEXT-FTPL more formally in Algorithm 4.

Algorithm 4 Reduction to CONTEXT-FTPL

```
Input: Separating set $S$
Create $S' = \{\forall x \in S : \xi_x = (x, v, \dots, v)\}$ where $v \in \mathcal{X}$ is chosen arbitrarily.
Initialize CONTEXT-FTPL with $S'$ and $\omega$.
for $t = 1, \dots, T$ do
    $\pi^t$ is deployed.
    Environment, without the knowledge of $\pi^t$, chooses $(x^t, y^t)$.
    Auditor chooses $\rho^t_{O_\alpha} = O_\alpha(x^t, \pi^t)$.
    // Incur misclassification and fairness loss
    Incur misclassification loss $\sum_{k=1}^{k} \ell(\pi^t(x^t), y^t)$
    Incur fairness loss Unfair$(\pi^t, \rho^t_{O_\alpha})$.
    // Reduction to online batch classification setting and to
        CONTEXT-FTPL's setting
    $(x'^t, y'^t) = R_C((x^t, y^t), \rho^t_{O_\alpha})$.
    $\xi^t = x'^t$ and $w^t_i = \frac{1-2y'^t_i}{2(k+2C)}$ for each $i \in [k+2C]$.
    Update history $h^{t+1} = \{(\xi^\tau, w^\tau)\}_{\tau=1}^t$.
    for $j \in [E]$ do
        $\psi_f^{t+1,j}$ = CONTEXT-FTPL$(h^{t+1})$.
        Set $f_j^{t+1} = f$ from $\psi_f^{t+1,j}$.
    end
    $\pi^{t+1}$ be a uniform distribution over $\{f_1^{t+1}, \dots, f_E^{t+1}\}$.
end
```

Unlike before, we have the environment choose (x^t, y^t) without the knowledge of π^t. This is so that the randomness used to form π^t by running CONTEXT-FTPL multiple times is not revealed to the environment. If the randomness is revealed, it's possible that the auditor can take advantage of the direction in which the empirical distribution that we deploy is off from the distribution maintained by CONTEXT-FTPL — more specifically, we would have to take a union bound over all possible x^t and y^t that the environment can choose after the environment knows how π^t has been chosen, which we can't do because there are infinitely many possible (x, y).

Instead, we could argue about the concentration of the policy π^t itself to its expected value instead of over the loss first and use the concentration over the distribution of hypotheses to argue for the concentration of the loss. However, the loss needs to average over each hypothesis or each possible labeling in the case of bounded VC-dimension, so our estimation error in the distribution over each hypothesis will add up over each $f \in \mathcal{F}$, resulting in linear dependence on $|\mathcal{F}|$, or over each possible labeling induced by $\mathcal{F}$ incurring estimation error linear in $O(k^V)$ where V is the VC-dimension of $\mathcal{F}$.

Hence, by hiding the randomness used to sample the empirical distribution from the environment (i.e. (x^t, y^t) has to be chosen without access to $\tilde{D}^t$), the only thing in the loss $L_{\text{BATCH},k+2C}$ that is adaptive to the deployed policy is the auditor. However, the auditor only has $k^2 + 1$ options (i.e. choose a pair out of k^2 pair or output null), so we can easily union bound over these options.

Lemma 11. *With probability $1-\delta$ over the randomness of $\tilde{D}^t$ (i.e. sampling f^t_j for $j \in [E]$), we have*

$$\left| L_{\text{BATCH},k+2C}(\tilde{D}^t, (\xi^t, w^t)) - L_{\text{BATCH},k+2C}(D^t, (\xi^t, w^t)) \right| \leq \sqrt{\frac{\ln(\frac{2T(k^2+1)}{\delta})}{2E}}$$

for every round $t \in [T]$ where $\psi^t = \text{CONTEXT-FTPL}((\xi^\tau, w^\tau)_{\tau=1}^{t-1})$ is distributed according D^t. $\tilde{D}^t$ is the uniform distribution over $\{\psi^{t,j}_f\}_{j \in [E]}$. (ξ^t, w^t) is determined according to (x^t, y^t) and $\rho^t_{O_\alpha} = O(x^t, \pi^t)$ where π^t is the corresponding policy for $\tilde{D}^t$ that is deployed in round t as shown in Algorithm 4.

Proof. Fix the round $t \in [T]$. Note that

$$
\begin{aligned}
&L_{\text{BATCH},k+2C}(\tilde{D}^t, (\xi^t, w^t)) - L_{\text{BATCH},k+2C}(D^t, (\xi^t, w^t)) \\
&= \left\langle \mathop{\mathbb{E}}_{\psi_f^t \sim \tilde{D}^t} [\psi_f^t(\xi^t)], w^t \right\rangle - \left\langle \mathop{\mathbb{E}}_{\psi_f^t \sim D^t} [\psi_f^t(\xi^t)], w^t \right\rangle \\
&= \frac{1}{|E|} \sum_{j \in [E]} \sum_{i=1}^{k+2C} f_j^t({x'}_i^t) w_i^t - \mathop{\mathbb{E}}_{\psi_f^t \sim D^t} \left[\sum_{i=1}^{k+2C} \sum_{i=1}^{k+2C} f({x'}_i^t) w_i^t \right] \\
&= \left\langle \mathop{\mathbb{E}}_{\psi_f^t \sim \tilde{D}^t} [\psi_f^t(\xi^t)], w^t \right\rangle - \left\langle \mathop{\mathbb{E}}_{\psi_f^t \sim D^t} [\psi_f^t(\xi^t)], w^t \right\rangle \\
&= \frac{1}{E} \sum_{j \in [E]} \sum_{i=1}^{k} f_j^t(x_i^t) w_i^t + C \left(f_j^t({x'}_{k+1}^t) w_{k+1}^t + f_j^t({x'}_{k+C+1}^t) w_{k+C+1}^t \right) \\
&- \mathop{\mathbb{E}}_{\psi_f \sim D^t} \left[\sum_{i=1}^{k} \sum_{i=1}^{k} f(x_i^t) w_i^t + C \left(f(x_{\rho_1^t}^t) w_{k+1}^t + f(x_{\rho_2^t}^t) w_{k+C+1}^t \right) \right].
\end{aligned}
$$

Note that just by the construction of ξ^t and ${x'}^t, {y'}^t$, we know that there are only $k^2 + 1$ possible options for $({x'}_{k+1}^t, {x'}_{k+C+1})$. More specifically, if $\rho_{O_\alpha}^t \neq null$, then ${x'}_{k+1}^t = x_{\rho_1^t}^t$ and ${x'}_{k+C+1} = x_{\rho_2^t}^t$ where $\rho_1^t \in [k]$ and $\rho_2^t \in [k]$. If $\rho_{O_\alpha}^t = null$, then $({x'}_{k+1}^t, {x'}_{k+C+1}) = (v, v)$ always. Furthermore, by construction, we have $w_{k+1}^t = \frac{-1}{2(k+2C)}$ and $w_{k+C+1}^t = \frac{1}{2(k+2C)}$ always. Write

$$
\begin{aligned}
V_j &= \sum_{i=1}^{k} f_j^t(x_i^t) w_i^t + C \left(f_j^t(x_{\rho_1^t}^t) w_{k+1}^t + f_j^t(x_{\rho_2^t}^t) w_{k+C+1}^t \right) \\
V &= \mathop{\mathbb{E}}_{\psi_f^t \sim D^t} \left[\sum_{i=1}^{k} \sum_{i=1}^{k} f(x_i^t) w_i^t + C \left(f(x_{\rho_1^t}^t) w_{k+1}^t + f(x_{\rho_2^t}^t) w_{k+C+1}^t \right) \right].
\end{aligned}
$$

Note that $\mathbb{E}[V_j] = V$ for each $j \in [E]$. Also, note that by construction,

$$
f(x_i^t) w_i^t \in \left[-\frac{1}{2(k+2C)}, \frac{1}{2(k+2C)} \right],
$$

meaning $V_j \in [-\frac{1}{2}, \frac{1}{2}]$. Therefore, union bounding over all possible $\rho^t \in [k]^2 \cup \{null\}$ with

Chernoff bound, we have

$$\Pr_{(f_j^t)_{j\in[E]}}\left(\left|\frac{1}{E}\sum_{j\in[E]} V_j - V\right| \geq \sqrt{\frac{\ln(\frac{2(k^2+1)}{\delta})}{2E}}\right) \leq \delta.$$

Union bounding over all round $t \in [T]$, we have with probability $1-\delta$,

$$\left|L_{\text{BATCH},k+2C}(\tilde{D}^t, (\xi^t, w^t)) - L_{\text{BATCH},k+2C}(D^t, (\xi^t, w^t))\right| \leq \sqrt{\frac{\ln(\frac{2T(k^2+1)}{\delta})}{2E}}.$$

□

Now, we can combine all the arguments we have developed so far in order to prove that Algorithm 4 achieves sublinear fairness loss and misclassification regret:

Theorem 10. *Set $C = \max(k, T^{\frac{2}{9}}), E = T$, and $\omega = n^{\frac{-1}{4}} k'^{\frac{-3}{4}} T^{\frac{-1}{2}} \log(|\mathcal{F}|)^{\frac{1}{2}}$ where n is the size of the separator set S. Algorithm 4 can guarantee that with probability $1-\delta$, the following holds true:*

$$\sum_{t=1}^{T} \mathbf{Unfair}_{\alpha}(\pi^t, z^t) \leq \frac{1}{\alpha} O\left(n^{\frac{3}{4}} \log(|\mathcal{F}|)^{\frac{1}{2}} T^{\frac{5}{9}} + \sqrt{T \ln\left(\frac{Tk}{\delta}\right)} + kT^{\frac{7}{9}}\right)$$

$$\sum_{t=1}^{T} \mathbf{Err}(\pi^t, z^t) - \min_{\pi^* \in Q_{\alpha}} \sum_{t=1}^{T} \mathbf{Err}(\pi^*, z^t) \leq O\left(n^{\frac{3}{4}} \log(|\mathcal{F}|)^{\frac{1}{2}} T^{\frac{7}{9}} + \sqrt{\ln\left(\frac{Tk}{\delta}\right)} T^{\frac{13}{18}}\right)$$

where $z^t = ((x^t, y^t), \rho^t_{O_{\alpha}})$ and $\rho^t_{O_{\alpha}} = O_{\alpha}(x^t, \pi^t)$. In other words, the misclassification regret and fairness loss is both bounded by $O(T^{\frac{7}{9}})$ with high probability.

Proof.

$$\begin{aligned}
&\sum_{t=1}^{T} \mathcal{L}_{C,\alpha}(\pi^t, z^t) - \min_{\pi^* \in Q_{\alpha-\epsilon}} \sum_{t=1}^{T} \mathcal{L}_{C,\alpha}(\pi^*, z^t) \\
&\leq \sum_{t=1}^{T} \mathcal{L}_{C,\alpha}(\pi^t, z^t) - \min_{\pi^* \in \Delta\mathcal{F}} \sum_{t=1}^{T} \mathcal{L}_{C,\alpha}(\pi^*, z^t) \\
&= \sum_{t=1}^{T} \sum_{i=1}^{k+2C} \ell(\pi^t(x'^t), y'^t_i) - \min_{\pi^* \in \Delta\mathcal{F}} \sum_{t=1}^{T} \sum_{i=1}^{k+2C} \ell(\pi^*(x'^t), y'^t_i) && \text{Lemma 8} \\
&= \sum_{t=1}^{T} \sum_{i=1}^{k+2C} \ell(\pi^t(x'^t), y'^t_i) - \min_{f^* \in \mathcal{F}} \sum_{t=1}^{T} \sum_{i=1}^{k+2C} \ell(f^*(x'^t), y'^t_i) && \text{Optimal solution over linear objective must happen at the support}
\end{aligned}$$

Then, applying Lemma 9 yields with probability $1-\delta$,

$$\begin{aligned}
&= (k+2C)\left(\sum_{t=1}^{T} L_{\text{BATCH},k+2C}(\tilde{D}^t, (\xi^t, w^t)) - \min_{\psi^* \in \Psi_{\text{BATCH},k+2C}} \sum_{t=1}^{T} L_{\text{BATCH},k+2C}(\psi^*, (\xi^t, w^t))\right) \\
&\leq (k+2C)\left(\sum_{t=1}^{T} L_{\text{BATCH},k+2C}(D^t, (\xi^t, w^t)) - \min_{\psi^* \in \Psi_{\text{BATCH},k+2C}} \sum_{t=1}^{T} L_{\text{BATCH},k+2C}(\psi^*, (\xi^t, w^t))\right) \\
&+ T\sqrt{\frac{\ln(\frac{2T(k^2+1)}{\delta})}{2E}}\Bigg)
\end{aligned}$$

Notice that $|L_{\text{BATCH},k+2C}(\psi, (\xi, w^t))| \in [-\frac{1}{2}, \frac{1}{2}]$ for any ψ, x and $t \in [T]$ because by construction $w_i^t \in \{\pm\frac{1}{2(k+2C)}\}$. In other words, we have $\mathbb{E}_{\psi \sim D^t}\left[||L_{\text{BATCH},k+2C}(\psi, (\cdot, w^t))||_*^2\right] \leq \frac{1}{4}$. Theorem 9 gives us that a sequence of distribution $(D^t)_{t=1}^T$ achieves the following, where D^t is equivalent to the distribution of CONTEXT-FTPL$(((\xi^\tau, w^\tau))_{\tau=1}^{t-1})$:

$$\begin{aligned}
&\sum_{t=1}^{T} L_{\text{BATCH},k+2C}(D^t, (\xi^t, w^t)) - \min_{\psi^* \in \Psi_{\text{BATCH},k+2C}} \sum_{t=1}^{T} L_{\text{BATCH},k+2C}(\psi^*, (\xi^t, w^t)) \\
&\leq \omega k' n T + \frac{10}{\omega}\sqrt{nk'} \ln(|\Psi_{\mathcal{F},k+2C}|).
\end{aligned}$$

Therefore, writing $k' = k + 2C$, we have with probability $1 - \delta$,

$$\begin{aligned}
&\sum_{t=1}^{T} \mathcal{L}_{C,\alpha}(\pi^t, z^t) - \min_{\pi^* \in Q_{\alpha-\epsilon}} \sum_{t=1}^{T} \mathcal{L}_{C,\alpha}(\pi^*, z^t) \\
&\leq k' \left(\sum_{t=1}^{T} \mathop{\mathbb{E}}_{\psi^t} \left[L_{\text{BATCH},k'}(\psi^t, (\xi^t, w^t)) \right] - \sum_{t=1}^{T} L_{\text{BATCH},k'}(\psi^*, (\xi^t, w^t)) + \sqrt{T \frac{\ln(\frac{2T(k^2+1)}{\delta})}{2}} \right) \\
&\leq k' \left(\omega k' n T + \frac{10}{\omega} \sqrt{nk'} \ln(|\Psi_{\mathcal{F},k+2C}|) + \sqrt{T \frac{\ln(\frac{2T(k^2+1)}{\delta})}{2}} \right).
\end{aligned}$$

Setting $\omega = n^{\frac{-1}{4}} k'^{\frac{-3}{4}} T^{\frac{-1}{2}} \log(|\mathcal{F}|)^{\frac{1}{2}}$, we then have

$$\begin{aligned}
&\sum_{t=1}^{T} \mathcal{L}_{C,\alpha}(\pi^t, z^t) - \min_{\pi^* \in Q_{\alpha-\epsilon}} \sum_{t=1}^{T} \mathcal{L}_{C,\alpha}(\pi^*, z^t) \\
&\leq O\left(\left(n^{\frac{3}{4}} k'^{\frac{5}{4}} \log(|\mathcal{F}|)^{\frac{1}{2}} + k' \sqrt{\ln\left(\frac{Tk}{\delta}\right)} \right) T^{\frac{1}{2}} \right)
\end{aligned}$$

With the same argument as in the proof of Theorem 8, we get that for $\epsilon = \alpha$

$$\begin{aligned}
&C\alpha \sum_{t=1}^{T} \mathbf{Unfair}_{\alpha}(\pi^t, z^t) \leq O\left(\left(n^{\frac{3}{4}} k'^{\frac{5}{4}} \log(|\mathcal{F}|)^{\frac{1}{2}} + k' \sqrt{\ln\left(\frac{Tk}{\delta}\right)} \right) T^{\frac{1}{2}} \right) + kT \\
&\Rightarrow C\alpha \sum_{t=1}^{T} \mathbf{Unfair}_{\alpha}(\pi^t, z^t) \leq O\left(\left(n^{\frac{3}{4}} C^{\frac{5}{4}} \log(|\mathcal{F}|)^{\frac{1}{2}} + C \sqrt{\ln\left(\frac{Tk}{\delta}\right)} \right) T^{\frac{1}{2}} \right) + kT \\
&\Rightarrow \sum_{t=1}^{T} \mathbf{Unfair}_{\alpha}(\pi^t, z^t) \leq \frac{1}{\alpha} O\left(\left(n^{\frac{3}{4}} C^{\frac{1}{4}} \log(|\mathcal{F}|)^{\frac{1}{2}} + \sqrt{\ln\left(\frac{Tk}{\delta}\right)} \right) T^{\frac{1}{2}} \right) + \frac{kT}{C} \\
&\Rightarrow \sum_{t=1}^{T} \mathbf{Unfair}_{\alpha}(\pi^t, z^t) \leq \frac{1}{\alpha} O\left(n^{\frac{3}{4}} \log(|\mathcal{F}|)^{\frac{1}{2}} T^{\frac{5}{9}} + \sqrt{T \ln\left(\frac{Tk}{\delta}\right)} + kT^{\frac{7}{9}} \right).
\end{aligned}$$

As for the misclassification regret, we have with $\epsilon = 0$

$$\begin{aligned}
&\sum_{t=1}^{T} \mathbf{Err}(\pi^t, z^t) - \min_{\pi^* \in Q_\alpha} \sum_{t=1}^{T} \mathbf{Err}(\pi^*, z^t) \\
&\leq O\left(\left(n^{\frac{3}{4}} k'^{\frac{5}{4}} \log(|\mathcal{F}|)^{\frac{1}{2}} + k' \sqrt{\ln\left(\frac{Tk}{\delta}\right)}\right) T^{\frac{1}{2}}\right) \\
&\leq O\left(n^{\frac{3}{4}} \log(|\mathcal{F}|)^{\frac{1}{2}} T^{\frac{7}{9}} + \sqrt{\ln\left(\frac{Tk}{\delta}\right)} T^{\frac{13}{18}}\right)
\end{aligned}$$

□

We only focus on their small separator set setting, although their transductive setting (i.e. the contexts $(x_t)_{t=1}^{T}$ are known in advance) and other bandit settings should naturally follow as well.

3.6. Discussion

In this chapter, we have removed several binding restrictions in the context of learning with individual fairness present in Chapter 2. Relieving the metric assumption as well as the assumption regarding full access to the similarity measure and only requiring the auditor to detect a single violation at every round can be helpful in making individual fairness more achievable and easier to implement in practice.

There are still interesting future directions. It would be interesting to explore the interaction with different models of feedback (one natural variant being one-sided feedback). We suspect that Syrgkanis et al. [90]'s way of handling the bandit feedback can be easily ported over to handle the bandit nature in our setting. Second, thinking about a model where the auditor only has access to binary decisions may be helpful in further closing the gap to practical use. Third, as most of the literature on individual fairness (including this work) is decoupling the similarity measure from the distribution over the target variable, it would be desirable to try to explore and quantify the compatibility of the two in specific instances.

Chapter 4

Fairness Elicitation

4.1. Introduction

In Chapter 2 and 3, we have shown how to elicit notions of fairness from stakeholders and domain experts in an online setting. Because there are multiple rounds, the auditor is able to peak into the deployed model at every round and determine whether there exists any particular pair that is not being fairly according to some fairness metric.

However, in practice, it is quite expensive to re-deploy a model over many rounds and have the auditor audit those models each time. In that regard, we study an offline setting where we want to elicit notions of fairness from stakeholders and domain experts prior to deploying and deploying a model.

Similarly as before, we aim to elicit stakeholders conceptions of fairness by asking them to compare pairs of individuals in specific scenarios. Specifically, we ask whether it's fair that one particular individual should receive an outcome that is as desirable or better than the other.

When pointing out fairness or unfairness, this kind of pairwise ranking is natural. For example, after Serena Williams was penalized for a verbal interaction with an umpire in the 2018 U.S. Open Finals, tennis player James Blake tweeted, "I have said worse and not gotten penalized. And I've also been given a 'soft warning' by the ump where they tell you knock it off or I will have to give you a violation. [The umpire] should have at least given [Williams] that courtesy" [96]. Here, Blake thinks that: 1) Williams should have been judged as or less severely than he would have been in a similar situation; and 2) the umpire's decision was unfair, because Williams was judged more severely.

Thus, we ask a set of stakeholders about a fixed set of pairs of individuals subject to a classification problem. For each pair of individuals (A, B), we ask the stakeholder to choose from amongst a set of four options:

1. Fair outcomes must classify A and B the *same* way (i.e. they must either both get a favorable classification or both get an unfavorable classification).

2. Fair outcomes must give A an outcome that is equal to *or preferable to* the outcome of B.

3. Fair outcomes must give B an outcome that is equal to *or preferable to* the outcome of A

4. Fair outcomes may treat A and B differently without any constraints.

These constraints, a data distribution, and a hypothesis class define a learning problem: minimize classification error subject to the constraint that the rate of violation of the elicited pairwise constraints is held below some fixed threshold. Crucially and intentionally we elicit relative pairwise orderings of outcomes (e.g. A and B should be treated equally), but do not elicit preferences for absolute outcomes (e.g. A should receive a positive outcome). This is because *fairness* — in contrast to *justice* — is often conceptualized as a measure of equality of outcomes, rather than correctness of outcomes[7]. In particular, it remains the job of the learning algorithm to optimize for correctness subject to elicited fairness constraints.

We remark that the premise (and the foundation for the enormous success) of machine learning is that accurate decision making rules in complex scenarios cannot be defined with simple analytic rules and instead are best derived directly from data. Our work can be viewed similarly, as deriving fairness constraints from data elicited from experts and stakeholders. In this chapter, we solve the computational, statistical, and conceptual issues necessary to do this, and demonstrate the effectiveness of our approach via a small behavioral study.

[7] Sidney Morgenbesser, following the Columbia University campus protests in the 1960s, reportedly said that the police had treated him unjustly, but not unfairly. He said that he was treated unjustly because the police hit him without provocation — but not unfairly, because the police were doing the same to everyone else as well.

4.1.1. Results

Our Model We model individuals as having features in $\mathcal{X}$ and binary labels, drawn from some distribution $\mathcal{P}$. A committee of *stakeholders*[8] $u \in \mathcal{U}$ has preferences about whether one individual should be judged better than another individual. We imagine presenting each stakeholder with a set of pairs of individuals and asking them to choose one of four options for each pair, e.g. given the features of Serena Williams and Jacob Blake:

1. No constraint;
2. Williams should be treated as well as Blake or better;
3. Blake should be treated as well as Williams or better; or
4. Williams and Blake should be treated similarly.

Here, when we refer to how an individual *should be treated*, we mean the probability that an individual is given a positive label by the classifier. This may be a bit of a relaxation of these judgments, since they are not about actualized classifications, but rather the *probabilities* of positive classification. For example, we may not consider it a violation of fairness preference (2) if Williams is judged worse than Blake in a specific scenario; yet, if an ump is *more likely* to judge Williams worse than Blake in general, then this would violate this fairness preference.

We represent these preferences abstractly as a set of ordered pairs $C_u \subseteq \mathcal{X} \times \mathcal{X}$ for each stakeholder u. If $(x, x') \in C_u$, this means that stakeholder u believes that individual x' must be treated as well as individual x or better, i.e. ideally the classifier h classifies such that $h(x') \geq h(x)$. This captures all possible responses above. For example, for Serena Williams (s) and Jacob Blake (b), if stakeholder u responds:

1. *No constraint* $\Leftrightarrow (s, b) \notin C_u$ nor $(b, s) \notin C_u$;

[8] Though we develop our formalism as a committee of stakeholders, note that it permits the special case of a single subjective stakeholder, which we make use of in our behavioral study.

2. *Williams as well as Blake* $\Leftrightarrow (b, s) \in C_u$;

3. *Blake as well as Williams* $\Leftrightarrow (s, b) \in C_u$; or

4. *Treated similarly* $\Leftrightarrow (s, b) \in C_u$ and $(b, s) \in C_u$ (since if $h(b) \geq h(s)$ and $h(s) \geq h(b)$, then $h(s) = h(b)$).

We impose no structure on how stakeholders form their views nor on the relationship between the views of different stakeholders — i.e. the sets $\{C_u\}_{u \in \mathcal{U}}$ are allowed to be arbitrary (for example, they need not satisfy a triangle inequality), and need not be mutually consistent. We write $C = \cup_u C_u$.

We then formulate an optimization problem constrained by these pairwise fairness constraints. Since it is intractable to require that all constraints in C be satisfied exactly, we formulate two different "knobs" with which we can quantitatively relax our fairness constraints.

For $\gamma > 0$ (our first knob), we say that the classification of an ordered pair of individuals $(x, x') \in C$ satisfies γ-fairness if the probability of positive classification for x' plus γ is no smaller than the probability of positive classification for x, i.e. $\mathbb{E}[h(x')] + \gamma \geq \mathbb{E}[h(x)]$. In this expression, the expectation is taken only over the randomness of the classifier h. Equivalently, a γ-fairness violation corresponds to the classification of an ordered pair of individuals $(x, x') \in C$ if the difference between these probabilities of positive classification is greater than γ, i.e. $\mathbb{E}[h(x) - h(x')] > \gamma$. Thus, γ acts as a buffer on how likely it is that x' be classified worse than x before a fairness violation occurs. For example, if Blake (b) receives a good label (i.e. no penalty) 80% of the time and Williams (s) 50% of the time, then for $\gamma = 0.1$ this constitutes a γ-fairness violation for the ordered pair $(b, s) \in C$, since $\mathbb{E}[h(b) - h(s)] = 0.3 \geq 0.1 = \gamma$.

We might ask that for *no* pair of individuals do we have a γ-fairness violation:

$$\max_{(x,x')\in C} \mathbb{E}[h(x) - h(x')] \leq \gamma.$$

On the other hand, we could ask for the weaker constraint that *over a random draw of a pair of individuals*, the expected fairness violation is at most η (our second knob): $\mathbb{E}_{(x,x')\sim\mathcal{P}^2}[(h(x) - h(x')) \cdot \mathbf{1}[(x,x') \in C]] \leq \eta$. We can also combine both relaxations to ask that the in expectation over random pairs, the "excess" fairness violation, on top of an allowed budget of γ, is at most η. For example, as above, if Blake receives a good label 80% of the time and Williams 50%, for $\gamma = 0.1$, the umpire classifier would pick up 0.2 excess fairness violation for $(b, s) \in C$. We weight these excess fairness violations by the proportion of stakeholders who agree with the corresponding fairness constraint and mandate their sum be less than η. Subject to these constraints, we would like to find the distribution over classifiers that minimizes classification error: given a setting of the parameters γ and η, this defines a benchmark with which we would like to compete.

Our Theoretical Results Even absent fairness constraints, learning to minimize 0/1 loss (even over linear classifiers) is computationally hard in the worst case (see e.g. [23, 24]). Despite this, learning seems to be empirically tractable in the real world. To capture the *additional* hardness of learning subject to fairness constraints, we follow several recent papers [2, 59] in aiming to develop *oracle efficient* learning algorithms. Oracle efficient algorithms are assumed to have access to an *oracle* (realized in experiments using a heuristic — see the next section) that can solve weighted classification problems. Given access to such an oracle, oracle efficient algorithms must run in polynomial time. We show that our fairness constrained learning problem is computationally no harder than unconstrained learning by giving such an oracle efficient algorithm (or reduction), and show moreover that its guarantees generalize from in-sample to out-of-sample in the usual way — with respect to both accuracy and the frequency and magnitude of fairness violations. Our algorithm is simple and amenable to implementation, and we use it in our experimental results.

Our Experimental Results We implement our algorithm and run a set of experiments on the COMPAS recidivism prediction dataset, using fairness constraints elicited from 43 human subjects. We establish that our algorithm converges quickly (even when implemented with fast learning heuristics, rather than "oracles"). We also explore the Pareto curves trading off error and fairness violations for different human subjects, and find empirically that there is a great deal of variability across subjects in terms of their conception of fairness, and in terms of the degree to which their expressed preferences are in conflict with accurate prediction. We find that most of the difficulty in balancing accuracy with the elicited fairness constraints can be attributed to a small fraction of the constraints.

4.2. Related Work

Our work is related to existing notions of *individual fairness* like Dwork et al. [19], Joseph et al. [50] that conceptualize fairness as a set of constraints binding on pairs of individuals, as we have studied in Chapter 2 and 3. In particular the notion of individual fairness proposed in Dwork et al. [19] and studied in previous chapters is closely related but distinct from the fairness notions we elicit in this chapter. In particular, in this chapter

1. We allow for constraints that require that individual A be treated better than or equal to individual B, whereas metric fairness constraints are symmetric, and only allow constraints of the form that A and B be treated similarly. In this sense, the fairness notion here may be more general.

2. We elicit binary judgements between pairs of individuals, whereas metric fairness is defined as a Lipschitz constraint on a real valued metric. In this sense, the new notion in this chapter is more restrictive, although it may be easier to elicit.

The most technically related piece of work is Rothblum and Yona [81], who prove similar generalization guarantees to ours for a relaxation of metric fairness: our definition is slightly more general, and our generalization guarantee somewhat tighter, but technically the results are closely related. Our conceptual focus and main results are quite different, however: for general learning problems, they prove worst-case hardness results, whereas we derive practical algorithms in the oracle-efficient model, and empirically evaluate them on real user data. Lahoti et al. [66] make a similar observation about guaranteeing fairness with respect to an unknown metric, although their aim is the orthogonal goal of fair representation learning.

Dwork et al. [19] first proposed the notion of individual metric-fairness that we take inspiration from, imagining fairness as a Lipschitz constraint on a randomized algorithm, with respect to some "task-specific metric". Since the original proposal, the question of where the metric should come from has been one of the primary obstacles to its adoption, and the focus of subsequent work. Zemel et al. [100] attempt to automatically learn a representation for the data (and hence, implicitly, a similarity metric) that causes a classifier to label an equal proportion of two protected groups as positive. Kim et al. [60] consider a group-fairness like relaxation of individual metric-fairness, asking that on average, individuals in pre-specified groups are classified with probabilities proportional to the average distance between individuals in those groups. They show how to learn such classifiers given access to an oracle which can evaluate the distance between two individuals according to the metric. Compared to our work, they assume the existence of a fairness metric which can be accessed using a quantitative oracle, and they use this metric to define a statistical rather than individual notion of fairness.

Ilvento [45] studies the problem of metric learning with the goal of using only a small number of numeric valued queries, which are hard for human beings to answer, relying more on comparison queries. In contrast with Ilvento [45], we do not attempt to learn a metric and instead directly learn a classifier consistent with the elicited pairwise fairness constraints.

4.3. Preliminaries

Let S denote a set of labeled examples $\{z_i = (x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathcal{X}$ is a feature vector and $y_i \in \mathcal{Y}$ is a label. We will also write $S_X = \{x_i\}_{i=1}^n$ and $S_Y = \{y_i\}_{i=1}^n$. Throughout this chapter, we will restrict attention to binary labels, so let $\mathcal{Y} = \{0, 1\}$. Let $\mathcal{P}$ denote the unknown distribution over $\mathcal{X} \times \mathcal{Y}$. Let $\mathcal{H}$ denote a hypothesis class containing binary classifiers $h : \mathcal{X} \to \mathcal{Y}$. We assume that $\mathcal{H}$ contains a constant classifier (which will imply that the "fairness constrained" ERM problem that we define is always feasible). We'll denote classification error of hypothesis h by $err(h, \mathcal{P}) := \Pr_{(x,y)\sim\mathcal{P}}(h(x) \neq y)$ and its empirical classification error by $err(h, S) := \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(h(x_i) \neq y_i)$.

We assume there is a set of one or more stakeholders $\mathcal{U}$, such that each stakeholder $u \in \mathcal{U}$ is identified with a set of ordered pairs (x, x') of individuals $C_u \subseteq \mathcal{X}^2$: for each $(x, x') \in C_u$, stakeholder u thinks that x' should be treated as well as x or better, i.e. ideally that for the learned classifier h, the classification $h(x') \geq h(x)$ (we will ask that this hold in expectation if the classifier is randomized, and will relax it in various ways). For each ordered pair (x, x'), let $w_{x,x'}$ be the fraction of stakeholders who would like individual x to be treated as well as x': that is, $w_{x,x'} = \frac{|\{u|(x,x')\in C_u\}|}{|\mathcal{U}|}$. Note that if $(x, x') \in C_u$ and $(x', x) \in C_u$, then the stakeholder wants x and x' to be treated similarly in that ideally $h(x) = h(x')$.

In practice, we will not have direct access to the sets of ordered pairs C_u corresponding to the stakeholders u, but we may ask them whether particular ordered pairs are in this set (see Section 4.6 for details about how we actually query human subjects). We model this by imagining that we present each stakeholder with a random set of pairs $A \subseteq [n]^2$, and for each ordered pair (x_i, x_j), ask if x_j should not be treated worse than x_i; we learn the set of ordered pairs in $A \cap C_u$ for each u. Define the empirical constraint set $\hat{C}_u = \{(x_i, x_j) \in C_u\}_{\forall (i,j)\in A}$ and $\hat{w}_{x_i x_j} = \frac{|\{u|(x,x')\in\hat{C}_u\}|}{|\mathcal{U}|}$, if $(i, j) \in A$ and 0 otherwise. We write that $\hat{C} = \cup_u \hat{C}_u$. For brevity, we will sometimes write w_{ij} instead of w_{x_i,x_j}. Note that $\hat{w}_{ij} = w_{ij}$ for every $(i, j) \in A$.

Our goal will be to find the distribution over classifiers from $\mathcal{H}$ that minimizes classification error, while satisfying the stakeholders' fairness preferences, captured by the constraints C. To do so, we'll try to find D, a probability distribution over $\mathcal{H}$, that minimizes the training error and satisfies the stakeholders' empirical fairness constraints, $\hat{C}$. For convenience, we denote the expected classification error of D as $err(D, \mathcal{P}) := \mathbb{E}_{h\sim D}[err(h, \mathcal{P})]$ and likewise its expected empirical classification error as $err(D,S) := \mathbb{E}_{h\sim D}[err(h,S)]$. We say that any distribution D over classifiers satisfies (γ,η)-approximate subjective fairness if it is a feasible solution to the following constrained empirical risk minimization problem:

$$\min_{D\in\Delta\mathcal{H},\alpha_{ij}\geq 0} err(D,S) \tag{4.1}$$

$$\text{such that } \forall(i,j)\in[n]^2: \quad \mathop{\mathbb{E}}_{h\sim D}[h(x_i)-h(x_j)] \leq \alpha_{ij}+\gamma \tag{4.2}$$

$$\sum_{(i,j)\in[n]^2} \frac{\hat{w}_{ij}\alpha_{ij}}{|A|} \leq \eta. \tag{4.3}$$

This "Fair ERM" problem, whose feasible region we denote by $\Omega(S,\hat{w},\gamma,\eta)$, has decision variables D and $\{\alpha_{ij}\}$, representing the distribution over classifiers and the "fairness violation" terms for each pair of training points, respectively. The parameters γ and η are constants which represent the two different "knobs" we have at our disposal to quantitatively relax the fairness constraint, in an ℓ_∞ and ℓ_1 sense, respectively.

The parameter γ defines, for any ordered pair (x_i,x_j), the maximum difference between the probabilities that x_i and x_j receive positive labels without constituting a fairness violation. The parameter α_{ij} captures the "excess fairness violation" beyond γ for (x_i,x_j). The parameter η upper bounds the sum of these allotted excess fairness violation terms α_{ij}, each weighted by the proportion of judges who perceive they ought to be treated similarly $\hat{w}_{ij}$ and normalized with the total number of pairs presented $|A|$. Thus, η bounds the expected degree of dissatisfaction of the panel of stakeholders $\mathcal{U}$, over the random choice of an ordered pair $(x_i,x_j)\in A$ and the randomness of their classification. We iterate over all $(i,j)\in[n]^2$ (not just those in $\hat{C}$) because $\hat{w}_{ij}=0$ if no judge prefers x_i should be classified as well as x_j.

To better understand γ and η, we consider them in isolation. First, suppose we set $\gamma=0$. Then, *any* difference in probabilities of positive classification between pairs is deemed a fairness violation. So, if we choose $(D,\{\alpha_{ij}\})$ such that the sum of weighted differences in positive classification probabilities exceeds η, i.e.

$$\sum_{(i,j)\in[n]^2} \frac{\hat{w}_{ij}\,\mathbb{E}_{h\sim D}[h(x_i)-h(x_j)]}{|A|} > \eta,$$

then this is an infeasible solution. For example, 50% of stakeholders think that Serena Williams (s) should be treated as well as James Blake (b), 70% of stakeholders think Williams should be treated as well as John McEnroe (m), and no other constraints ($|A|=6$); if Williams receives a good label 50% of the time, Blake 80%, McEnroe 90%, and $\eta = 0.07$, this is an η-fairness violation, since

$$\begin{aligned}&\left(\hat{w}_{bs}\,\mathbb{E}[h(b)-h(s)] + \hat{w}_{ms}\,\mathbb{E}[h(m)-h(s)]\right)/|A|\\ &= \left(0.5(0.8-0.5)+0.7(0.9-0.5)\right)/6 \approx 0.071 > 0.07 = \eta.\end{aligned}$$

Second, suppose that $\eta = 0$. Then, for any $(x_i, x_j) \in C$ (for which $\hat{w}_{ij} > 0$), if the expected difference in labels exceeds γ, i.e. $\mathbb{E}_{h\sim D}[h(x_i)-h(x_j)] > \gamma$, then this is an infeasible solution.

4.3.1. Fairness Loss

Our goal is to develop an algorithm that will minimize its empirical error $err(D,S)$, while satisfying the empirical fairness constraints $\hat{C}$. The standard VC dimension argument states that empirical classification error will concentrate around the true classification error: we hope to show the same kind of generalization for fairness as well. To do so, we first define fairness loss with respect to our elicited fairness preferences here.

For some fixed randomized hypothesis $D \in \Delta\mathcal{H}$ and w, define γ-fairness loss between an ordered pair as

$$\Pi_{D,w,\gamma}\left((x,x')\right) = w_{x,x'} \max\left(0, \mathop{\mathbb{E}}_{h\sim D}\left[h(x)-h(x')\right]-\gamma\right)$$

For a set of pairs $M \subset \mathcal{X} \times \mathcal{X}$, the γ-fairness loss of M is defined to be:

$$\Pi_{D,w,\gamma}(M) = \frac{1}{|M|} \sum_{(x,x') \in M} \Pi_{D,w,\gamma}\left((x, x')\right)$$

This is the expected degree to which the difference in classification probability for a randomly selected pair exceeds the allowable budget γ, weighted by the fraction of stakeholders who think that x' should be treated as well as x. By construction, the empirical fairness loss is bounded by η (i.e. $\Pi_{D,w,\gamma}(M) \leq \sum_{ij} \frac{\hat{w}_{ij}\alpha_{ij}}{|A|} \leq \eta$), and we show in Section 4.5, the empirical fairness should concentrate around the true fairness loss $\Pi_{D,w,\gamma}(\mathcal{P}) := \mathbb{E}_{x,x' \sim \mathcal{P}^2}\left[\Pi_{D,w,\gamma}(x, x')\right]$.

4.3.2. Cost-sensitive Classification

In our algorithm, we will make use of a cost-sensitive classification (CSC) oracle. An instance of CSC problem can be described by a set of costs $\{(x_i, c_i^0, c_i^1)\}_{i=1}^n$ and a hypothesis class, $\mathcal{H}$. Costs c_i^0 and c_i^1 correspond to the cost of labeling x_i as 0 and 1 respectively. Invoking a CSC oracle on $\{(x_i, c_i^0, c_i^1)\}_{i=1}^n$ returns a hypothesis h^* such that $h^* \in \text{argmin}_{h \in \mathcal{H}} \sum_{i=1}^n \left(h(x_i)c_i^1 + (1 - h(x_i))\, c_i^0\right)$. We say that an algorithm is *oracle-efficient* if it runs in polynomial time assuming access to a CSC oracle.

4.4. Empirical Risk Minimization

In this section, we give an oracle-efficient algorithm 5 for approximately solving our (in-sample) constrained empirical risk minimization problem. Details are deferred to Appendix 4.A. We prove the following theorem:

Theorem 11. *Fix parameters ν, C_τ, C_λ that serve to trade off running time with approximation error. There is an efficient algorithm that makes $T = \left(\frac{2C_\lambda\sqrt{\log(n)}+C_\tau}{\nu}\right)^2$ CSC oracle calls and outputs a solution $(\hat{D}, \hat{\alpha})$ with the following guarantee. The objective value is approximately optimal:*

$$err(\hat{D}, S) \leq \min_{(D,\alpha) \in \Omega(S,\hat{w},\gamma,\eta)} err(D, S) + 2\nu.$$

And the constraints are approximately satisfied: $\mathbb{E}_{h \sim \hat{D}}[h(x_i) - h(x_j)] \leq \hat{\alpha}_{ij} + \gamma + \frac{1+2\nu}{C_\lambda}, \forall (i,j) \in [n]^2$ *and* $\frac{1}{|A|} \sum_{(i,j) \in [n]^2} \hat{w}_{ij} \hat{\alpha}_{ij} \leq \eta + \frac{1+2\nu}{C_\tau}$.

4.4.1. Outline of the Solution

We frame the problem of solving our constrained ERM problem (equations (4.1) through (4.3)) as finding an approximate equilibrium of a zero-sum game between a primal player and a dual player, trying to minimize and maximize respectively the Lagrangian of the constrained optimization problem.

The Lagrangian for our optimization problem is

$$\begin{aligned}\mathcal{L}(D, \alpha, \lambda, \tau) = err(D, S) + &\sum_{(i,j) \in [n]^2} \lambda_{ij} \left(\mathop{\mathbb{E}}_{h \sim D}[h(x_i) - h(x_j)] - \alpha_{ij} - \gamma \right) \\ &+ \tau \left(\frac{1}{|A|} \sum_{(i,j) \in [n]^2} w_{ij} \alpha_{ij} - \eta \right)\end{aligned}$$

For the constraint in equation (4.2), corresponding to the γ-fairness violation for each ordered pair of individuals (x_i, x_j), we introduce a dual variable λ_{ij}. For the constraint (4.3), which corresponds to the η-fairness violation over all pairs of individuals, we introduce a dual variable of τ. For brevity, we define vectors $\lambda \in \Lambda$ and α which are made up of all the multipliers λ_{ij} and the excess fairness violation allotments α_{ij}, respectively. The primary player's action space is $(D, \alpha) \in (\Delta\mathcal{H}, [0,1]^{n^2})$, and the dual player's action space is $(\lambda, \tau) \in (\mathbb{R}^{n^2}, \mathbb{R})$.

Solving our constrained ERM problem is equivalent to finding a minmax equilibrium of $\mathcal{L}$:

$$\operatorname*{argmin}_{(D,\alpha) \in \Omega(S, \hat{w}, \gamma, \eta)} err(D, S) = \operatorname*{argmin}_{D \in \Delta\mathcal{H}, \alpha \in [0,1]^{n^2}} \max_{\lambda \in \mathbb{R}^{n^2}, \tau \in \mathbb{R}} \mathcal{L}(D, \alpha, \lambda, \tau)$$

Because $\mathcal{L}$ is linear in terms of its parameters, Sion's minimax theorem [88] gives us

$$\min_{D \in \Delta\mathcal{H}, \alpha \in [0,1]^{n^2}} \max_{\lambda \in \mathbb{R}^{n^2}, \tau \in \mathbb{R}} \mathcal{L}(D, \alpha, \lambda, \tau) = \max_{\lambda \in \mathbb{R}^{n^2}, \tau \in \mathbb{R}} \min_{D \in \Delta\mathcal{H}, \alpha \in [0,1]^{n^2}} \mathcal{L}(D, \alpha, \lambda, \tau).$$

By a classic result of Freund and Schapire [31], one can compute an approximate equilibrium by simulating "no-regret" dynamics between the primal and dual player. "No-regret" meaning that the average *regret* –or difference between our algorithm's plays and the single best play in hindsight– is bounded above by a term that converges to zero with increasing rounds.

In our case, we define a zero-sum game wherein the primary player's plays from action space $(D, \alpha) \in (\Delta\mathcal{H}, [0,1]^{n^2})$, and the dual player's plays from action space $(\lambda, \tau) \in (\mathbb{R}^{n^2}_{\geq 0}, \mathbb{R}_{\geq 0})$. In any given round t, the dual player plays first and the primal second. The primal player can simply best respond to the dual player (see Algorithm 5).

However, since the dual player plays first, they cannot simply best respond to the primal player's action. The dual player has to anticipate the primal player's best response in order to figure out what to play. Ideally, the dual player would enumerate every possible primal play and calculate the best dual response. However, this is intractable. So, the dual player updates dual variables $\{\lambda, \tau\}$ according to *no-regret* learning algorithms (exponentiated gradient descent [62] and online gradient descent [102], respectively).

The time-averaged play of both players converges to an approximate equilibrium of the zero-sum game, where the approximation is controlled by the regret of the dual player. This approximate equilibrium corresponds to an approximate saddle point for the Lagrangian $\mathcal{L}$, which is equivalent to an approximate solution to the Fair ERM problem.

We organize the rest of this section as follows. First, for simplicity, we show how the primal player updates $\{D, \alpha\}$ (even though the dual player plays first). Second, we show how the dual player updates $\{\lambda, \tau\}$. Finally, we prove that these updates are no-regret and relate the regret of the dual player to the approximation of the solution to the Fair ERM problem.

4.4.2. The Primal Player's Best Response

In each round t, given the actions chosen by the dual player (λ^t, τ^t), the primal player needs to best respond by choosing (D^t, α^t) such that $(D^t, \alpha^t) \in \operatorname{argmin}_{D \in \Delta\mathcal{H}, \alpha \in [0,1]^{n^2}} \mathcal{L}(D, \alpha, \lambda^t, \tau^t)$. In Lemma 12, we separate the optimization problem into two: one optimization over hypothesis D and one over violation factor α. In Lemma 14, the primal player updates the hypothesis D by leveraging a CSC oracle. Given λ^t, we can set the costs as follows

$$c_i^0 = \frac{1}{n}\mathbb{E}_{h \sim D}\left[\mathbb{1}(y_i \neq 0)\right] \qquad\qquad c_i^1 = \frac{1}{n}\mathbb{E}_{h \sim D}\left[\mathbb{1}\left(y_i \neq 1\right)\right] + (\lambda_{ij}^t - \lambda_{ji}^t).$$

Then, $D^t = h^t = CSC\left(\{(x_i, c_i^0, c_i^1)\}_{i=1}^n\right)$ (we note that the best response is always a deterministic classifier h^t).

As for α^t, we show in Lemma 13 that the primal player sets $\alpha_{ij}^t = 1$ if $\tau^t \frac{w_{ij}}{|A|} - \lambda_{ij}^t \leq 0$ and 0 otherwise. We provide the pseudo-code in Algorithm 5.

Algorithm 5 Best Response, $BEST_\rho(\lambda, \tau)$, for the primal player

```
Input: training examples $S = \{x_i, y_i\}_{i=1}^n$, $\lambda \in \Lambda$, $\tau \in \mathcal{T}$, CSC oracle $CSC$
for $i = 1, \ldots, n$ do
    if $y_i = 0$ then
        Set $c_i^0 = 0$
        Set $c_i^1 = \frac{1}{n} + \sum_{j \neq i} \lambda_{ij} - \lambda_{ji}$
    end
    else
        Set $c_i^0 = \frac{1}{n}$
        Set $c_i^1 = \sum_{j \neq i} \lambda_{ij} - \lambda_{ji}$
    end
end
$D = CSC(S, c)$
for $(i, j) \in [n]^2$ do
    $\alpha_{ij} = \begin{cases} 1: & \tau\frac{w_{ij}}{|A|} - \lambda_{ij} \leq 0 \\ 0: & \tau\frac{w_{ij}}{|A|} - \lambda_{ij} > 0. \end{cases}$
end
Output: $D, \alpha$
```

Lemma 12. *For fixed λ, τ, the best response optimization for the primal player is separable, i.e.*

$$\operatorname*{argmin}_{D,\alpha} \mathcal{L}(D, \alpha, \lambda, \tau) = \operatorname*{argmin}_{D} \mathcal{L}^{\rho_1}_{\lambda,\tau}(D) \times \operatorname*{argmin}_{\alpha} \mathcal{L}^{\rho_2}_{\lambda,\tau}(\alpha),$$

where

$$\mathcal{L}^{\rho_1}_{\lambda,\tau}(D) = err(h, D) + \sum_{(i,j)\in[n]^2} \lambda_{ij} \mathop{\mathbb{E}}_{h\sim D}\left[h(x_i) - h(x_j)\right]$$

and

$$\mathcal{L}^{\rho_2}_{\lambda,\tau}(\alpha) = \sum_{(i,j)\in[n]^2} \lambda_{ij} \left(-\alpha_{ij}\right) + \tau \left(\frac{1}{|A|} \sum_{(i,j)\in[n]^2} w_{ij}\alpha_{ij} \right)$$

Lemma 13. *For fixed λ and τ, the output α from $BEST_\rho(\lambda, \tau)$ minimizes $\mathcal{L}^{\rho_2}_{\lambda,\tau}$*

Proof. The optimization

$$\begin{aligned}
\operatorname*{argmin}_{\alpha} \mathcal{L}^{\rho_2}_{\lambda,\tau} &= \operatorname*{argmin}_{\alpha} \sum_{(i,j)\in[n]^2} \lambda_{ij} \left(-\alpha_{ij}\right) + \tau \left(\frac{1}{|A|} \sum_{(i,j)\in[n]^2} w_{ij}\alpha_{ij} \right) \\
&= \operatorname*{argmin}_{\alpha} \sum_{(i,j)\in[n]^2} -\lambda_{ij}\alpha_{ij} + \sum_{(i,j)\in[n]^2} \tau \frac{w_{ij}}{|A|}\alpha_{ij} \\
&= \operatorname*{argmin}_{\alpha} \sum_{(i,j)\in[n]^2} \alpha_{ij} \left(\tau \frac{w_{ij}}{|A|} - \lambda_{ij} \right).
\end{aligned}$$

Note that for any pair $(i, j) \in [n]^2$, the term $\alpha_{ij} \in [0, 1]$. Thus, when the constant $\tau\frac{w_{ij}}{|A|} - \lambda_{ij} \leq 0$, we assign α_{ij} as the maximum bound, 1, in order to minimize $\mathcal{L}_{\rho_2}$. Otherwise, when $\tau\frac{w_{ij}}{|A|} - \lambda_{ij} > 0$, we assign α_{ij} as the minimum bound, 0. □

Lemma 14. *For fixed λ and τ, the output D from $BEST_\rho(\lambda, \tau)$ minimizes $\mathcal{L}^{\rho_1}_{\lambda,\tau}$*

Proof.

$$\begin{aligned}
&\operatorname*{argmin}_{D} \mathcal{L}^{\rho_1}_{\lambda,\tau} \\
&= \operatorname*{argmin}_{D} err(D,S) + \sum_{(i,j)\in[n]^2} \lambda_{ij} \mathop{\mathbb{E}}_{h\sim D} [h(x_i) - h(x_j)] \\
&= \operatorname*{argmin}_{D} \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{h\sim D}\left[\mathbb{1}(h(x_i) \neq y_i)\right] + \sum_{(i,j)\in[n]^2} \lambda_{ij} \mathop{\mathbb{E}}_{h\sim D}[h(x_i) - h(x_j)] \\
&= \operatorname*{argmin}_{D} \sum_{i=1}^{n} \left(\frac{1}{n}\mathbb{E}_{h\sim D}\left[\mathbb{1}(h(x_i) \neq y_i)\right] + \sum_{j\neq i} \lambda_{ij} h(x_i) - \sum_{j\neq i} \lambda_{ji} h(x_i) \right) \\
&= \operatorname*{argmin}_{D} \sum_{i=1}^{n} \left(\frac{1}{n}\mathbb{E}_{h\sim D}\left[\mathbb{1}(h(x_i) \neq y_i)\right] + \sum_{j\neq i} h(x_i)\left(\lambda_{ij} - \lambda_{ji}\right) \right).
\end{aligned}$$

For each $i \in [n]$, we assign the cost

$$c_i^{h(x_i)} = \frac{1}{n}\mathbb{E}_{h\sim D}\left[\mathbb{1}(h(x_i) \neq y_i)\right] + h(x_i)\left(\lambda_{ij} - \lambda_{ji}\right).$$

Note that the cost depends on whether $y_i = 0$ or 1. For example, take $y_i = 1$ and $h(x_i) = 0$. The cost

$$\begin{aligned}
c_i^{h(x_i)} = c_i^0 &= \frac{1}{n}\mathbb{E}_{h\sim D}\left[\mathbb{1}(h(x_i) \neq y_i)\right] + \sum_{j\neq i} h(x_i)\left(\lambda_{ij} - \lambda_{ji}\right) \\
&= \frac{1}{n}\cdot 1 + \sum_{j\neq i} 0 \cdot \left(\lambda_{ij} - \lambda_{ji}\right) = \frac{1}{n}
\end{aligned}$$

□

4.4.3. The Dual Player's No-regret Updates

In order to reason about convergence we need to restrict the dual player's action space to lie within a bounded ℓ_1 ball, defined by the parameters C_τ and C_λ that appear in our theorem

— and serve to trade off running time with approximation quality:

$$\Lambda = \left\{\lambda \in \mathbb{R}_+^{n^2} : \|\lambda\|_1 \le C_\lambda\right\}, \mathcal{T} = \{\tau \in \mathbb{R}_+ : \|\tau\|_1 \le C_\tau\}.$$

The dual player will use exponentiated gradient descent [62] to update λ and online gradient descent [102] to update τ, where the reward function will be defined as:

$$r_\lambda(\lambda^t) = \sum_{(i,j)\in[n]^2} \lambda_{ij}^t \left(\mathop{\mathbb{E}}_{h\sim D}[h(x_i) - h(x_j)] - \alpha_{ij} - \gamma\right)$$

and

$$r_\lambda(\tau^t) = \tau^t \left(\frac{1}{|A|}\sum_{(i,j)\in[n]^2} w_{ij}\alpha_{ij} - \eta\right).$$

We provide the pseudo-code in Algorithm 6 but defer some of the proofs to Appendix 4.A.

Algorithm 6 No-Regret Dynamics

```
Input: training examples $\{x_i, y_i\}_{i=1}^n$, bounds $C_\lambda$ and $C_\tau$, time horizon $T$, step sizes $\mu_\lambda$ and $\{\mu_\tau^t\}_T^{t=1}$
Set $\theta_1^0 = \mathbf{0} \in \mathbb{R}^{n^2}$
 Set $\tau^0 = 0$
 for $t = 1, 2, \ldots, T$ do
     Set $\lambda_{ij}^t = C_\lambda \frac{\exp\theta_{ij}^{t-1}}{1+\sum_{i',j'\in[n]^2}\exp\theta_{i'j'}^{t-1}}$ for all pairs $(i,j) \in [n]^2$
     Set $\tau^t = proj_{[0,C_\tau]}\left(\tau^{t-1} + \mu_\tau^t\left(\frac{1}{|A|}\sum_{i,j} w_{ij}\alpha_{ij}^{t-1} - \eta\right)\right)$
     $D^t, \alpha^t \leftarrow \mathrm{BEST}_\rho(\lambda^t, \tau^t)$
     for $(i,j) \in [n]^2$ do
         $\theta_{ij}^t = \theta_{ij}^{t-1} + \mu_\lambda^{t-1}\left(\mathbb{E}_{h\sim D^t}[h(x_i) - h(x_j)] - \alpha_{ij}^t - \gamma\right)$
     end
end
Output: $\frac{1}{T}\sum_{t=1}^T D^t$
```

Lemma 15. *For fixed D and α, the best response optimization for the dual player is separable, i.e.*

$$\mathop{\mathrm{argmax}}_{\lambda\in\Lambda,\tau\in\mathcal{T}} \mathcal{L}(D,\alpha,\lambda,\tau) = \mathop{\mathrm{argmax}}_{\lambda\in\Lambda} \mathcal{L}_{D,\alpha}^{\psi_1}(\lambda) \times \mathop{\mathrm{argmax}}_{\tau\in\mathcal{T}} \mathcal{L}_{D,\alpha}^{\psi_2}(\tau),$$

where

$$\mathcal{L}^{\psi_1}_{D,\alpha}(\lambda) = \sum_{(i,j)\in[n]^2} \lambda_{ij} \left(\mathop{\mathbb{E}}_{h\sim D}[h(x_i) - h(x_j)] - \alpha_{ij} - \gamma \right)$$

and

$$\mathcal{L}^{\psi_2}_{D,\alpha}(\tau) = \tau \left(\frac{1}{|A|} \sum_{(i,j)\in[n]^2} w_{ij}\alpha_{ij} - \eta \right).$$

Lemma 16. *Running online gradient descent for τ^t, i.e.*

$$\tau^t = proj_{[0,C_\tau]}\left(\tau^{t-1} + \mu^{t-1} \cdot \nabla\mathcal{L}^{\psi_2}_{D^t,\alpha^t}\left(\tau^{t-1}\right)\right),$$

with step size $\mu^t = \frac{C_\tau}{\sqrt{T}}$ yields the following regret

$$\max_{\tau\in\mathcal{T}} \sum_{t=1}^{T} \mathcal{L}^{\psi_2}_{D^t,\alpha^t}(\tau) - \sum_{t=1}^{T} \mathcal{L}^{\psi_2}_{D^t,\alpha^t}\left(\tau^t\right) \le C_\tau\sqrt{T}.$$

Proof. First, note that $\nabla\mathcal{L}^{\psi_2}_{D^t,\alpha^t}\left(\tau^{t-1}\right) = \frac{1}{W}\sum_{ij} w_{ij}\alpha^{t-1}_{ij} - \eta$ and

$$\tau^t = proj_{[0,C_\tau]}\left(\tau^{t-1} + \mu^t_\tau\left(\frac{1}{W}\sum_{ij} w_{ij}\alpha^{t-1}_{ij} - \eta\right)\right).$$

From [102], we find that the regret of this online gradient descent (translated into our notations) is bounded as follows:

$$\max_{\tau\in\mathcal{T}} \sum_{t=1}^{T} \mathcal{L}^{\psi_2}_{D^t,\alpha^t}(\tau) - \sum_{t=1}^{T} \mathcal{L}^{\psi_2}_{D^t,\alpha^t}\left(\tau^t\right) \le \frac{C_\tau^2}{2\mu^T_\tau} + \frac{\left|\left|\nabla\mathcal{L}^{\psi_2}_{D,\alpha}\right|\right|^2}{2}\sum_{t=1}^{T}\mu^t_\tau, \tag{4.4}$$

where the bound on our target τ term is C_τ, the gradient of our cost function at round t is $\nabla\mathcal{L}^{\psi_2}_{D^t,\alpha^t}\left(\tau^{t-1}\right)$, and the bound $\left|\left|\nabla\mathcal{L}^{\psi_2}_{D,\alpha}\right|\right| = \sup_{\tau\in\mathcal{T},\ t\in[T]}\left|\left|\nabla\mathcal{L}^{\psi_2}_{D^t,\alpha^t}\left(\tau^{t-1}\right)\right|\right|$. To prove the above lemma, we first need to show that this bound $\left|\left|\nabla\mathcal{L}^{\psi_2}_{D,\alpha}\right|\right| \le 1$.

Since $w_{ij}, \alpha_{ij}, \eta \in [0,1]$ for all pairs (i,j), the Lagrangian $\frac{1}{|A|}\sum_{ij} w_{ij}\alpha_{ij} - \eta = \frac{\sum_{ij} w_{ij}\alpha_{ij}}{|A|} - \eta \le$

1. For all t, the gradient

$$\left|\nabla \mathcal{L}^{\psi_2}_{D^t,\alpha^t}\left(\tau^{t-1}\right)\right| = \frac{\sum_{ij} w_{ij}\alpha^{t-1}_{ij}}{|A|} - \eta \leq 1.$$

Thus,

$$\left|\nabla \mathcal{L}^{\psi_2}_{D,\alpha}\right| \leq 1.$$

Note that if we define $\mu^t_\tau = \frac{C_\tau}{\sqrt{T}}$, then the summation of the step sizes is equal to

$$\sum_{t=1}^{T} \mu^t_\tau = C_\tau \sqrt{T}$$

Substituting these two results into inequality (4.4), we get that the regret

$$\max_{\tau\in\mathcal{T}} \sum_{t=1}^{T} \mathcal{L}^{\psi_2}_{D^t,\alpha^t}(\tau) - \sum_{t=1}^{T} \mathcal{L}^{\psi_2}_{D^t,\alpha^t}\left(\tau^t\right) \leq \frac{C_\tau^2}{2\left(C_\tau \ /\sqrt{T}\right)} + \frac{1}{2} C_\tau \sqrt{T} = C_\tau \sqrt{T}$$

□

Lemma 17. *Running exponentiated gradient descent for λ^t yields the following regret:*

$$\max_{\lambda\in\Lambda} \sum_{t=1}^{T} \mathcal{L}^{\psi_1}_{D^t,\alpha^t}(\lambda) - \sum_{t=1}^{T} \mathcal{L}^{\psi_1}_{D^t,\alpha^t}\left(\lambda^t\right) \leq 2C_\lambda \sqrt{T \log n}.$$

Proof. In each round, the dual player gets to charge either some (i, j) constraint or no constraint at all. In other words, he is presented with $n^2 + 1$ options. Therefore, to account for the option of not charging any constraint, we define vector $\lambda' = (\lambda, 0)$, where the last coordinate, which will always be 0, corresponds to the option of not charging any constraint.

Next, we define the reward vector ζ^t for λ'^t as

$$\zeta^t = \left(\left(\mathop{\mathbb{E}}_{h\sim D^t}\left[h(x_i) - h(x_j)\right] - \alpha^t_{ij} - \gamma\right)_{i,j\in[n]^2}, 0\right).$$

Hence, the reward function is

$$r(\lambda'^t) = \zeta^t \cdot \lambda'^t = \mathcal{L}^{\psi_1}_{D^t,\alpha^t}\left(\lambda^t\right).$$

The gradient of the reward function is

$$\nabla r(\lambda'^t) = \left(\left(\nabla r(\lambda^t)\right)_{i,j\in[n^2]}, 0\right) = \left(\zeta^t, 0\right)$$

Note that the L-∞ norm of the gradient is bounded by 1, i.e.

$$\left|\left|\nabla r(\lambda'^t)\right|\right|_\infty \leq 1$$

because for any t, each respective component of the gradient, $\mathop{\mathbb{E}}_{h\sim D^t}\left[h(x_i) - h(x_j)\right] - \alpha^t_{ij} - \gamma$, is bounded by 1.

Here, by the regret bound of [62], we obtain the following regret bound:

$$\begin{aligned}&\max_{\lambda\in\Lambda}\sum_{t=1}^{T}\mathcal{L}^{\psi_1}_{D^t,\alpha^t}(\lambda) - \sum_{t=1}^{T}\mathcal{L}^{\psi_1}_{D^t,\alpha^t}(\lambda^t)\\ &\leq \frac{\log n}{\mu} + \mu\left|\left|\lambda'\right|\right|_1^2\left|\left|\nabla r(\lambda')\right|\right|_\infty^2 T\\ &\leq \frac{\log n}{\mu} + \mu C_\lambda^2 T.\end{aligned}$$

If we take $\mu = \frac{1}{C_\lambda}\sqrt{\frac{\log n}{T}}$, the regret is bounded as follows:

$$\max_{\lambda\in\Lambda}\sum_{t=1}^{T}\mathcal{L}^{\psi_1}_{D^t,\alpha^t}(\lambda) - \sum_{t=1}^{T}\mathcal{L}^{\psi_1}_{D^t,\alpha^t}(\lambda^t) \leq 2C_\lambda\sqrt{T\log n}. \tag{4.5}$$

□

Remark 4. *If the primal learner's approximate best response satisfies*

$$\sum_{t=1}^{T} \mathcal{L}\left(D^t, \alpha^t, \lambda^t, \tau^t\right) - \min_{D \in \Delta(H), \alpha \in [0,1]^{n^2}} \sum_{t=1}^{T} \mathcal{L}\left(D, \alpha, \lambda^t, \tau^t\right) \leq \xi_\rho T$$

along with dual player's regret of $\xi_\rho T$*, then* $\left(\bar{D}, \bar{\alpha}, \bar{\lambda}, \bar{\tau}\right)$ *is an* $(\xi_\rho + \xi_\psi)$*-approximate solution*

Theorem 12. *Let* $\left(\hat{D}, \hat{\alpha}, \hat{\lambda}, \hat{\tau}\right)$ *be a* v*-approximate solution to the Lagrangian problem. More specifically,*

$$\mathcal{L}\left(\hat{D}, \hat{\alpha}, \hat{\lambda}, \hat{\tau}\right) \leq \min_{D \in \Delta(\mathcal{H}), \alpha \in [0,1]^{n^2}} \mathcal{L}\left(D, \alpha, \hat{\lambda}, \hat{\tau}\right) + v,$$

and

$$\mathcal{L}(\hat{D}, \hat{\alpha}, \hat{\lambda}, \hat{\tau}) \geq \max_{\lambda \in \Lambda, \tau \in \mathcal{T}} \mathcal{L}\left(\hat{D}, \hat{\alpha}, \lambda, \tau\right) - v.$$

Then, $err\left(\hat{D}, S\right) \leq OPT + 2v$*. And as for the constraints, we have*

$$\mathop{\mathbb{E}}_{h \sim \hat{D}} [h(x_i) - h(x_j)] \leq \hat{\alpha}_{ij} + \gamma + \frac{1+2v}{C_\lambda}, \forall (i,j) \in [n]^2$$

$$\frac{1}{|A|} \sum_{(i,j) \in [n]^2} \hat{w}_{ij} \hat{\alpha}_{ij} \leq \eta + \frac{1+2v}{C_\tau}.$$

Proof. Let $(D^*, \alpha^*) = \text{argmin}_{(D,\alpha) \in \Omega(S, \hat{w}, \gamma, \eta)} \, err(D, S)$, the optimal solution to the Fair ERM. Also, define

$$\begin{aligned} &penalty_{S,w}\left(D, \alpha, \lambda, \tau\right) \\ &:= \sum_{(i,j)} \lambda_{ij} \left(\mathop{\mathbb{E}}_{h \sim D} [h(x_i) - h(x_j)] - \alpha_{ij} - \gamma \right) + \tau \left(\frac{1}{|A|} \sum_{(i,j)} \hat{w}_{ij} \alpha_{ij} - \eta \right). \end{aligned}$$

Note that for any D and α, $\max_{\lambda \in \Lambda, \tau \in \mathcal{T}} penalty_{S,\hat{w}}(D, \alpha, \lambda, \tau) \geq 0$ because one can always set $\lambda = 0$ and $\tau = 0$.

$$\begin{aligned}
\max_{\lambda \in \Lambda, \tau \in \mathcal{T}} \mathcal{L}\left(\hat{D}, \hat{\alpha}, \lambda, \tau\right) &\leq \mathcal{L}\left(\hat{D}, \hat{\alpha}, \hat{\lambda}, \hat{\tau}\right) + v \\
&\leq \min_{D \in \Delta(\mathcal{H}), \alpha \in [0,1]^{n^2}} \mathcal{L}\left(D, \alpha\hat{\lambda}, \hat{\tau}\right) + 2v \\
&\leq \mathcal{L}\left(D^*, \alpha^*, \hat{\lambda}, \hat{\tau}\right) + 2v \\
&= err\left(D^*, S\right) + penalty_{S,\hat{w}}\left(D^*, \alpha^*, \hat{\lambda}, \hat{\tau}\right) + 2v \\
&\leq err\left(D^*, S\right) + 2v
\end{aligned}$$

The first inequality and the third inequality are from the definition of v-approximate saddle point, and the second to last equality comes from the fact that (D^*, a^*) is a feasible solution.

Now, we consider two cases when$(\hat{D}, \hat{\alpha})$ is a feasible solution and when it's not.

1. $\left(\hat{D}, \hat{\alpha}\right) \in \Omega\left(S, \hat{w}, \gamma, \eta\right)$
   In this case, $\max_{\lambda \in \Lambda, \tau \in \mathcal{T}} penalty_{S,\hat{w}}\left(\hat{D}, \hat{\alpha}, \lambda, \tau\right) = 0$ because by the definition of being a feasible solution, we have $\mathbb{E}_{h \sim D}\left[h(x_i) - h(x_j)\right] \leq \alpha_{ij} + \gamma, \forall (i,j) \in [n]^2$ and $\frac{1}{|A|}\sum_{(i,j) \in [n]^2} \hat{w}_{ij}\alpha_{ij} \leq \eta$. Hence, $\max_{\lambda \in \Lambda, \tau \in \mathcal{T}} \mathcal{L}\left(\hat{D}, \hat{\alpha}, \lambda, \tau\right) = err\left(\hat{D}, S\right)$. Therefore, we have $err\left(\hat{D}, S\right) \leq err\left(D^*, S\right) + 2v$.

2. $\left(\hat{D}, \hat{\alpha}\right) \notin \Omega\left(S, \hat{w}, \gamma, \eta\right)$

   $$\max_{\lambda \in \Lambda, \tau \in \mathcal{T}} \mathcal{L}\left(\hat{D}, \hat{\alpha}, \lambda, \tau\right) = err\left(\hat{D}, S\right) + \max_{\lambda \in \Lambda, \tau \in \mathcal{T}} penalty_{S,\hat{w}}\left(\hat{D}, \hat{\alpha}, \lambda, \tau\right).$$

   Therefore, $err\left(\hat{D}, S\right) \leq err\left(D^*, S\right) + 2v$ because

   $$\max_{\lambda \in \Lambda, \tau \in \mathcal{T}} penalty_{S,\hat{w}}\left(\hat{D}, \hat{\alpha}, \lambda, \tau\right) \geq 0.$$

   Now, we show that even when $(\hat{D}, \hat{\alpha})$ is not a feasible solution, the constraints are violated only by so much. Note that

$$\begin{aligned}
&\max_{\lambda\in\Lambda,\tau\in\mathcal{T}} \mathcal{L}(\hat{D},\hat{\alpha},\lambda,\tau) \\
&= err(\hat{D},S) + \max_{\lambda\in\Lambda,\tau\in\mathcal{T}} penalty_{S,\hat{w}}(\hat{D},\hat{\alpha},\lambda,\tau) \leq err(D^*,S) + 2v
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\max_{\lambda\in\Lambda,\tau\in\mathcal{T}} penalty_{S,\hat{w}}(\hat{D},\hat{\alpha},\hat{\lambda},\hat{\tau}) \leq err(D^*,S) - err(\hat{D},S) + 2v \\
&\max_{\lambda\in\Lambda,\tau\in\mathcal{T}} penalty_{S,\hat{w}}(\hat{D},\hat{\alpha},\hat{\lambda},\hat{\tau}) \leq 1 + 2v
\end{aligned}$$

Let $\lambda^*, \tau^* = BEST_\psi\left(\hat{D},\hat{\alpha}\right)$, which minimizes the function as shown in Lemma 18 and 19. Now, consider

$$\sum_{(i,j)} \lambda^*_{ij}\left(\mathop{\mathbb{E}}_{h\sim D}\left[h(x_i)-h(x_j)\right]-\alpha_{ij}-\gamma\right)+\tau^*\left(\frac{1}{|A|}\sum_{(i,j)}\hat{w}_{ij}\alpha_{ij}-\eta\right)\leq 1+2v$$

Say $(i^*,j^*) = \text{argmax}_{(i,j)\in[n^2]} \mathop{\mathbb{E}}_{h\sim D}\left[h(x_i)-h(x_j)\right]-\alpha_{ij}-\gamma$. Remember that if

$$\mathop{\mathbb{E}}_{h\sim D}\left[h(x_{i^*})-h(x_{j^*})\right]-\alpha_{i^*j^*}-\gamma>0,$$

then $\lambda^*_{i^*j^*} = C_\tau$ and 0 for the other coordinates and else, it's just a zero vector. Also, $\tau = C_\tau$ if $\sum_{(i,j)}\hat{w}_{ij}\alpha_{ij}-\eta>0$ and 0 otherwise. Thus,

$$\sum_{(i,j)} \lambda^*_{ij}\left(\mathop{\mathbb{E}}_{h\sim D}\left[h(x_i)-h(x_j)\right]-\alpha_{ij}-\gamma\right)\geq 0$$

$$\tau^*\left(\frac{1}{|A|}\sum_{(i,j)}\hat{w}_{ij}\alpha_{ij}-\eta\right)\geq 0$$

Therefore, we have

$$\max_{i,j\in[n]^2}\left(\mathop{\mathbb{E}}_{h\sim D}[h(x_i)-h(x_j)]-\alpha_{ij}-\gamma\right)\leq\frac{1+2v}{C_\lambda},$$

and

$$\frac{1}{|A|}\sum_{(i,j)\in[n]^2}\hat{w}_{ij}\hat{\alpha}_{ij}\leq\eta+\frac{1+2v}{C_\tau}$$

□

Now, the proof of Theorem 11 is simply plugging in the best response guarantee of the learner, Lemma 13 and 14, and the no-regret guarantee of the auditor, Lemma 16 and 17, into Theorem 12. We defer the actual proof to Appendix 4.A.

4.5. Generalization

In this section, we show that fairness loss generalizes out-of-sample. (Error generalization follows from the standard VC-dimension bound, which — because it is a uniform convergece statement is unaffected by the addition of fairness constraints. See Appendix 4.B for the standard statement.)

Proving that the fairness loss generalizes doesn't follow immediately from a standard VC-dimension argument for several reasons: it is not linearly separable, but defined as an average over non-disjoint *pairs* of individuals in the sample. The difference between empirical fairness loss and true fairness loss of a randomized hypothesis $D\in\Delta\mathcal{H}$ is also a non-convex function of the supporting hypotheses h, and so it is not sufficient to prove a uniform convergence bound merely for the base hypotheses in our hypothesis class $\mathcal{H}$. We circumvent these difficulties by making use of an ϵ-net argument, together with an application of a concentration inequality, and an application of Sauer's lemma. Briefly, we show that with respect to fairness loss, the continuous set of distributions over classifiers have an ϵ-net of sparse distributions. Using the two-sample trick and Sauer's lemma, we can bound the number of such sparse distributions. The end result is the following generalization theorem:

Theorem 13. *Let S consists of n i.i.d points drawn from $\mathcal{P}$ and let M represent a set of m pairs randomly drawn from $S \times S$. Then we have:*

$$\Pr_{\substack{S\sim\mathcal{P}^n \\ M\sim(S\times S)^m}}\left(\sup_{D\in\Delta\mathcal{H}}\left|\Pi_{D,w,\gamma}(M) - \mathop{\mathbb{E}}_{(x,x')\sim\mathcal{P}^2}\left[\Pi_{D,w,\gamma}(x,x')\right]\right| > 2\epsilon\right)$$
$$\leq \left(8\cdot\left(\frac{e\cdot 2n}{d}\right)^{dk}\exp\left(\frac{-n\epsilon^2}{32}\right) + \left(\frac{e\cdot 2n}{d}\right)^{dk'}\exp\left(-8m\epsilon^2\right)\right),$$

where $k' = \frac{2\ln(2m)}{\epsilon^2} + 1$, $k = \frac{\ln(2n^2)}{8\epsilon^2} + 1$, and d is the VC-dimension of $\mathcal{H}$.

See Appendix 4.B for the proof. To interpret this theorem, note that the right hand side (the probability of a failure of generalization) begins decreasing exponentially fast in the data and fairness constraint sample parameters n and m as soon as $n \geq \Omega(d\log(n)\log(n/d))$ and $m \geq \Omega(d\log(m)\log(n/d))$.

4.6. A Behavioral Study

The framework and algorithm we have provided can be viewed as a tool to elicit and enforce a notion of fairness defined by a collection of stakeholders. In this section, we describe preliminary results from a human-subject study we performed in which pairwise fairness preferences were elicited and enforced by our algorithm.

We note that the subjects included in our empirical study were not stakeholders affected by the algorithm we used (the COMPAS algorithm). Thus, our results should not be interpreted as cogent for any policy modifications to the COMPAS algorithm. We instead report our empirical findings primarily to showcase the performance of our algorithm and to act as a template for what should be reported if our framework were applied with relevant stakeholders (for example, if fairness preferences about COMPAS data were elicited from inmates).[9]

The framework and algorithm we have provided can be viewed as a potentially powerful tool for empirically studying subjective individual fairness as a *behavioral* phenomenon.

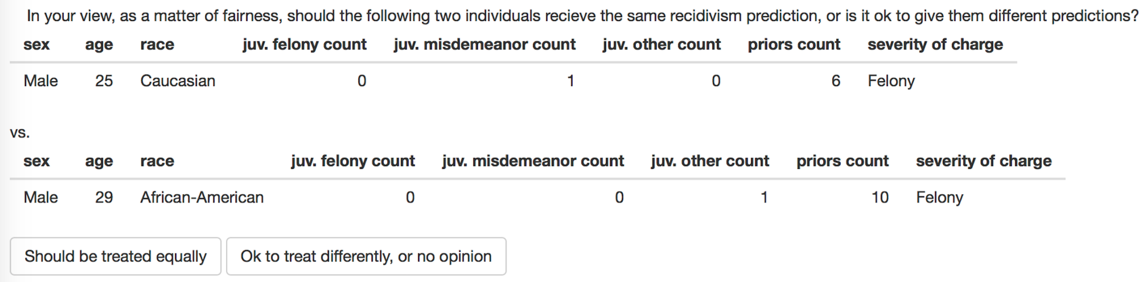
[9] We omit such an empirical study due to the difficulty of accessing such stakeholders and leave this for future work.

In this section we describe preliminary results from a human-subject study we performed in which subjective fairness was elicited and then enforced by our algorithm.

4.6.1. Data

Our study used the COMPAS recidivism data gathered by ProPublica [10] in their celebrated analysis of Northepointe's risk assessment algorithm [4]. This data consists of defendants from Broward County in Florida between 2013 to 2014. For each defendant the data consists of sex (male, female), age (18-96), race (African-American, Caucasian, Hispanic, Asian, Native American), juvenile felony count, juvenile misdemeanor count, number of other juvenile offenses, number of prior adult criminal offenses, the severity of the crime for which they were incarcerated (felony or misdemeanor), as well as the outcome of whether or not they did in fact recidivate. Recidivism is defined as a new arrest within 2 years, not counting traffic violations and municipal ordinance violations.

4.6.2. Subjective Fairness Elicitation



In your view, as a matter of fairness, should the following two individuals recieve the same recidivism prediction, or is it ok to give them different predictions?

sex	age	race	juv. felony count	juv. misdemeanor count	juv. other count	priors count	severity of charge
Male	25	Caucasian	0	1	0	6	Felony

vs.

sex	age	race	juv. felony count	juv. misdemeanor count	juv. other count	priors count	severity of charge
Male	29	African-American	0	0	1	10	Felony

Should be treated equally
Ok to treat differently, or no opinion

Figure 3: Screenshot of sample subjective fairness elicitation question posed to human subjects.

We implemented our fairness framework via a web app that elicited subjective fairness notions from 43 undergraduates at a major research university. After reading a document describing the data and recidivism prediction task, each subject was presented with 50 randomly chosen pairs of records from the COMPAS data set and asked whether in their opinion the two individuals should treated (predicted) equally or not. Importantly, the subjects were shown only the features for the individuals, and not their actual recidivism

[10] The data can be accessed on ProPublica's Github page here. We cleaned the data as in the ProPublica study, removing any records with missing data. This left 5829 records, where the base rate of two-year recidivism was 46%.

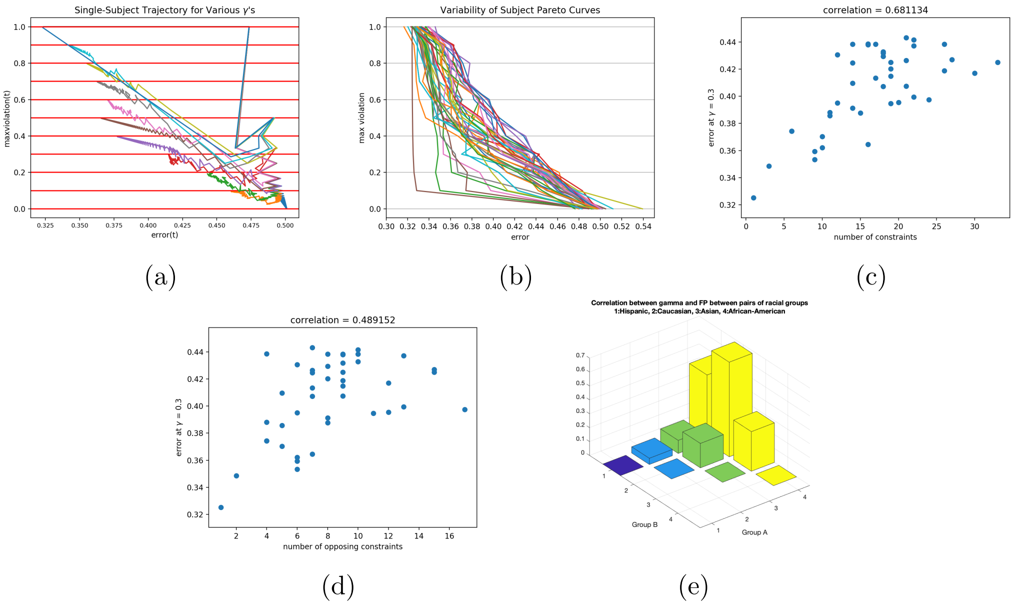
outcomes, since we sought to elicit subjects' fairness notions regarding the predictions of those outcomes. While absolutely no guidance was given to subjects regarding fairness, the elicitation framework allows for rich possibilities. For example, subjects could choose to ignore demographic factors or criminal histories entirely if they liked, or a subject who believes that minorities are more vulnerable to overpolicing could discount their criminal histories relative to Caucasians in their pairwise elicitations.

For each subject, the pairs they identified to be treated equally were taken as constraints on error minimization with respect to the actual recidivism outcomes over the entire COMPAS dataset, and our algorithm was applied to solve this constrained optimization problem, using a linear threshold heuristic as the underlying learning oracle [59]. We ran our algorithm with $\eta = 0$ and variable γ in Equations (4.1) through (4.3), which represents the strongest enforcement of subjective fairness — the difference in predicted values must be at most γ on *every* pair selected by a subject. Because the issues we are most interested in here (convergence, tradeoffs with accuracy, and heterogeneity of fairness preferences) are orthogonal to generalization — and because we prove VC-dimension based generalization theorems — for simplicity, the results we report are in-sample.

4.6.3. Results

Since our algorithm relies on a learning heuristic for which worst-case guarantees are not possible, the first empirical question is whether the algorithm converges rapidly on the behavioral data. We found that it did so consistently; a typical example is Figure 4a, where we show the trajectories of model error vs. fairness violation for a particular subject's data for variable values of the input γ (horizontal lines). After 1000 iterations, the algorithm has converged to the optimal errors subject to the allowed γ.

Perhaps the most basic behavioral questions we might ask involve the extent and nature of subject variability. For example, do some subjects identify constraint pairs that are much harder to satisfy than other subjects? And if so, what factors seem to account for such variation?



(a) (b) (c)

(d) (e)

Figure 4: (a) Sample algorithm trajectory for a particular subject at various γ. (b) Sample subjective fairness Pareto curves for a sample of subjects. (c) Scatterplot of number of constraints specified and number of opposing constraints vs. error at $\gamma = 0.3$. (d) Scatterplot of number of constraints where the true labels are different vs. error at $\gamma = 0.3$. (e) Correlation between false positive rate difference and γ for racial groups.

Figure 4b shows that there is indeed considerable variation in subject difficulty. For each of the 43 subjects, we have plotted the error vs. fairness violation Pareto curves obtained by varying γ from 0 (pairs selected by subjects must have identical probabilistic predictions of recidivism) to 1.0 (no fairness enforced whatsoever). Since our model space is closed under probabilistic mixtures, the worst-case Pareto curve is linear, obtained by all mixtures of the error-optimal model and random predictions. Easier constaint sets are more convex. We see in the figure that both extremes are exhibited behaviorally — some subjects yield linear or near-linear curves, while others permit huge reductions in unfairness for only slight increases in error, and virtually all the possibilities in between are realized as well. [11]

Since each subject was presented with 50 random pairs and was free to constrain as many or as few as they wished, it is natural to wonder if the variation in difficulty is explained simply by the number of constraints chosen. In Figure 4c we show a scatterplot of the the number

[11] The slight deviations from true convexity are due to approximate rather than exact convergence.

of constraints selected by a subject (x axis) versus the error obtained (y axis) for $\gamma = 0.3$ (an intermediate value that exhibits considerable variation in subject error rates) for all 43 subjects. While we see there is indeed strong correlation (approximately 0.69), it is far from the case that the number of constraints explains all the variability. For example, amongst subjects who selected approximately 16 constraints, the resulting error varies over a range of nearly 8%, which is over 40% of the range from the optimal error (0.32) to the worst fairness-constrained error (0.5). More surprisingly, when we consider only the 'opposing' constraints, pairs of points with different true labels, the correlation (0.489) seems to be weaker. Enforcing a classifier to predict similarly on a pair of points with different true labels should increase the error, and yet, it is less correlated with error than the raw number of constraints. This suggests that the variability in subject difficulty is due to the nature of the constraints themselves rather than their number or disagreement with the true labels.

It is also interesting to consider the collective force of the 1432 constraints selected by all 43 subjects together, which we can view as a "fairness panel" of sorts. Given that there are already individual subjects whose constraints yield the worst-case Pareto curve, it is unsurprising that the collective constraints do as well. But we can exploit the flexibility of our optimization framework in Equations (4.1) through constraint (4.3), and let $\gamma = 0.0$ and vary only η, thus giving the learner discretion in which subjects' constraints to discount or discard at a given budget η. In doing so we find that the unconstrained optimal error can be obtained while having the average (exact) pairwise constraint be violated by only roughly 25%, meaning roughly that only 25% of the collective constraints account for all the difficulty.

Finally, we can investigate the extent to which behavioral subjective fairness notions align with more standard statistical fairness definitions, such as equality of false positive rates. For instance, for each subject and a pair of racial groups, we take the absolute difference in false positive rates of the classifier at $\gamma \in \{0.0, 0.1, \ldots, 1.0\}$ and calculate the correlation coefficient between realized values of γ (which measure violation of subjective unfairness) and the false positive rate differences. Figure 4e shows the average correlation coefficient across subjects for each pair of racial groups. We note that subjective fairness correlates with a smaller gap between the false positive rates across Caucasians and African Americans: but correlates substantially less for other pairs of racial groups.

We leave a more complete investigation of our behavioral study for future work, including the detailed nature of subject variability and further comparison of behavioral subjective fairness to standard algorithmic fairness notions.

Appendix

4.A. Missing Details from Section 4.4

4.A.1. Primal Player's Best Response

Lemma 12. *For fixed λ, τ, the best response optimization for the primal player is separable, i.e.*

$$\operatorname*{argmin}_{D,\alpha} \mathcal{L}(D, \alpha, \lambda, \tau) = \operatorname*{argmin}_{D} \mathcal{L}^{\rho_1}_{\lambda,\tau}(D) \times \operatorname*{argmin}_{\alpha} \mathcal{L}^{\rho_2}_{\lambda,\tau}(\alpha),$$

where

$$\mathcal{L}^{\rho_1}_{\lambda,\tau}(D) = err(h, D) + \sum_{(i,j)\in[n]^2} \lambda_{ij} \mathop{\mathbb{E}}_{h\sim D} [h(x_i) - h(x_j)]$$

and

$$\mathcal{L}^{\rho_2}_{\lambda,\tau}(\alpha) = \sum_{(i,j)\in[n]^2} \lambda_{ij} \left(-\alpha_{ij}\right) + \tau \left(\frac{1}{|A|} \sum_{(i,j)\in[n]^2} w_{ij}\alpha_{ij} \right)$$

Proof. First, note that α is not dependent on D and vice versa. Thus, we may separate the optimization $\operatorname{argmin}_{D,\alpha} \mathcal{L}$ as such:

$$
\begin{aligned}
&\operatorname*{argmin}_{D,\alpha} \mathcal{L}(D,\alpha,\lambda,\tau) \\
&= \operatorname*{argmin}_{D,\alpha} err(D,S) + \sum_{(i,j)\in[n]^2} \lambda_{ij} \left(\mathop{\mathbb{E}}_{h\sim D} [h(x_i) - h(x_j)] - \alpha_{ij} - \gamma \right) \\
&+ \tau \left(\frac{1}{|A|} \sum_{(i,j)\in[n]^2} w_{ij}\alpha_{ij} - \eta \right) \\
&= \operatorname*{argmin}_{D} err(D,S) + \sum_{(i,j)\in[n]^2} \lambda_{ij} \mathop{\mathbb{E}}_{h\sim D} [h(x_i) - h(x_j)] \times \sum_{(i,j)\in[n]^2} \lambda_{ij} \left(-\alpha_{ij}\right) \\
&+ \tau \left(\frac{1}{|A|} \sum_{(i,j)\in[n]^2} w_{ij}\alpha_{ij} \right) \\
&= \operatorname*{argmin}_{D} \mathcal{L}^{\rho_1}_{\lambda,\tau}(D) \times \operatorname*{argmin}_{\alpha} \mathcal{L}^{\rho_2}_{\lambda,\tau}(\alpha)
\end{aligned}
$$

□

4.A.2. Dual Player's Best Response

Lemma 15. *For fixed D and α, the best response optimization for the dual player is separable, i.e.*

$$\operatorname*{argmax}_{\lambda\in\Lambda,\tau\in\mathcal{T}} \mathcal{L}(D,\alpha,\lambda,\tau) = \operatorname*{argmax}_{\lambda\in\Lambda} \mathcal{L}^{\psi_1}_{D,\alpha}(\lambda) \times \operatorname*{argmax}_{\tau\in\mathcal{T}} \mathcal{L}^{\psi_2}_{D,\alpha}(\tau),$$

where

$$\mathcal{L}^{\psi_1}_{D,\alpha}(\lambda) = \sum_{(i,j)\in[n]^2} \lambda_{ij} \left(\mathop{\mathbb{E}}_{h\sim D} [h(x_i) - h(x_j)] - \alpha_{ij} - \gamma \right)$$

and

$$\mathcal{L}^{\psi_2}_{D,\alpha}(\tau) = \tau \left(\frac{1}{|A|} \sum_{(i,j)\in[n]^2} w_{ij}\alpha_{ij} - \eta \right).$$

Proof.

$$
\begin{aligned}
&\underset{\lambda\in\Lambda,\tau\in\mathcal{T}}{\operatorname{argmax}}\,\mathcal{L}(D,\alpha,\lambda,\tau)\\
&=\underset{\lambda\in\Lambda,\tau\in\mathcal{T}}{\operatorname{argmax}}\,\underset{h\sim D}{\mathbb{E}}\left[err(h,S)\right]+\sum_{(i,j)\in[n]^2}\lambda_{ij}\left(\underset{h\sim D}{\mathbb{E}}\left[h(x_i)-h(x_j)\right]-\alpha_{ij}-\gamma\right)\\
&+\tau\left(\frac{1}{|A|}\sum_{(i,j)\in[n]^2}w_{ij}\alpha_{ij}-\eta\right)\\
&=\underset{\lambda\in\Lambda}{\operatorname{argmax}}\sum_{(i,j)\in[n]^2}\lambda_{ij}\left(\underset{h\sim D}{\mathbb{E}}\left[h(x_i)-h(x_j)\right]-\alpha_{ij}-\gamma\right)\\
&\times\underset{\tau\in\mathcal{T}}{\operatorname{argmax}}\,\tau\left(\frac{1}{|A|}\sum_{(i,j)\in[n]^2}w_{ij}\alpha_{ij}-\eta\right)\\
&=\underset{\lambda\in\Lambda}{\operatorname{argmax}}\,\mathcal{L}^{\psi_1}_{D,\alpha}(\lambda)\times\underset{\tau\in\mathcal{T}}{\operatorname{argmax}}\,\mathcal{L}^{\psi_2}_{D,\alpha}(\tau)
\end{aligned}
$$

□

Algorithm 7 Best Response, $BEST_\psi(D,\alpha)$, for the dual player

```
Input: training examples $S=\{x_i,y_i\}_{i=1}^n$, $D\in\Delta(H)$, $\alpha\in[0,1]^{n^2}$
$\lambda=0\in\mathbb{R}^{n^2}$
 $(i^*,j^*)=\text{argmax}_{(i,j)\in[n]^2}\mathbb{E}_{h\sim D}\left[h(x_i)-h(x_j)\right]-\alpha_{ij}-\gamma$
 if $\mathbb{E}_{h\sim D}\left[h(x_{i^*})-h(x_{j^*})\right]-\alpha_{i^*j^*}-\gamma\leq 0$ then
 |   $\lambda_{i^*j^*}=C_\lambda$
end
Set $\tau=\begin{cases}0 & \frac{1}{|A|}\sum_{(i,j)\in[n]^2}w_{ij}\alpha_{ij}-\eta\leq 0\\ C_\tau & o.w.\end{cases}$
Output: $\lambda,\tau$
```

Lemma 18. *For fixed D and α, the output λ from $BEST_\psi(D,\alpha)$ minimizes $\mathcal{L}^{\psi_1}_{D,\alpha}$*

Proof. Because $\mathcal{L}^{\psi_1}_{D,\alpha}$ is linear in terms of λ and the feasible region is the non-negative orthant bounded by 1-norm, the optimal solution must include putting all the weight to the pair (i,j) where $\mathbb{E}_{h\sim D}[h(x_i)-h(x_j)-\alpha_{ij}]$ is maximized. □

Lemma 19. *For fixed D and α, the output τ from $BEST_\psi(D,\alpha)$ minimizes $\mathcal{L}^{\psi_2}_{D,\alpha}$*

Proof. Because $\mathcal{L}^{\psi_2}_{D,\alpha}$ is linear in terms of τ, the optimal solution is trivially to set τ at either C_τ or 0 depending on the sign. □

4.A.3. No-Regret Dynamics

Theorem 14 (Freund and Schapire [31]). *Let $(D^1, \alpha^1), \ldots, (D^T, \alpha^T)$ be the primal player's sequence of actions, and $(\lambda^1, \tau^1), \ldots, (\lambda^T, \tau^T)$ be the dual player's sequence of actions. Let $\bar{D} = \frac{1}{T}\sum_{t=1}^T D^t$, $\bar{\alpha} = \frac{1}{T}\sum_{t=1}^T \alpha^t$, $\bar{\lambda} = \frac{1}{T}\sum_{t=1}^T \lambda^t$, and $\bar{\tau} = \frac{1}{T}\sum_{t=1}^T \tau^t$. Then, if the regret of the dual player satisfies*

$$\max_{\lambda\in\Lambda,\tau\in\mathcal{T}} \sum_{t=1}^T \mathcal{L}\left(D^t, \alpha^t, \lambda^t, \tau^t\right) - \sum_{t=1}^T \mathcal{L}\left(D^t, \alpha^t, \lambda^t, \tau^t\right) \le \xi_\psi T,$$

and the primal player best responds in each round

$$(D^t, \alpha^t) = \operatorname*{argmax}_{D\in\Delta(H),\alpha\in[0,1]^{n^2}} \mathcal{L}\left(D, \alpha, \lambda^t, \tau^t\right),$$

then $(\bar{D}, \bar{\alpha}, \bar{\lambda}, \bar{\tau})$ is an ξ_ψ-approximate solution

4.A.4. Omitted Proof of Theorem 11

Theorem 11. *Fix parameters ν, C_τ, C_λ that serve to trade off running time with approximation error. There is an efficient algorithm that makes $T = \left(\frac{2C_\lambda\sqrt{\log(n)}+C_\tau}{\nu}\right)^2$ CSC oracle calls and outputs a solution $(\hat{D}, \hat{\alpha})$ with the following guarantee. The objective value is approximately optimal:*

$$err(\hat{D}, S) \le \min_{(D,\alpha)\in\Omega(S,\hat{w},\gamma,\eta)} err(D, S) + 2\nu.$$

And the constraints are approximately satisfied: $\mathbb{E}_{h\sim\hat{D}}[h(x_i)-h(x_j)] \le \hat{\alpha}_{ij}+\gamma+\frac{1+2\nu}{C_\lambda}, \forall(i,j) \in [n]^2$ and $\frac{1}{|A|}\sum_{(i,j)\in[n]^2} \hat{w}_{ij}\hat{\alpha}_{ij} \le \eta + \frac{1+2\nu}{C_\tau}$.

Proof. Observe that

$$\mathcal{L}(D, \alpha, \lambda, \tau) = err(D, S) + \mathcal{L}^{\psi_1}_{D,\alpha}(\lambda) + \mathcal{L}^{\psi_2}_{D,\alpha}(\tau)$$

By how we constructed $\mathcal{L}_{D,\alpha}^{\psi_1}$ and $\mathcal{L}_{D,\alpha}^{\psi_2}$, combining Lemma 16 and 17 yields

$$\begin{aligned}
&\max_{\lambda\in\Lambda,\tau\in\mathcal{T}} \sum_{t=1}^{T} \mathcal{L}\left(D^t,\alpha^t,\lambda^t,\tau^t\right) - \sum_{t=1}^{T} \mathcal{L}\left(D^t,\alpha^t,\lambda^t,\tau^t\right) \\
&= \max_{\tau\in\mathcal{T}} \sum_{t=1}^{T} \mathcal{L}_{D^t,\alpha^t}^{\psi_2}(\tau) - \sum_{t=1}^{T} \mathcal{L}_{D^t,\alpha^t}^{\psi_2}\left(\tau^t\right) + \max_{\lambda\in\Lambda} \sum_{t=1}^{T} \mathcal{L}_{D^t,\alpha^t}^{\psi_1}(\lambda) - \sum_{t=1}^{T} \mathcal{L}_{D^t,\alpha^t}^{\psi_1}\left(\lambda^t\right) \\
&\le \xi_\psi T,
\end{aligned}$$

where $\xi_\psi = \frac{2C_\lambda\sqrt{T\log n}+C_\tau\sqrt{T}}{T}$.

Then, theorem 14 tells us that $\bar{D},\bar{\alpha},\bar{\lambda},\bar{\alpha}$ form a ξ_ψ-approximate equilibrium, where $\bar{D} = \frac{1}{T}\sum_{t=1}^{T} D^t$, $\bar{\alpha} = \frac{1}{T}\sum_{t=1}^{T}\alpha^t$, $\bar{\lambda} = \frac{1}{T}\sum_{t=1}^{T}\lambda^t$, and $\bar{\tau} = \frac{1}{T}\sum_{t=1}^{T}\tau^t$. And finally, with $T = \left(\frac{2C_\lambda\sqrt{\log(n)}+C_\tau}{v}\right)^2$ results in $\xi_\psi = \nu$, theorem 12 gives

$$err(\hat{D},S) \le \min_{(D,\alpha)\in\Omega(S,\hat{w},\gamma,\eta)} err(D,S) + 2\nu.$$

And as for the constraints,

$$\mathop{\mathbb{E}}_{h\sim\hat{D}}\left[h(x_i)-h(x_j)\right] \le \hat{\alpha}_{ij} + \gamma + \frac{1+2\nu}{C_\lambda}, \forall (i,j)\in[n]^2$$

and

$$\frac{1}{|A|}\sum_{(i,j)\in[n]^2} \hat{w}_{ij}\hat{\alpha}_{ij} \le \eta + \frac{1+2v}{C_\tau}.$$

□

4.B. Missing Details from Section 4.5

4.B.1. Error

Theorem 15 (Kearns and Vazirani [58]). *Fix some hypothesis class $\mathcal{H}$ and distribution $\mathcal{P}$. Let $S \sim P^n$ be a dataset consisting of n examples $\{x_i,y_i\}_{i=1}^n$ sampled i.i.d. from $\mathcal{P}$. Then,* *for any* $0 < \delta < 1$*, with probability* $1 - \delta$*, for every* $h \in \mathcal{H}$*, we have*

$$|err(h, \mathcal{P}) - err(h, S)| \leq O\left(\sqrt{\frac{VCDIM(\mathcal{H}) + log(\frac{1}{\delta})}{n}}\right)$$

4.B.2. Fairness Loss

At a high level, our argument proceeds as follows: using McDiarmid's inequality, for any *fixed* hypothesis, its empirical fairness loss concentrates around its expectation. This argument extends to an infinite family of hypotheses with bounded VC-dimension via the standard two-sample trick, together with Sauer's lemma: the only catch is that we need to use a variant of McDiarmid's inequality that applies to sampling without replacement. However, proving that the fairness loss for each fixed hypothesis h concentrates around its expectation is not sufficient to obtain the same result for arbitrary distributions over hypotheses, because the difference between a randomized classifier's fairness loss and its expectation is a non-convex function of the mixture weights. To circumvent this issue, we show that with respect to fairness loss, there is an ϵ-net consisting of sparse distributions over hypotheses. Once we apply Sauer's lemma and the two-sample trick, there are only finitely many such distributions, and we can union bound over them.

We begin by stating the standard version of McDiarmid's inequality:

Theorem 16 (McDiarmid's Inequality)**.** *Suppose* $X_1, \ldots, X_n$ *are independent and* f *satisfies*

$$\sup_{x_1,\ldots,x_n,\hat{x}_i} |f(x_1, \ldots, x_n) - f(x_1, \ldots, x_{i-1}, \hat{x}_i, x_{i+1}, \ldots, x_n)| \leq c_i.$$

Then, for any $\epsilon > 0$*,*

$$\Pr_{X^1,\ldots,X^n}\left(\left|f(X_1, \ldots, X_n) - \mathop{\mathbb{E}}_{X_1,\ldots,X_n}[f(X_1, \ldots, X_n)]\right| \geq \epsilon\right) \leq 2\exp\left(-\frac{2\epsilon^2}{\sum_{i=1}^n c_i^2}\right)$$

Lemma 20. *Fix a randomized hypothesis* $D \in \Delta\mathcal{H}$*. Over the randomness of* $S \sim \mathcal{P}^n$*, we* *have*

$$\Pr_{S\sim\mathcal{P}^n}\left(\left|\Pi_{D,w,\gamma}(S\times S)-\mathop{\mathbb{E}}_{S}\left[\Pi_{D,w,\gamma}(S\times S)\right]\right|\geq\epsilon\right)\leq 2\exp\left(-2n\epsilon^2\right)$$

Proof. Define a slightly modified fairness loss function that depends on each instance instead of a pair.

$$\Pi'_{D,w,\gamma}\left(x_1,x_2,\ldots,x_n\right)=\frac{1}{n^2}\sum_{(i,j)\in[n]^2}\Pi_{D,w,\gamma}\left((x_i,x_j)\right).$$

Note that $\Pi'_{D,w,\gamma}(x_1,\ldots,x_n)=\Pi_{D,w,\gamma}(S\times S)$. The sensitivity of $\Pi'_{D,w,\gamma}(x_1,x_2,\ldots,x_n)$ is $\frac{1}{n}$, so applying McDiarmid's inequality yields the above concentration. □

Theorem 17. *If* $n\geq\frac{2\ln(2)}{\epsilon^2}$,

$$\Pr_{S}\left(\sup_{D\in\Delta\mathcal{H}}\left|\Pi_{D,w,\gamma}(S\times S)-\mathop{\mathbb{E}}_{x,x'}\left[\Pi_{D,w,\gamma}(x,x')\right]\right|>\epsilon\right)\leq 8\cdot\left(\frac{e\cdot 2n}{d}\right)^{dk}\exp\left(\frac{-n\epsilon^2}{32}\right)$$

where d is the VC-dimension of $\mathcal{H}$, and $k=\frac{\ln(2n^2)}{8\epsilon^2}+1$.

Proof. First, by linearity of expectation, we note that $\mathbb{E}_S\left[\Pi_{D,w,\gamma}(S\times S)\right]=\mathbb{E}_{x,x'}\left[\Pi_{D,w,\gamma}(x,x')\right]$. Given S, let D^*_S be some randomized classifier such that

$$\left|\Pi_{D^*_S,w,\gamma}(S\times S)-\mathop{\mathbb{E}}_{x,x'}\left[\Pi_{D^*_S,w,\gamma}(x,x')\right]\right|>\epsilon.$$

If such hypothesis does not exist, let it be some fixed hypothesis in $\mathcal{H}$. We now use standard symmetrization argument, which allows us to bound the difference between the fairness loss of our sample S and that of another independent 'ghost' sample $S'=(x'_1,\ldots,x'_n)$ instead of bounding the difference between the empirical fairness loss and its expected fairness loss.

$$
\begin{aligned}
&\Pr_{S\sim\mathcal{P}^n, S'\sim\mathcal{P}^n}\left(\sup_{D\in\Delta\mathcal{H}} \left|\Pi_{D,w,\gamma}(S\times S) - \Pi_{D,w,\gamma}(S'\times S')\right| > \frac{\epsilon}{2}\right)\\
&\geq \Pr_{S,S'}\left(\left|\Pi_{D_S^*,w,\gamma}(S\times S) - \Pi_{D_S^*,w,\gamma}(S'\times S')\right| > \frac{\epsilon}{2}\right)\\
&\geq \Pr_{S,S'}\left(\left|\Pi_{D_S^*,w,\gamma}(S\times S) - \mathop{\mathbb{E}}_{x,x'}\left[\Pi_{D_S^*,w,\gamma}(x,x')\right]\right| > \epsilon\right.\\
&\text{and } \left.\left|\Pi_{D^*,w,\gamma}(S'\times S') - \mathop{\mathbb{E}}_{x,x'}\left[\Pi_{D^*,w,\gamma}(x,x')\right]\right| \leq \frac{\epsilon}{2}\right)\\
&= \mathop{\mathbb{E}}_{S,S'}\left[\mathbb{1}\left(\left|\Pi_{D_S^*,w,\gamma}(S\times S) - \mathop{\mathbb{E}}_{x,x'}\left[\Pi_{D_S^*,w,\gamma}(x,x')\right]\right| > \epsilon\right)\right.\\
&\left.\cdot\, \mathbb{1}\left(\left|\Pi_{D^*,w,\gamma}(S'\times S') - \mathop{\mathbb{E}}_{x,x'}\left[\Pi_{D^*,w,\gamma}(x,x')\right]\right| \leq \frac{\epsilon}{2}\right)\right]\\
&= \mathop{\mathbb{E}}_{S}\left[\mathbb{1}\left(\left|\Pi_{D_S^*,w,\gamma}(S\times S) - \mathop{\mathbb{E}}_{x,x'}\left[\Pi_{D_S^*,w,\gamma}(x,x')\right]\right| > \epsilon\right)\right.\\
&\left.\cdot\, \Pr_{S'|S}\left(\left|\Pi_{D^*,w,\gamma}(S'\times S') - \mathop{\mathbb{E}}_{x,x'}\left[\Pi_{D^*,w,\gamma}(x,x')\right]\right| \leq \frac{\epsilon}{2}\right)\right]\\
&\geq \Pr_{S}\left(\left|\Pi_{D_S^*,w,\gamma}(S\times S) - \mathop{\mathbb{E}}_{x,x'}\left[\Pi_{D_S^*,w,\gamma}(x,x')\right]\right| > \epsilon)\right)\cdot\left(1-\exp(-\frac{n\epsilon^2}{2})\right)\\
&\geq \frac{1}{2}\Pr_{S}\left(\sup_{D\in\Delta\mathcal{H}}\left|\Pi_{D,w,\gamma}(S\times S) - \mathop{\mathbb{E}}_{x,x'}\left[\Pi_{D,w,\gamma}(x,x')\right]\right| > \epsilon\right)
\end{aligned}
$$

We used Lemma 20 for the second to last inequality, and the last inequality follows from the theorem's condition and the definition of D_S^*.

Now, imagine sampling $\bar{S} = 2n$ points from $\mathcal{P}$, and uniformly choosing n points without replacement to be S and the remaining n points to be S'. This process is equivalent to sampling n points from $\mathcal{P}$ to form S and another independent set of n points from $\mathcal{P}$ to form S'.

$$\begin{aligned}&\Pr_{\bar{S},S,S'}\left(\sup_{D\in\Delta\mathcal{H}}\left|\Pi_{D,w,\gamma}(S\times S)-\Pi_{D,w,\gamma}(S'\times S')\right|>\frac{\epsilon}{2}\right)\\&=\sum_{\bar{S}}\Pr\left(\bar{S}\right)\Pr_{S,S'}\left(\sup_{D\in\Delta\mathcal{H}}\left|\Pi_{D,w,\gamma}(S\times S)-\Pi_{D,w,\gamma}(S'\times S')\right|>\frac{\epsilon}{2}\middle|\bar{S}\right)\end{aligned}$$

Now, instead of bounding the supremum over $\Delta\mathcal{H}$, we pay approximation error of ϵ' in order to bound the supremum over $\mathcal{H}$.

Lemma 21. *For some fixed data sample S of size n, any $D\in\Delta\mathcal{H}$ can be approximated by some uniform mixture over $k := \frac{2\ln(2n^2)}{\epsilon'^2}+1$ hypotheses $\hat{D}=\frac{1}{k}\{h_1,\ldots,h_k\}$ such that for every $(x,x')\in S\times S$,*

$$\left|\mathop{\mathbb{E}}_{h\sim D}\left[h(x)-h(x')\right]-\mathop{\mathbb{E}}_{h\sim\hat{D}}\left[h(x)-h(x')\right]\right|\le\epsilon'.$$

Proof. Fix some $(x,x')\in S\times S$. Randomly sample k hypotheses from D: $\{h_i\}_{i=1}^k\sim D^k$. Because for each randomly drawn hypothesis $h_i\sim D$, the difference in its prediction for x and x' is exactly $\mathbb{E}_{h\sim D}[h(x)-h(x')]$, Hoeffding's inequality yields that

$$\begin{aligned}&\Pr_{h_i\sim D,i\in[k]}\left(\left|\mathop{\mathbb{E}}_{h\sim D}\left[h(x)-h(x')\right]-\frac{1}{k}\sum_{i=1}^{k}\left[h_i(x)-h_i(x')\right]\right|>\epsilon'\right)\\&\le 2\exp\left(-\frac{2k^2\epsilon'^2}{4k}\right)\\&=2\exp\left(-\frac{k\epsilon'^2}{2}\right).\end{aligned}$$

However, there are n^2 fixed pairs in $S\times S$, and if we distribute the failure property between n^2 pairs and union bound over all of them, we get

$$\Pr_{h_i\sim D,i\in[k]}\left(\max_{(x,x')\in S\times S}\left|\mathop{\mathbb{E}}_{h\sim D}\left[h(x)-h(x')\right]-\frac{1}{k}\sum_{i=1}^{k}[h_i(x)-h_i(x')]\right|>\epsilon'\right)\le 2n^2\exp\left(-\frac{k\epsilon'^2}{2}\right).$$

In order to achieve non-zero probability of having

$$\left| \mathop{\mathbb{E}}_{h \sim D} \left[h(x) - h(x') \right] - \frac{1}{k} \sum_{i=1}^{k} [h_i(x) - h_i(x')] \right| \leq \epsilon', \forall (x, x') \in S \times S,$$

we need to make sure $2n^2 \exp\left(-\frac{k\epsilon'^2}{2}\right) < 1$ or $k > \frac{2\ln\left(2n^2\right)}{\epsilon'^2}$.

□

Corollary 2. *For some fixed data sample S, any $D \in \Delta\mathcal{H}$ can be approximated by a uniform mixture of $k := \frac{2\ln(2n^2)}{\epsilon'^2} + 1$ hypotheses $\hat{D} = \frac{1}{k}\{h_1, \ldots, h_k\}$ such that*

$$\left| \Pi_{D,w,\gamma}(S \times S) - \Pi_{\hat{D},w,\gamma}(S \times S) \right| \leq \epsilon'$$

Proof. It simply follows from Lemma 21 and the fact that $\max\left(0, \mathbb{E}_{h\sim D}\left[h(x_i) - h(x_j)\right] - \gamma\right)$ is 1-Lipschitz in terms of $\mathbb{E}_{h\sim D}[h(x_i) - h(x_j)]$. □

Using Corollary 2 and using Sauer's lemma that bounds the total number of possible labelings by $\mathcal{H}$ over $2n$ points to be $\left(\frac{e \cdot 2n}{d}\right)^d$, we can show

$$\begin{aligned}
&\sum_{\bar{S}} \Pr\left(\bar{S}\right) \Pr_{S,S'} \left(\sup_{D \in \Delta\mathcal{H}} \left| \Pi_{D,w,\gamma}(S \times S) - \Pi_{D,w,\gamma}(S' \times S') \right| > \frac{\epsilon}{2} \;\middle|\; \bar{S} \right) \\
&\leq \sum_{\bar{S}} \Pr\left(\bar{S}\right) \Pr_{S,S'} \left(\sup_{\hat{D} \in \mathcal{H}^k} \left| \Pi_{\hat{D},w,\gamma}(S \times S) - \Pi_{\hat{D},w,\gamma}(S' \times S') \right| > \frac{\epsilon}{2} + \epsilon' \;\middle|\; \bar{S} \right) \\
&\leq \sum_{\bar{S}} \Pr\left(\bar{S}\right) \cdot \left(\frac{e \cdot 2n}{d}\right)^{dk} \sup_{\hat{D} \in \mathcal{H}^k} \Pr_{S,S'} \left(\left| \Pi_{\hat{D},w,\gamma}(S \times S) - \Pi_{\hat{D},w,\gamma}(S' \times S') \right| > \frac{\epsilon}{2} + \epsilon' \;\middle|\; \bar{S} \right)
\end{aligned}$$

Now, for any $\hat{D}$, we will try to bound the probability that the difference in fairness loss between S and S' is big. We do so by union bounding over cases where both of them deviate from its mean by too much.

If we have

$$\left|\Pi_{\hat{D},w,\gamma}(S \times S) - E_{S|\bar{S}}\left[\Pi_{\hat{D},w,\gamma}(S \times S)\right]\right| \leq \frac{\epsilon}{4} + \frac{\epsilon'}{2}$$

$$\left|\Pi_{\hat{D},w,\gamma}(S' \times S') - E_{S|\bar{S}}\left[\Pi_{\hat{D},w,\gamma}(S \times S)\right]\right| \leq \frac{\epsilon}{4} + \frac{\epsilon'}{2},$$

then $\left|\Pi_{\hat{D},w,\gamma}(S \times S) - \Pi_{\hat{D},w,\gamma}(S' \times S')\right| \leq \frac{\epsilon}{2} + \epsilon'$. In other words,

$$\begin{aligned}
&\Pr_{S,S'}\left(\left|\Pi_{\hat{D},w,\gamma}(S \times S) - \Pi_{\hat{D},w,\gamma}(S' \times S')\right| \leq \frac{\epsilon}{2} + \epsilon' \;\middle|\; \bar{S}\right) \\
&\geq \Pr_{S,S'}\Bigg(\left|\Pi_{\hat{D},w,\gamma}(S \times S) - E_{S|\bar{S}}\left[\Pi_{\hat{D},w,\gamma}(S \times S)\right]\right| \leq \frac{\epsilon}{4} + \frac{\epsilon'}{2} \text{ and} \\
&\left|\Pi_{\hat{D},w,\gamma}(S' \times S') - E_{S|\bar{S}}\left[\Pi_{\hat{D},w,\gamma}(S \times S)\right]\right| \leq \frac{\epsilon}{4} + \frac{\epsilon'}{2}\Bigg|\bar{S}\Bigg).
\end{aligned}$$

Therefore, by looking at the compliment probabilities, we have

$$\begin{aligned}
&\Pr_{S,S'}\left(\left|\Pi_{\hat{D},w,\gamma}(S \times S) - \Pi_{\hat{D},w,\gamma}(S' \times S')\right| > \frac{\epsilon}{2} + \epsilon' \;\middle|\; \bar{S}\right) \\
&\leq \Pr_{S,S'}\Bigg(\left|\Pi_{\hat{D},w,\gamma}(S \times S) - E_{S|\bar{S}}\left[\Pi_{\hat{D},w,\gamma}(S \times S)\right]\right| > \frac{\epsilon}{4} + \frac{\epsilon'}{2} \\
&\text{or } \left|\Pi_{\hat{D},w,\gamma}(S' \times S') - E_{S|\bar{S}}\left[\Pi_{\hat{D},w,\gamma}(S \times S)\right]\right| > \frac{\epsilon}{4} + \frac{\epsilon'}{2} \;\Big|\; \bar{S}\Bigg) \\
&\leq 2\Pr_{S}\left(\left|\Pi_{\hat{D},w,\gamma}(S \times S) - E_{S|\bar{S}}\left[\Pi_{\hat{D},w,\gamma}(S \times S)\right]\right| > \frac{\epsilon}{4} + \frac{\epsilon'}{2} \;\middle|\; \bar{S}\right).
\end{aligned}$$

Here, we can't appeal to McDiarmid's because S is sampled without replacement from $\bar{S}$. However, we can use the same technique that [74] leveraged – stochastic covering property can be used to show concentration for sampling without replacement [77].

Definition 14 ([77]). *$Z_1, \ldots, Z_n$ satisfy the stochastic covering property, if for any $I \subset [n]$ and $a \geq a' \in \{0,1\}^I$ coordinate-wise such that $||a' - a||_1 = 1$, there is a coupling ν of the distributions μ, μ' of $(Z_j : j \in [n] \setminus I)$ conditioned on $Z_I = a$ or $Z_I = a'$, respectively, such that $\nu(x, y) = 0$ unless $x \leq y$ coordinate-wise and $||x - y||_1 \leq 1$.*

Theorem 18 ([77])**.** *Let $(Z_1, \ldots, Z_n) \in \{0, 1\}$ be random variables such that $\Pr(\sum_{i=1}^{n} Z_i = k) = 1$ and the stochastic covering property is satisfied. Let $f : \{0, 1\}^n \to \mathbb{R}$ be an c-Lipschitz function. Then, for any $\epsilon > 0$,*

$$\Pr\left(|f(Z_1, \ldots, Z_n) - \mathbb{E}\left[f(Z_1, \ldots, Z_n)\right]| \geq \epsilon\right) \leq 2\exp\left(\frac{-\epsilon^2}{8c^2k}\right)$$

Lemma 22 ([74])**.** *Given a set S of n points, sample $k \leq n$ elements without replacement. Let $Z_i = \{0, 1\}$ indicate whether ith element has been chosen. Then, $(Z_1, \ldots, Z_n)$ satisfy the stochastic covering property.*

Let $\bar{S} = \{x_1, \ldots, x_{2n}\}$. If we slightly change the definition of the fairness loss so that it depends on the indicator variables $Z_1, \ldots, Z_{2n}$,

$$\Pi''_{\hat{D},w,\gamma,\bar{S}}(Z_1, \ldots, Z_{2n}) = \frac{1}{n^2} \sum_{i,j \in [2n]^2} Z_i Z_j \Pi_{\hat{D},w,\gamma}(x_i, x_j) = \Pi_{\hat{D},w,\gamma}(S \times S).$$

We see that $\Pi''_{\hat{D},w,\gamma,\bar{S}}$ is $\frac{1}{n}$-Lipschitz, so by theorem 18 and lemma 22, we get

$$\begin{aligned}
&\Pr_S\left(\left|\Pi_{\hat{D},w,\gamma}(S \times S) - E_{S|\bar{S}}[\Pi_{\hat{D},w,\gamma}(S \times S)]\right| > \frac{\epsilon}{4} + \frac{\epsilon'}{2} \;\middle|\; \bar{S}\right) \\
&\leq 2\exp\left(\frac{-\left(\frac{\epsilon}{4} + \frac{\epsilon'}{2}\right)^2}{8\frac{1}{n^2} \cdot n}\right) = 2\exp\left(\frac{-n\left(\frac{\epsilon}{4} + \frac{\epsilon'}{2}\right)^2}{8}\right)
\end{aligned}$$

Combining everything, we get

$$
\begin{aligned}
&\Pr_{S}\left(\sup_{D\in\Delta\mathcal{H}}\left|\Pi_{D,w,\gamma}(S\times S)-\mathop{\mathbb{E}}_{x,x'}[\Pi_{D,w,\gamma}(x,x')]\right|>\epsilon\right)\\
&\le 2\sum_{\bar{S}}\Pr\left(\bar{S}\right)\cdot\left(\frac{e\cdot 2n}{d}\right)^{dk}\sup_{\hat{D}\in\mathcal{H}^k}\Pr_{S,S'}\left(\left|\Pi_{\hat{D},w,\gamma}(S\times S)-\Pi_{\hat{D},w,\gamma}(S'\times S')\right|>\frac{\epsilon}{2}+\epsilon'\;\middle|\;\bar{S}\right)\\
&\le 4\sum_{\bar{S}}\Pr\left(\bar{S}\right)\cdot\left(\frac{e\cdot 2n}{d}\right)^{dk}\sup_{\hat{D}\in\mathcal{H}^k}\Pr_{S}\left(\left|\Pi_{\hat{D},w,\gamma}(S\times S)-E_{S|\bar{S}}\left[\Pi_{\hat{D},w,\gamma}(S\times S)\right]\right|>\frac{\epsilon}{4}+\frac{\epsilon'}{2}\;\middle|\;\bar{S}\right)\\
&\le 8\cdot\left(\frac{e\cdot 2n}{d}\right)^{dk}\exp\left(\frac{-n\left(\frac{\epsilon}{4}+\frac{\epsilon'}{2}\right)^2}{8}\right)
\end{aligned}
$$

For convenience, we set $\epsilon'=\frac{\epsilon}{2}$. □

However, in our case, instead of finding the average over all pairs in S, we calculate the fairness loss only over m pairs. Fixing S, if m is sufficiently large, our empirical fairness loss should concentrate around the fairness loss over all the pairs for S.

Lemma 23. *For fixed S, randomly chosen pairs $M\subset S\times S$, and randomized hypothesis D,*

$$\Pr_{M\sim(S\times S)^m}\left(\Pi_{D,w,\gamma}(M)-\Pi_{D,w,\gamma}(S\times S)\ge\epsilon\right)\le\exp\left(-2m\epsilon^2\right)$$

Proof. Write a random variable $L_a=\Pi_{D,w,\gamma}((x_{2a-1},x_{2a}))$ for the fairness loss of the ath pair. Note that

$$E[L_a]=\sum_{(i,j)\in[n]^2}\frac{1}{n^2}\Pi_{D,w,\gamma}\left((x_i,x_j)\right)=\Pi_{D,w,\gamma}(S\times S),\forall a\in[|M|].$$

Therefore, by Hoeffding's inequality, we have

$$\Pr_{M}\left(\Pi_{D,w,\gamma}(M)-\Pi_{D,w,\gamma}(S\times S)\ge\epsilon\right)\le\exp\left(-2m\epsilon^2\right).$$

□

Lemma 24. *For fixed S and randomly chosen pairs $M \subset S \times S$,*

$$\Pr_{M \sim (S\times S)^m} \left(\sup_{D \in \Delta\mathcal{H}} |\Pi_{D,w,\gamma}(M) - \Pi_{D,w,\gamma}(S \times S)| \geq \epsilon \right) \leq \left(\frac{e \cdot 2n}{d} \right)^{dk'} \exp\left(-8m\epsilon^2\right),$$

where $k' = \frac{2\ln(2m)}{\epsilon^2} + 1$.

Proof.

$$\begin{aligned}
&\Pr_{M \sim (S\times S)^m} \left(\sup_{D \in \Delta\mathcal{H}} |\Pi_{D,w,\gamma}(M) - \Pi_{D,w,\gamma}(S \times S)| \geq \epsilon \right) \\
&\leq \Pr_{M \sim (S\times S)^m} \left(\sup_{\hat{D} \in \mathcal{H}^k} \left|\Pi_{\hat{D},w,\gamma}(M) - \Pi_{\hat{D},w,\gamma}(S \times S)\right| \geq \epsilon + 2\epsilon' \right) \\
&\leq \sum_{\hat{D} \in \mathcal{H}^k} \Pr_{M \sim (S\times S)^m} \left(\left|\Pi_{\hat{D},w,\gamma}(M) - \Pi_{\hat{D},w,\gamma}(S \times S)\right| \geq \epsilon + 2\epsilon' \right) \\
&\leq \left(\frac{e \cdot 2n}{d} \right)^{dk} \exp\left(-2m\left(\epsilon + 2\epsilon'\right)^2\right),
\end{aligned}$$

where $k = \frac{2\ln(2m)}{4\epsilon'^2} + 1$. The last inequality is from Corollary 2 and Lemma 23. For convenience, we just set $\epsilon' = \epsilon/2$. □

4.B.3. Omitted Proof of Theorem 13

Combining theorem 17 and lemma 24 yields the following theorem for fairness loss generalization.

Theorem 13. *Let S consists of n i.i.d points drawn from $\mathcal{P}$ and let M represent a set of m pairs randomly drawn from $S \times S$. Then we have:*

$$\begin{aligned}
&\Pr_{\substack{S \sim \mathcal{P}^n \\ M \sim (S\times S)^m}} \left(\sup_{D \in \Delta\mathcal{H}} \left| \Pi_{D,w,\gamma}(M) - \mathop{\mathbb{E}}_{(x,x') \sim \mathcal{P}^2} \left[\Pi_{D,w,\gamma}(x,x')\right] \right| > 2\epsilon \right) \\
&\leq \left(8 \cdot \left(\frac{e \cdot 2n}{d} \right)^{dk} \exp\left(\frac{-n\epsilon^2}{32} \right) + \left(\frac{e \cdot 2n}{d} \right)^{dk'} \exp\left(-8m\epsilon^2\right) \right),
\end{aligned}$$

where $k' = \frac{2\ln(2m)}{\epsilon^2} + 1$, $k = \frac{\ln(2n^2)}{8\epsilon^2} + 1$, and d is the VC-dimension of $\mathcal{H}$.

Proof. With probability $1 - \left(8 \cdot (\frac{e \cdot 2n}{d})^{dk} \exp\left(\frac{-n\epsilon^2}{32}\right) + \left(\frac{e \cdot 2n}{d}\right)^{dk'} \exp\left(-8m\epsilon^2\right)\right)$, where $k' = \frac{2\ln(2m)}{\epsilon^2} + 1$ and $k = \frac{\ln(2n^2)}{8\epsilon^2} + 1$, we have

$$\sup_{D \in \Delta\mathcal{H}} |\Pi_{D,w,\gamma}(M) - \Pi_{D,w,\gamma}(S \times S)| \leq \epsilon$$

and

$$\sup_{D \in \Delta\mathcal{H}} \left|\Pi_{D,w,\gamma}(S \times S) - \mathop{\mathbb{E}}_{x,x'}[\Pi_{D,w,\gamma}(x,x')\right| \leq \epsilon.$$

Then, by triangle inequality,

$$\sup_{D \in \Delta\mathcal{H}} \left|\Pi_{D,w,\gamma}(M) - \mathop{\mathbb{E}}_{x,x'}[\Pi_{D,w,\gamma}(x,x')\right| \leq 2\epsilon.$$

In other words, with probability $\left(8 \cdot (\frac{e \cdot 2n}{d})^{dk} \exp\left(\frac{-n\epsilon^2}{32}\right) + \left(\frac{e \cdot 2n}{d}\right)^{dk'} \exp\left(-8m\epsilon^2\right)\right)$, we have

$$\sup_{D \in \Delta\mathcal{H}} \left|\Pi_{D,w,\gamma}(M) - \mathop{\mathbb{E}}_{x,x'}\left[\Pi_{D,w,\gamma}(x,x')\right]\right| > 2\epsilon.$$

□

II

Uncertainty Estimation for Subgroups

“Wisdom is knowing what you don’t know.”

Socrates

Chapter 5

Uncertainty Estimation for Subgroups: Offline

5.1. Introduction

Uncertainty estimation is fundamental to prediction and regression. Given a training set of labelled points $D \subseteq \mathcal{X} \times [0,1]$ consisting of feature vectors $x \in \mathcal{X}$ and labels $y \in [0,1]$, the standard regression problem is to find a function $\overline{\mu} : \mathcal{X} \to [0,1]$ that delivers a good point estimate of $\mu(x) = \mathbb{E}[y|x]$. We also desire the *variance* of the label distribution $\mathbb{E}[(y - \mu(x))^2|x]$ as a measure of the inherent uncertainty of a prediction. Higher central moments would yield even more information about this uncertainty which can be represented by *prediction intervals*: An interval $[\ell(x), u(x)]$ that with high probability contains y, i.e., $\Pr_y[y \in [\ell(x), u(x)]|x] \geq 1 - \delta$ for some $\delta \in (0,1)$.

If the data are generated according to a parametric model as in the classic ordinary least squares setting, one can form confidence regions around the underlying model parameters and translate these into both mean and uncertainty estimates about individual predictions. In non-parametric settings, it is unclear how one should reason about uncertainty. We typically observe each feature vector x infrequently, so we have essentially no information about the true distribution on y conditional on x. One solution to this problem is to compute *marginal* prediction intervals which average over *data points* x to give guarantees of the form: $\Pr_{x,y}[y \in [\ell(x), u(x)]] \geq 1 - \delta$. This is the approach that is taken in the *conformal prediction* literature — see e.g. Shafer and Vovk [86].

Marginal prediction intervals, unlike prediction intervals, do *not* condition on x. They offer a promise not over the randomness of the label conditional on the features, but over an average over data points. To make the distinction vivid, imagine one is a patient with high blood pressure, and a statistical model asserts that a certain drug will lower one's diastolic blood pressure to between 70 and 80 mm Hg. If $[70, 80]$ were a 95% prediction interval *conditional on all of one's observable features*, then one could reason that over the unrealized randomness in the world, there is a 95% chance that one's new blood pressure will lie in $[70, 80]$. If $[70, 80]$ is a 95% marginal prediction interval, however, it means that *95% of all patients who take the drug* will see their blood pressure decline to a level contained within the interval. Because the average is taken over a large, heterogeneous collection of people, the guarantee of the marginal prediction interval offers no meaningful promise to individuals. For example, it is possible that patients that share one's demographic characteristics (e.g. women of Sephardic Jewish descent with a family history of diabetes) will tend to see their blood pressure elevated by the drug.

This fundamental problem with uncertainty estimation in non-parametric settings is also a problem for mean estimation: what does it mean that a point prediction $\overline{\mu}(x)$ is an estimate of $\mathbb{E}[y|x]$ if we have no knowledge of the distribution on y conditional on x (because we have observed no samples from this distribution)? A standard performance measure is *calibration* [17], which similarly averages over data points: a predictor $\overline{\mu}$ is calibrated (roughly) if $\mathbb{E}_{(x,y)}[\overline{\mu}(x) - y|\overline{\mu}(x) = i] = 0$ for all predictions i: i.e. for every i, conditioned on x being such that the prediction $\overline{\mu}(x)$ was (close to) i, the expected outcome y is also (close to) i. Just as with marginal prediction intervals, guarantees of calibration mean little to individuals, who differ substantially from the majority of people over whom the average is taken.

Hébert-Johnson et al. [43] proposed *multicalibration* as a way to interpolate between the (unattainable) ideal of being able to correctly predict $\mathbb{E}[y|x]$ for each x and offering a guaranteed averaged over the entire data distribution. The idea is to fix a large, structured set of (possibly overlapping) sub-populations ($\mathcal{G} \in 2^X$). A predictor $\overline{\mu}$ is multicalibrated if, informally, for all predictions i and groups $G \in \mathcal{G}$, $\mathbb{E}_{(x,y)}[\overline{\mu}(x) - y|\overline{\mu}(x) = i, x \in G] = 0$. Thus, $\overline{\mu}$ is calibrated not just on the overall population, but also simultaneously on many different finely defined sub-populations that one might care about (e.g. different demographic groups). Hébert-Johnson et al. [43] show how to compute an approximately multicalibrated predictor $\overline{\mu}$ on all subgroups in $\mathcal{G}$ that have substantial probability mass—we provide a high level description of their algorithm, which we use, below.

The main contribution of this chapter is to show how to achieve what can loosely be termed multicalibration for higher moment estimates. We provide not just estimates $\overline{\mu}(x)$ of means ($\mu(x) = \mathbb{E}[y|x]$), but also estimates, $\overline{m}_k(x)$, for higher central moments, ($m_k(x) = \mathbb{E}[(y - \mu(x))^k|x]$) such that all of these forecasts are appropriately multicalibrated in a sense made precise below. This is useful for a number of basic tasks. One we briefly highlight is that it can help diagnose data iniquities: for example, if the set of collected features is much less predictive of the target label on certain demographic groups $G \in \mathcal{G}$ this will necessarily manifest itself in multicalibrated moment predictions by having higher variance predictions on individual members of those populations.

As an important application, we show that standard concentration inequalities which could be applied using the true moments of a distribution to obtain prediction intervals can also be applied using our multicalibrated moment estimates. Doing so produces intervals $[\ell(x), u(x)]$ for each data point that are *simultaneously* valid marginal prediction intervals not just overall, but also conditioned on x lying in any of the (sufficiently large) subgroups over which we are multicalibrated. This allows one to interpret these prediction intervals as predicting something meaningful not just an average over all people, but — simultaneously — as averages over all of the people who were given the same prediction, across many finely defined subgroups (like women of Sephardic Jewish descent with a family history of diabetes). Note that because the groups $G \in \mathcal{G}$ may overlap, a single individual can belong to many such groups and can at her option interpret the prediction interval as averaging over any of them.

5.1.1. Overview of Our Approach and Results

Mean Multicalibration and Impediments to Extensions to Higher Moments

We first review the algorithm of Hébert-Johnson et al. [43], recast in the framework in which we will conduct our analysis. We here elide some issues such as how we deal with discretization and how calibration error is parameterized — see Section 5.3 for the formal model and definitions. Fix a feature space $\mathcal{X}$, labels $\mathcal{Y} = [0,1]$, and an unknown distribution $\mathcal{P}$ over $\mathcal{X} \times \mathcal{Y}$. Given are sets $\mathcal{G} \subseteq 2^{\mathcal{X}}$, corresponding to sub-populations of interest. The goal is to construct a predictor, $\overline{\mu} : \mathcal{X} \to \mathcal{Y}$, that is multicalibrated, i.e. calibrated on each group $G \in \mathcal{G}$. This means that we want a predictor, $\overline{\mu}$, that is *mean-consistent* on every set of the form $G(\overline{\mu}, i) = \{x \in G : \overline{\mu}(x) = i\}$ for some i: in other words, for every such set $G(\overline{\mu}, i)$ we want $\mathbb{E}_{(x,y)\sim\mathcal{P}}[\overline{\mu}(x) - y]|x \in G(\overline{\mu}, i)] = 0$. We describe the algorithm as if it has direct access to the true distribution $\mathcal{P}$ and defer for now a description of how to implement the algorithm using a finite sample.

It is helpful to conceive of the task as a zero-sum game between two players: a "(mean) consistency player", and an "audit player" who knows the true distribution $\mathcal{P}$. The consistency player chooses a predictor $\overline{\mu}$, and the audit player, given a predictor, attempts to identify a subset S of $\mathcal{X}$ on which the predictor is not mean consistent.[12] Given a pair of choices, the corresponding cost (which the consistency player wishes to minimize and the audit player wishes to maximize) is the absolute value difference between the average prediction of the consistency player and the average expected label on the subset S identified by the audit player. The value of this game is 0, since the consistency player can obtain perfect consistency using the true conditional label distribution $\overline{\mu}(x) = \mathbb{E}[y|x]$. The algorithm of Hébert-Johnson et al. [43] can be interpreted as solving this zero sum game by simulating repeated play, using online gradient descent for the consistency player, and "best response" for an audit player, who stops play if there are no remaining sets $S = G(\overline{\mu}, i)$ witnessing violations of multicalibration. This works because by linearity of expectation, we can formulate the game so that the consistency player's utility function is *linear* in her individual predictions $\overline{\mu}(x)$. A formal description and proof of correctness can be found in Section 5.4.1.

There are two—related—impediments to extending this approach to higher moments, i.e., finding predictors $\overline{m}_k(x) \approx m_k(x) = \mathbb{E}[(y - \mathbb{E}[y|x])^k|x]$, that are "consistent" with $\mathcal{P}$ on many

[12] Here, and in what follows, we adopt the convention that G refers to a group in $\mathcal{G}$, while S refers to any generic subset of $\mathcal{X}$.

sets. The first of these is definitional—what do we mean by "consistent" for higher moments? The second is algorithmic—given a definition, how do we achieve it? Both are impediments because, unlike means, higher moments are not linear functionals of the distribution. A consequence is that moments for $k > 1$ do not combine linearly in the way expectations do. In particular for $S = S_1 \cup S_2$ where S_1 and S_2 are disjoint,

$$\begin{aligned}&\mathbb{E}[(y - \mathbb{E}[y|x \in S])^k|S] \\ &\neq \Pr[x \in S_1|S]\,\mathbb{E}[(y - \mathbb{E}[y|x \in S_1])^k|S_1] + \Pr[x \in S_2|S]\,\mathbb{E}[(y - \mathbb{E}[y|x \in S_2])^k|S_2].\end{aligned}$$

It is therefore silly to require that moment predictions $\overline{m}_k(x)$ satisfy the same "average consistency" condition asked of means: i.e. we cannot demand that the population variance on the subset of the population on which we predict variance v be v, because this is not a property that the true moments $m_k(x)$ satisfy. Consider, for example, a setting in which there are two types of points, x_1 and x_2. The true distribution is uniform over $\{(x_1, 0), (x_2, 1)\}$ (and so in particular the label y is deterministically fixed by the features). We have that for all $k > 1$, $\mu(x_1) = 0, \mu(x_2) = 1$, and $m_k(x_0) = m_k(x_1) = 0$. Nevertheless, the variance over the set of points on which the true distribution satisfies $m_k(x) = 0$ is $1/4$, not 0. We cannot ask that our "moment calibrated" predictors satisfy properties violated by the true distribution because we would have no guarantee of feasibility — and our ultimate goal in multicalibration is to find a set of mean and moment predictors that are indistinguishable from the true distribution with respect to some class of tests.

Mean Conditioned Moment Multicalibration and Marginal Prediction Intervals

A key observation (Observation 1) is that higher moments *do* linearize over sets that have the same mean: in other words, if we have $S = S_1 \cup S_2$ for disjoint S_1 and S_2 such that $\mathbb{E}[y|x \in S_1] = \mathbb{E}[y|x \in S_2]$, then, it follows that

$$\begin{aligned}&\mathbb{E}[(y - \mathbb{E}[y|x \in S])^k|S] \\ &= \Pr[x \in S_1|S]\,\mathbb{E}[(y - \mathbb{E}[y|x \in S_1])^k|S_1] + \Pr[x \in S_2|S]\,\mathbb{E}[(y - \mathbb{E}[y|x \in S_2])^k|S_2].\end{aligned}$$

An implication of this is that the true distribution does satisfy what we term *mean-conditioned moment multicalibration.* Namely, if for a fixed $k > 1$ we define for each set $G \in \mathcal{G}$ and each pair of mean and k^{th} moment values i, j the sets: $G(\mu, m_k, i, j) = \{x \in G : \mu(x) = i, m_k(x) = j\}$, then we have *both* mean consistency: $\mathbb{E}[(y - i)|x \in G(\mu, m_k, i, j)] = 0$ *and* moment consistency: $\mathbb{E}[(y - i)^k - j|x \in G(\mu, m_k, i, j)] = 0$ over these sets. Therefore, we require the same condition to hold for our mean and moment predictors $\overline{\mu}$ and $\{\overline{m}_a\}_{a=1}^k$: namely that simultaneously for every a, that over each of the sets $G(\overline{\mu}, \overline{m}_a, i, j)$, the true label mean should be i and the true label a-th moment should be j. In other words, if we have a set of predictors that are mean conditioned moment multicalibrated, then an individual who receives a particular mean and (e.g.) variance prediction can be assured that amongst all the people who received the same mean and variance prediction *even averaged over any of the possibly large number of sub-groups G of which the individual is a member*, the true mean and variance are faithful to the prediction.

Section 5.6 demonstrates a key application of mean-conditioned moment-multicalibrated estimators: They can be used in place of real distributional moments to derive prediction intervals. Given moments of a random variable X, a standard way to derive concentration inequalities for X is by using the following inequality for any even moment (for $k = 2$ this is Cheybychev's inequality):

$$\Pr[|X - \mu(X)| \geq t] \leq \frac{\mathbb{E}\left[(X - \mu(X))^k\right]}{t^k}.$$

If X is the label distribution conditional on features x, this yields the prediction interval:

$$\Pr_y\left[y \in \left[\mu(x) - \left(\frac{m_k(x)}{\delta}\right)^{1/k}, \mu(x) + \left(\frac{m_k(x)}{\delta}\right)^{1/k}\right] \middle| x\right] \geq 1 - \delta.$$

In Section 5.6, we show that if we have a mean-conditioned moment-multicalibrated pair $(\overline{\mu}, \overline{m}_k)$, we can replace the true mean and moments in the derivation of this prediction interval, and get *marginal* prediction intervals, which are valid not just averaged over all points, but simultaneously as averaged over all point that received the same prediction within any of the groups within which we are mean-conditioned moment multicalibrated. In other words, for all $G \in \mathcal{G}$ and for all i, j:

$$\Pr_{(x,y)\sim P}\left[y \in \left[\overline{\mu}(x) - \left(\frac{\overline{m}_k(x)}{\delta}\right)^{1/k}, \overline{\mu}(x) + \left(\frac{\overline{m}_k(x)}{\delta}\right)^{1/k}\right] \middle| x \in G(\overline{\mu}, \overline{m}_k, i, j)\right] \geq 1 - \delta.$$

Achieving Mean Conditioned Moment Multicalibration

What is the difficulty with finding sets of predictors $(\overline{\mu}, \{\overline{m}_a\}_{a=2}^k)$ such that simultaneously each pair $(\overline{\mu}, \overline{m}_a)$ are mean-conditioned moment multicalibrated? It is that moments do not have the linear structure that means do. Hence, the zero-sum game formulation we describe for mean-multicalibration cannot be applied directly. A naïve approach (which fails but will be a useful sub-routine for us) is to first train a mean-multicalibrated predictor $\overline{\mu}$, and then define "pseudo-moment" labels for each x as $\widetilde{m}_{k,\overline{\mu}}(x) = (y - \overline{\mu}(x))^k$. Since these are constant values, we can then use the algorithm for mean multicalibration to achieve "pseudo-moment calibration with respect to $\overline{\mu}$" — i.e. mean consistency on each set $G(\overline{\mu}, \overline{m}_k, i, j)$ with respect to our pseudo-moment labels $\widetilde{m}_{k,\overline{\mu}}(x)$. By itself this doesn't guarantee any sort of "moment consistency," but we show in Section 5.4.2 that if we can:

1. Find moment predictors $\overline{m}_k$ that satisfy pseudo-moment calibration with respect to $\overline{\mu}$, *and*

2. Our mean predictor $\overline{\mu}$ satisfies mean consistency on every set of the form $G(\overline{\mu}, \overline{m}_k, i, j)$,

then the pair $(\overline{\mu}, \overline{m}_k)$ will satisfy mean-conditioned moment calibration.

The difficulty is that these two requirements are circularly defined. Once we have a *fixed* mean predictor $\overline{\mu}$, we can use a gradient descent procedure to find moment predictors $\{\overline{m}_a\}_{a=2}^k$ that are pseudo-calibrated with respect to $\overline{\mu}$. However, we also require our mean predictor to be mean consistent on the sets $G(\overline{\mu}, \overline{m}_a, i, j)$, which are undefined until we fix our moment predictors $\{\overline{m}_a\}_{a=2}^k$. Section 5.4.3 resolves the circularity by using an alternating descent procedure that toggles between updating $\overline{\mu}$ and $\{\overline{m}_a\}_{a=2}^k$, each aiming for a mean calibration target that is defined with respect to the other. We prove that this alternating gradient descent procedure is guaranteed to converge after only a small number of rounds.

Finally, we show in Section 5.5 how to implement our algorithm using a finite sample from the distribution and furnish sample complexity bounds, in a way analogous to Hébert-Johnson et al. [43]. The sample complexity bounds are logarithmic in the number of groups $|\mathcal{G}|$ that we wish to be multicalibrated with respect to, and polynomial in our desired calibration error parameters and the number of moments k with which we wish to achieve mean-conditioned moment multicalibrated predictors. In particular, because dependence on $|G|$ is only logarithmic, we can satisfy mean-conditioned moment-multicalibration on an exponentially large collection of intersecting sets $\mathcal{G}$ from just a polynomial sample of data from the unknown population distribution. Note, however, that despite our polynomial dependence on k, the natural scale of the k'th moment decreases exponentially in k, and so to obtain non-trivial approximation guarantees for k'th moments with polynomial sample complexity, we should think of taking k at most logarithmic in the relevant parameters of the problem. See Theorem 22 and Corollary 4 for details. Our running time scales polynomially with our approximation error parameters, the number of moments k with which we wish to be multicalibrated, and the running time of solving learning problems over $\mathcal{G}$ (which is at most linear in $|\mathcal{G}|$, but can be much faster). See Theorems 22 and 25 for details. In other words, our algorithms are "oracle efficient" in the sense that if we have a subroutine for solving learning problems over $\mathcal{G}$, then we can use it to solve mean-conditioned moment-multicalibration problems with at most polynomial overhead. In theory, for almost every interesting class $\mathcal{G}$, learning over $\mathcal{G}$ is hard in the worst case — but oracle efficiency has proven to be a useful paradigm in the design of learning algorithms (especially in the fairness in machine learning literature — see e.g. [2, 43, 59, 61]) because in practice we have extremely powerful heuristics for solving complex learning problems. Moreover, this kind of oracle efficiency is the best running time guarantee that we can hope for, because as shown by Hébert-Johnson et al. [43], even mean-multicalibration is as hard as solving arbitrary learning problems over $\mathcal{G}$ in the worst case.

5.2. Related Work

Calibration as a means of evaluating forecasts of expectations dates back to Dawid [17]. This literature focuses on a simple online forecasting setting, motivated by weather prediction problems: in a sequence of rounds, nature chooses the probability of some binary event (e.g. rain), and a forecaster predicts a probability of that event. Dawid [17] shows that a Bayesian forecaster will always be subjectively calibrated (i.e. he will believe himself to be calibrated). Foster and Vohra [29] show that there exist *randomized* forecasters that can asymptotically satisfy calibration against arbitrary sequences of outcomes (this is impossible for deterministic forecasters [76]). These papers focus on the online setting, because simple calibration is trivial in a batch/distributional setting: simply predicting the mean outcome on every point satisfies calibration. Within this literature, the most related works are Lehrer [68] and Sandroni et al. [82], which give very general asymptotic results that are able to achieve (mean) multicalibration as a special case. Lehrer [68], operating in the sequential online setting, asks for calibration to hold not just on the entire sequence of realized outcomes, but on countably many infinite *sub-sequences* (e.g. the set of all computable subsequences). He proves that there exists an online forecasting algorithm which can asymptotically achieve this. Sandroni et al. [82] extend this result to subsequences which can be defined in terms of the forecasters predictions as well. Both of these papers operate in a setting that is general enough to encode the constraint of mean multicalibration (by encoding the features of datapoints in the "state space") even in an online, adversarial setting — albeit not in a computationally or sample efficient way. In contrast, Hébert-Johnson et al. [43], who define the notion of mean multicalibration, give an algorithm for achieving it in a batch distributional setting — in a much more computationally and sample efficient manner than could have been achieved by applying the machinery of Lehrer [68] and Sandroni et al. [82]. Recently, Zhao et al. [101] gave a notion of "individual level" (mean) calibration, defined over the randomness of the forecaster, that is valid conditional on individual data points (i.e. without needing to average over a population). They provide promising empirical results, but the theoretical guarantees of predictors satisfying this notion do not provide non-trivial information about a data distribution because (as the authors note) their notion of individual calibration can be satisfied without observing any data.

Hébert-Johnson et al. [43] also proposed the notion of "multi-accuracy," a weaker notion than multicalibration which asks for a predictor $\overline{\mu}$ that satisfies mean consistency on each set $G \in \mathcal{G}$, but not on sets $G(\overline{\mu}, i)$. Kim et al. [61] gave a practical algorithm for achieving multi-accuracy, and a promising set of experiments suggesting that it could be used to correct for error disparities between different demographic groups on realistic data sets, without sacrificing overall accuracy. Dwork et al. [21] propose notions of fairness and evidence consistency for ranking individuals by their "probability of success" when historical data only records binary outcomes: they show that their proposed notions are closely related to multicalibration of the probability predictions implicitly underlying the rankings. Shabat et al. [85] prove uniform convergence bounds for multicalibration error over hypothesis classes of bounded complexity. In this chapter, as in Hébert-Johnson et al. [43], we learn over hypothesis classes that are only implicitly defined by the set of groups $\mathcal{G}$, and so we bound generalization error in the same manner that Hébert-Johnson et al. [43] do, rather than using uniform convergence arguments.

Conformal prediction is similarly motivated to calibration, but is focused on finding marginal prediction intervals rather than mean estimates: see e.g. Shafer and Vovk [86] for an overview of this literature. Finding marginal prediction intervals on its own (i.e. when prediction intervals only have to be valid on average over the entire population) is easy in the batch/distributional setting, and so just as with the calibration literature, the conformal prediction literature is primarily focused on the online setting in which predictions must be made as points arrive. The most closely related paper related to this literature is Barber et al. [6] who also study the batch distributional setting, and also aim to find marginal prediction intervals which hold not just over the entire population, but on a collection $\mathcal{G}$ of more finely defined sub-populations. Barber et al. [6] obtain prediction intervals of this sort by using a holdout set method from conformal prediction: roughly speaking, they compute empirical $1 - \delta$ coverage intervals on each set $G \in \mathcal{G}$ in the holdout set, and then for an individual x, select the *widest* such interval amongst all groups G that contain x, which is a very conservative choice. The algorithm given by Barber et al. [6] relies on explicit enumeration of groups $G \in \mathcal{G}$ over the holdout set.

There are also several papers in the "fairness in machine learning" literature (in addition to [43, 61]), which are similarly motivated by replacing coarse statistical constraints with constraints that come closer to offering individual guarantees: see Chouldechova and Roth [14] for a survey. Kearns et al. [57, 59] propose to learn classifiers which equalize statistical measures of harm like false positive or negative rates across a very large number of demographic subgroups $G \in \mathcal{G}$, and give practical algorithms for this problem by solving a zero-sum game formulation using techniques from no-regret learning. Kim et al. [60] give algorithms for satisfying a notion of metric fairness which similarly enforces constraints averaged over a large number of subgroups $G \in \mathcal{G}$. Rothblum and Yona [81] define a PAC-like version of the individual fairness notion of Dwork et al. [19] and prove generalization bounds showing how to achieve their notion out of sample on all sufficiently large groups of individuals. Sharifi-Malvajerdi et al. [87] show how to equalize statistical measures of harm like false positive rates across *individuals* — when the rates in question are defined over the randomness of the problem distribution and the classifier. Joseph et al. [49, 51] propose an individual-level notion of "weakly meritocratic fairness" that can be satisfied in bandit learning settings whenever it is possible to compute confidence or prediction intervals around individual labels. They analyze the parametric setting, when actual (conditional) prediction and confidence intervals are possible — but the techniques from our paper could be used for learning in the assumption-free setting (with a slightly weaker notion of fairness) using marginal prediction intervals.

5.3. Preliminaries

Let $\mathcal{X}$ be the domain of features, $\mathcal{Y} = [0, 1]$ the label domain, and $\mathcal{P}$ the true (unknown) probability distribution over $\mathcal{X} \times \mathcal{Y}$.[13] Let $\mathcal{P}_{\mathcal{X}}$ refer to the induced marginal distribution on $\mathcal{X}$ and define $\mathcal{P}_{\mathcal{Y}}$ analogously. Going forward, we refer to the associated random variables with capital letters (e.g. X, Y), and realizations with lowercase letters (x, y).

Let $\mathcal{G} \subseteq 2^{\mathcal{X}}$ be a collection of subsets of $\mathcal{X}$,[14] and for each $G \in \mathcal{G}$, let χ_G denote that associated indicator function, i.e. $\chi_G(x) = 1 \Leftrightarrow x \in G$. For implementation purposes, we assume that each indicator function $\chi_G(x)$ can be computed by a polynomially sized circuit[15].

Definition 15. *Given the true distribution $\mathcal{P}$, we write*

$$\mu = \mathop{\mathbb{E}}_{\mathcal{P}}[y],$$

and its kth central moment is:

$$m_k = \mathop{\mathbb{E}}_{\mathcal{P}}\left[(y-\mu)^k\right].$$

Given a set $S \subseteq \mathcal{X}$, we abuse notation and write

$$\mu(S) = \mathop{\mathbb{E}}_{\mathcal{P}}[y|x \in S] \quad \textit{and} \quad m_k(S) = \mathop{\mathbb{E}}_{\mathcal{P}}\left[(y-\mu(S))^k \,|x \in S\right]$$

for the conditional mean and k^{th} central moment of labels on the distribution conditional on $x \in S$.

We are given n independent draws from $\mathcal{X} \times \mathcal{Y}$ according to distribution $\mathcal{P}$, denoted $D =$

[13] Our approach applies for both finite and infinite feature domains. If $\mathcal{X}$ is uncountably infinite, define an associated measure space, and $\mathcal{P}$ is a countably additive probability measure on this space. We omit the associated notation since it will have no use in what follows.

[14] If $\mathcal{X}$ is uncountably infinite, then $\mathcal{G}$ is a collection of measurable, computable sets. We abuse notation and write $2^{\mathcal{X}}$ to denote this.

[15] Our algorithm in the end will need to manipulate these indicator functions. We might imagine e.g. that $\mathcal{G}$ is the hypothesis class of some learning algorithm for a binary prediction problem, and that the functions $\chi_G(x)$ are particular hypotheses from this class — e.g. linear threshold functions.

$\{(x_b, y_b)\}_{b=1}^n$. The goal is to predict means and higher moments of $\mathcal{Y}|\mathcal{X}$, i.e. to construct functions $\overline{\mu} : \mathcal{X} \to [0,1]$, (we shall refer to this as a mean predictor) and $\overline{m}_k : \mathcal{X} \to [0,1]$ (analogously, k^{th}-moment predictor)—as $\mathcal{Y}$ is the unit interval, means and moments also lie in the unit interval.

To define calibration, we need to reason about all points that receive a particular prediction. For real valued predictors, this can be a measure zero set. One solution is to to restrict attention to predictors that are discretized to lie on the grid $G_m = \{\frac{1}{2m}, \frac{3}{2m}, \ldots, \frac{2m-1}{2m}\}$, for some (large) number m. If one were to do this, the discretization parameter m would be coupled to the error one could ultimately obtain: since it may be inevitable to suffer error at least $1/2m$ if one is restricted to making predictions on a discrete grid. Alternately, one can define calibration by "bucketing" real valued predictions into m buckets of width $\frac{1}{m}$ each. This allows us to treat m (a parameter controlling the fineness of our calibration constraint) as an orthogonal parameter to our calibration error. To that end, given a set $S \subseteq \mathcal{X}$, mean predictor $\overline{\mu}$, and some $i \in [m]$, define

$$S(\overline{\mu}, i) \equiv \left\{ x \in S : \left| \overline{\mu}(x) - \frac{2i-1}{2m} \right| \leq \frac{1}{2m} \right\}$$

to be the set of points in S whose mean predictions fall into the i^{th} bucket, i.e. $[\frac{2i-1}{2m} - \frac{1}{2m}, \frac{2i-1}{2m} + \frac{1}{2m}]$. Analogously, define

$$S(\overline{\mu}, \overline{m}_k, i, j) \equiv \left\{ x \in S : \left| \overline{\mu}(x) - \frac{2i-1}{2m} \right| \leq \frac{1}{2m}, \left| \overline{m}_k(x) - \frac{2j-1}{2m} \right| \leq \frac{1}{2m} \right\}$$

to be the set of points in S that receive mean predictions in the i^{th} bucket *and* k^{th} moment predictions in the j^{th} bucket. Given mean and k^{th} moment predictors $\overline{\mu}$ and $\overline{m}_k$, and any set $S \subseteq \mathcal{X}$ we write

$$\overline{\mu}(S) = \mathop{\mathbb{E}}_{\mathcal{P}}[\overline{\mu}(x)|x \in S] \quad \text{and} \quad \overline{m}_k(S) = \mathop{\mathbb{E}}_{\mathcal{P}}\left[\overline{m}_k(x)|x \in S\right],$$

i.e. $\overline{\mu}(S)$ is the average mean prediction of $\overline{\mu}$ when x's are drawn according to the true distribution, $\mathcal{P}$, conditional on $x \in S$, and $\overline{m}_k(S)$ is the analogous quantity for k'th moment predictions.

To be clear, we will maintain the convention for means and higher moments that quantities with bars refer to predictions ($\overline{\mu}, \overline{m}_k$) and unmodified notation (μ, m_k) refer to true (unknown) population values.

Definition 16 (Consistency)**.** *Call a mean predictor $\overline{\mu}$ (α, ϵ)-mean consistent on a set S if*

$$|\mu\left(S\right) - \overline{\mu}\left(S\right)| \leq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S)} + \epsilon.$$

Similarly, a moment predictor $\overline{m}_k$ is called (α, ϵ)-moment consistent on a set S if:

$$|m_k\left(S\right) - \overline{m}_k\left(S\right)| \leq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S)} + \epsilon.$$

When $\epsilon = 0$, we say $\overline{\mu}$ is α-mean consistent and $\overline{m}_k$ is α-moment consistent. Note that (α, ϵ)-mean consistency implies $(\alpha + \epsilon)$-mean consistency.

Remark 5. *Our notion of consistency on a set S corresponds to error that smoothly degrades with the size (measure) of the set S. This is essential to giving out of sample guarantees. Hébert-Johnson et al. [43] handles this slightly differently, by giving uniform guarantees, but only for sets that have measure at least γ. Our approach of giving smoothly parameterized error guarantees for all sets is only stronger (up to a reparameterization of $\alpha \leftarrow \alpha\gamma$), and makes the analysis of our algorithms more transparent because it corresponds more directly to the guarantees they achieve.*

The following simple observation will be useful in understanding our approach.

Observation 1. *Let $\mathcal{P}$ be a mixture distribution over m component distributions $\mathcal{P}_\ell$ with mixture weights $w_\ell \geq 0$, $\sum_{\ell=1}^{m} w_\ell = 1$. Let $\mu_\ell, m_{k\ell}$ be the mean and k^{th} moment associated* *with $\mathcal{P}_\ell$. Then, we have*

$$m_k = \sum_{\ell=1}^{m} w_\ell \left(\sum_{a=0}^{k} \binom{k}{a} (\mu_\ell - \mu)^{k-a} \, m_{a\ell} \right).$$

If the mixture variables have the same mean, i.e. $\mu_\ell = \mu$ for all ℓ, then, the above expression reduces to:

$$m_k = \sum_{\ell=1}^{m} w_\ell m_{k\ell}.$$

Observation 1 highlights the key challenge: unlike means, higher moments combine non-linearly over mixtures. That is to say, that although $\overline{m}_k(S)$ is defined to be an average over the values $\overline{m}_k(x)$ for $x \in S$, $m_k(S)$ is *not* an average over the values $m_k(x)$ for $x \in S$ for $k > 1$. Observation 1 also makes clear what we are trying to exploit in defining mean-*conditioned* moment calibration: $m_k(S)$ *is* an average over the values $m_k(x)$ for $x \in S$ whenever $\mu(x)$ is constant over S.

We are now ready to define calibration, which asks for mean and moment consistency on particular sets defined by the mean and moment predictors themselves:

Definition 17 (Calibration)**.** *Fix a set $S \subseteq \mathcal{X}$ and a true distribution $\mathcal{P}$.*

1. *A mean predictor $\overline{\mu}$ is (α, ϵ)-mean calibrated on a set S if it is (α, ϵ)-mean consistent on every set $S(\overline{\mu}, i)$, i.e. if for each $i \in [m]$:*

   $$|\mu\left(S(\overline{\mu}, i)\right) - \overline{\mu}(S(\overline{\mu}, i))| \leq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S(\overline{\mu}, i))} + \epsilon.$$

   Again, if $\epsilon = 0$, we say $\overline{\mu}$ is α-mean calibrated.

2. *Predictors $(\overline{\mu}, \overline{m}_k)$ are $(\alpha, \beta, \epsilon)$-mean-conditioned-moment calibrated on a set S if they are (α, ϵ)-mean and (β, ϵ)-moment consistent on every set $S(\overline{\mu}, \overline{m}_k, i, j)$, i.e. if for*

every $i, j \in [m]$:

$$|\mu\left(S(\overline{\mu}, \overline{m}_k, i, j)\right) - \overline{\mu}\left(S(\overline{\mu}, \overline{m}_k, i, j)\right)| \leq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S(\overline{\mu}, \overline{m}_k, i, j))} + \epsilon,$$

$$\text{and } |m_k\left(S(\overline{\mu}, \overline{m}_k, i, j)\right) - \overline{m}_k\left(S(\overline{\mu}, \overline{m}_k, i, j)\right)| \leq \frac{\beta}{\mathcal{P}_{\mathcal{X}}(S(\overline{\mu}, \overline{m}_k, i, j))} + \epsilon.$$

If $\epsilon = 0$, we say $(\overline{\mu}, \overline{m}_k)$ are (α, β)-mean-conditioned-moment calibrated.

We say that $\overline{\mu}, \overline{m}_k$ are (α, ϵ)-multicalibrated and $(\alpha, \beta, \epsilon)$-mean-conditioned-moment multicalibrated with respect to (a collection of sets) $\mathcal{G}$ if they are (α, ϵ)-mean calibrated and $(\alpha, \beta, \epsilon)$-mean conditioned moment calibrated respectively on every $G \in \mathcal{G}$.

Remark 6. *Observe that by construction, the true feature conditional mean and moment functions $\mu(x), m_k(x)$ are mean-conditioned-moment multicalibrated on every collection of sets G. We can view the goal of multicalibration as coming up with mean and moment predictors $\overline{\mu}, \overline{m}_k$ that are almost indistinguishable from the true distributional means and moments, with respect to a class of consistency checks defined by $\mathcal{G}$. Note that it is only because we have defined our goal as mean* conditioned *moment calibration that the true moments $m_k(x)$ of the distribution satisfy these consistency conditions, which are defined as expectations.*

We highlight the difference between calibration and consistency on a given set S in terms of mean prediction $\overline{\mu}$; an analogous discussion applies to higher moments. Consistency requires that the prediction $\overline{\mu}(x)$, averaged over x's in S according to the conditional distribution, approximately equals the true label average $\mu(S)$. It doesn't impose a similar requirement on subsets of S. Therefore, a predictor consistent on S will be correct on average for the set S but could be systematically biased for each prediction in S that it makes.

Calibration on S requires, for every prediction $i \in [m]$, that $\overline{\mu}$ is consistent on the set $S(\overline{\mu}, i)$. That is to say it ensures consistency on every subset of x's in S on which the predictor $\overline{\mu}$ makes predictions in some some fixed bucket i. Exact calibration implies exact consistency, but the reverse is not true.

5.4. Achieving Mean Conditioned Moment Multicalibration

5.4.1. Mean Multicalibration

We summarize an algorithm to achieve mean multicalibration. It is a modest extension to the one in Hébert-Johnson et al. [43] that accommodates arbitrary distributions over a possibly infinite domain and arbitrary initializations. We present it in somewhat greater generality than needed for mean-calibration, because our final algorithm in Section 5.4.3 needs to achieve mean consistency on more sets than are required for mean calibration alone.

For intuition, consider the following mini-max problem, which captures a more difficult problem than mean multicalibration (as there is no restriction at all on the sets S):

$$\min_{\overline{\mu}:\mathcal{X}\to[0,1]} \max_{\substack{S\subseteq\mathcal{X},\\ \lambda\in\{-1,1\}}} \lambda\cdot\mathcal{P}_{\mathcal{X}}(S)\cdot\left(\overline{\mu}(S)-\mu(S)\right).$$

We can associate a zero-sum game with this mini-max problem by viewing the minimization player as a *consistency* player who must commit to a mean predictor $\overline{\mu}$, and viewing the maximization player as an *auditor* who attempts to identify sets S on which the consistency player fails to be mean consistent. Observe that the inclusion of the measure term $\mathcal{P}_{\mathcal{X}}(S)$ in the objective makes the learner's utility function linear in her individual predictions $\overline{\mu}(x)$. There is a strategy for the consistency player that would guarantee her a payoff of 0—or in other words, would guarantee consistency on all possible sets S: she could simply set $\overline{\mu}(x)=\mathbb{E}[y|x]$. This establishes the value of the game, but of course it requires knowledge of $\mathcal{P}$. Given only a finite sample of the data, we will be unable to determine $\mathbb{E}[y|x]$ for all x, and so this strategy is not implementable.

One way to solve our problem absent knowledge of the distribution is to allow the consistency player to play online gradient descent [102] on the set of mean predictors over rounds t, and to allow the auditor to "best respond" at every round, by exhibiting a set S corresponding to a large consistency violation[16]. This is guaranteed to converge quickly to an approximate equilibrium of the game: i.e. a mean predictor satisfying approximate consistency on all sets. If the auditor limits herself to choosing sets $S(\overline{\mu}^t, i)$ corresponding to mean calibration, then we converge quickly to approximate mean calibration. Here we give a direct analysis of a general gradient descent procedure of the sort we need, in terms of the sets that the auditor happens to choose during this interaction. For finite support distributions $\mathcal{P}$, this bound could be derived directly from the regret bound of online projected gradient descent [102] or from the analysis of the similar algorithm in Hébert-Johnson et al. [43]. We reproduce a direct analysis in the Appendix to match the theorem statement we want for distributions which may have infinite support. (Note that for such distributions the mean predictor will have to be maintained implicitly). In Algorithm 3, after each gradient update, we project $\overline{\mu}^t$ back into the set of functions with range $[0, 1]$ using an ℓ_2 projection. Because squared ℓ_2 distance is linearly separable, it can be accomplished by a simple coordinate-wise operation which we write as $\text{project}_{[0,1]}(x) = \min(\max(x, 0), 1)$.

Algorithm 3: Projected Gradient Descent(η) for $\overline{\mu}$

```
Start with an arbitrary initial mean predictor $\overline{\mu}^1 : \mathcal{X} \to [0, 1]$
for $t = 1, \ldots, T$ do
    Audit player plays some $S^t \subseteq \mathcal{X}, \lambda^t \in \{-1, 1\}$
    $\overline{\mu}^{t+1}(x) = \begin{cases} \text{project}_{[0,1]}\left(\overline{\mu}^t(x) - \eta\lambda^t\right) & \text{if } x \in S^t, \\ \overline{\mu}^t(x) & \text{otherwise.} \end{cases}$
end
```

Lemma 25. *For any initial mean predictor $\overline{\mu}^1 \in \mathcal{X} \to [0, 1]$ and any sequence of $(S^t, \lambda^t)_{t=1}^T$, Algorithm 3 satisfies:*

$$\sum_{t=1}^{T} \lambda^t \mathcal{P}_{\mathcal{X}}(S^t) \left(\overline{\mu}^t(S^t) - \mu(S^t)\right) \leq \frac{1}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \mathcal{P}_{\mathcal{X}}(S^t).$$

[16] Because the objective function of our game weights the consistency violations $\overline{\mu}(S) - \mu(S)$ by the measure of the set $\mathcal{P}_{\mathcal{X}}(S)$, these violations are linear functions of the individual predictions $\overline{\mu}(x)$. Thus it suffices to run gradient descent over the space of individual predictions $\mathcal{X}$, rather than the space of all possible functions $\overline{\mu} : \mathcal{X} \to [0, 1]$.

The proof is in the Appendix. A direct consequence of the bound in Lemma 25 is that, when interacting with a consistency player who uses gradient descent with learning rate $\eta = \alpha$, an auditor will be able to find sets that fail to be α-mean consistent for at most $1/\alpha^2$ many rounds.[17] The following theorem is a direct consequence of Lemma 25 — its short proof is in the Appendix.

Theorem 19. *Set $T = \frac{1}{\alpha^2} - 1$ and $\eta = \alpha = \frac{1}{\sqrt{T+1}}$ in Algorithm 3. Assume that for every $t \in [T]$,*

$$\lambda^t \left(\overline{\mu}^t(S^t) - \mu(S^t)\right) \geq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S^t)},$$

Then, for every $S \subseteq \mathcal{X}$, we have

$$\left|\overline{\mu}^{T+1}(S) - \mu(S)\right| \leq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S)}.$$

In particular, if the auditor selects sets $G(\overline{\mu}^t, i)$ that fail to satisfy approximate mean consistency whenever they exist, then we quickly converge to a mean-multicalibrated predictor. Either we reach a state in which $\overline{\mu}^t$ is approximately mean consistent on every set $G(\overline{\mu}^t, i)$ before T rounds, in which case we are done, or after T rounds, the conclusion of Theorem 19 implies not only that we are approximately mean-multicalibrated with respect to $\mathcal{G}$, but that we are approximately mean-consistent on every set.

5.4.2. Pseudo-Moment Consistency

In this section we make a simple observation: Algorithm 3 from Section 5.4.1 for achieving mean consistency and calibration did not depend on *any* properties of the labels y. It would have worked equally well had we invented an arbitrary label for each datapoint x, and asked for mean consistency with respect to that label. Using this observation, we consider a (naïve, and incorrect) attempt at achieving calibration for higher moments — but one that will be a useful subroutine in our final algorithm. Recall that $m_k(S) = \mathbb{E}[(y - \mu(x))^k | x \in S]$.

[17] A somewhat better bound is achievable by using a non-uniform learning rate that depends on the measure of the sets S^t chosen by the auditor; we use a uniform learning rate throughout this chapter for clarity.

If we have a mean predictor $\overline{\mu}(x)$, it is therefore tempting to imagine that each point x is associated with an alternative label $\tilde{y}(x) = \widetilde{m}_{k,\overline{\mu}}(x)$, where:

$$\widetilde{m}_{k,\overline{\mu}}(x) = \mathbb{E}\left[(y - \overline{\mu}(x))^k \middle| x\right].$$

We could then use the algorithm from Section 5.4.1 to construct an predictor $\overline{m}_k$ that was *mean* multicalibrated with respect to these labels. We refer to the property of being mean consistent with respect to the moment-like labels $\widetilde{m}_{k,\overline{\mu}}(x)$ as “pseudo-moment-consistency”:

Definition 18 (Pseudo-Moment-Consistency)**.** *Fixing a mean predictor $\overline{\mu}$, define the k^{th}* pseudo-moment labels *to be $\widetilde{m}_{k,\overline{\mu}}(x) = \mathbb{E}\left[(y - \overline{\mu}(x))^k \middle| x\right]$. A moment predictor $\overline{m}_k$ is (β, ϵ)-*pseudo-moment-consistent *on a set S, with respect to a mean predictor $\overline{\mu}$ if*

$$|\overline{m}_k(S) - \widetilde{m}_{k,\overline{\mu}}(S)| \le \frac{\beta}{\mathcal{P}_{\mathcal{X}}(S)} + \epsilon$$

We simply say β-pseudo-moment consistent if the predictor is $(\beta, 0)$-pesudo-moment-consistent.

We can achieve pseudo-moment consistency using the following gradient descent procedure, analogous to Algorithm 3.

Algorithm 4: Projected Gradient Descent(η) for $\overline{m}_k$

```
Start with an arbitrary initial pseudo-moment predictor ${\overline{m}_k}^1 : \mathcal{X} \to [0, 1]$
for $t = 1, \ldots, T$ do
    Audit player plays some $R^t \subseteq \mathcal{X}, \psi^t \in \{-1, 1\}$
    ${\overline{m}_k}^{t+1}(x) = \begin{cases} \text{project}_{[0,1]}\left({\overline{m}_k}^t(x) - \eta\psi^t\right) & \text{if } x \in R^t, \\ {\overline{m}_k}^t(x) & \text{otherwise.} \end{cases}$
end
```

In particular, we obtain the following theorem, whose proof is deferred to the Appendix.

Theorem 20. *Let $T = \frac{1}{\beta^2} - 1$ and $\eta = \frac{1}{\sqrt{T+1}} = \beta$ in Algorithm 4, and fix any mean* *predictor $\overline{\mu}$, which defines the function $\widetilde{m}_{k,\overline{\mu}}(x)$. Assume that for every $t \in [T]$,*

$$\left|\overline{m}_k{}^t(R^t) - \widetilde{m}_{k,\overline{\mu}}(R^t)\right| \geq \frac{\beta}{\mathcal{P}_{\mathcal{X}}(R^t)},$$

Then, for every $R \subseteq \mathcal{X}$, we have

$$|\overline{m}_k(R) - \widetilde{m}_{k,\overline{\mu}}(R)| \leq \frac{\beta}{\mathcal{P}_{\mathcal{X}}(R)}.$$

i.e. $\overline{m}_k$ is β-pseudo-moment-consistent on every set R.

Now, a guarantee of "pseudo-moment-consistency" is really a guarantee of *mean* consistency with respect to "moment-like" labels $\widetilde{m}_{k,\overline{\mu}}(x)$, and does *not* correspond to moment consistency. This is because moments m_k for $k > 1$ don't combine linearly the way means do: recall Observation 1. But also recall from Observation 1 that higher moments *do* happen to combine linearly if we average only over points that share the same mean.

We take advantage of this to prove the following key lemma: if we achieve pseudo-moment consistency on all sets $G(\overline{\mu}, \overline{m}_k, i, j)$ (for $G \in \mathcal{G}$) with respect to a mean predictor $\overline{\mu}$ that happens also to be mean-consistent on all sets $G(\overline{\mu}, \overline{m}_k, i, j)$, then, the pair of predictors is in fact approximately mean-conditioned moment multicalibrated with respect to $\mathcal{G}$.

Lemma 26. *Assume $\overline{\mu}$ is such that for all $G \in \mathcal{G}$ and $i, j \in [m]$, $\overline{\mu}$ is α-mean consistent on every set $G(\overline{\mu}, \overline{m}_k, i, j)$):*

$$|\overline{\mu}(G(\overline{\mu}, \overline{m}_k, i, j)) - \mu(G(\overline{\mu}, \overline{m}_k, i, j))| \leq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(G(\overline{\mu}, \overline{m}_k, i, j))}.$$

Assume also that $\overline{m}_k$ is β-pseudo-moment-consistent with respect to $\overline{\mu}$ on every set $G(\overline{\mu}, \overline{m}_k, i, j))$ for $G \in \mathcal{G}$ and $i, j \in [m]$:

$$|\overline{m}_k(G(\overline{\mu}, \overline{m}_k, i, j)) - \widetilde{m}_{k,\overline{\mu}}(G(\overline{\mu}, \overline{m}_k, i, j))| \leq \frac{\beta}{\mathcal{P}_{\mathcal{X}}(G(\overline{\mu}, \overline{m}_k, i, j))}.$$

Then, for every $G \in \mathcal{G}$, $i, j \in [m]$, we have

$$|\overline{m}_k(G(\overline{\mu}, \overline{m}_k, i, j)) - m_k(G(\overline{\mu}, \overline{m}_k, i, j))| \leq \frac{\beta + k\alpha}{\mathcal{P}_{\mathcal{X}}(G(\overline{\mu}, \overline{m}_k, i, j)} + \frac{k}{m}.$$

This implies in particular that $(\overline{\mu}, \overline{m}_k)$ are $(\alpha, \beta', \epsilon)$-mean-conditioned moment multicalibrated with respect to $\mathcal{G}$, for $\beta' = \beta + k\alpha$ and $\epsilon = \frac{k}{m}$.

Proof. Fix $G \in \mathcal{G}$ and $i, j \in [m]$ and let $S \equiv G(\overline{\mu}, \overline{m}_k, i, j)$. Because $\overline{\mu}$ is α-mean consistent on S, we have that:

$$|\mu(S) - \overline{\mu}(S)| \leq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S)}. \tag{5.1}$$

We can use this to bound the difference between the true moment $m_k(S)$ and the pseudo-moment $\widetilde{m}_{k,\overline{\mu}}(S)$ on S. First, note that:

$$\begin{aligned} m_k(S) &= \mathop{\mathbb{E}}_{\mathcal{P}}\left[(y - \mu(S))^k \middle| x \in S\right], \\ &= \mathop{\mathbb{E}}_{\mathcal{P}}\left[[(y - \overline{\mu}(x)) + (\overline{\mu}(x) - \mu(S))]^k \middle| x \in S\right]. \end{aligned}$$

We will make use of the following fact:

Lemma 27. *For any $a, b \in [0,1]$, $|a^k - b^k| \leq k|a - b|$.*

Proof. Observe that:

$$|a^k - b^k| = \left|(a-b)\left(\sum_{\ell=0}^{k-1} a^\ell b^{k-1-\ell}\right)\right| \leq |a-b||k(\max(a,b))^{k-1}| \leq k|a-b|. \qquad \square$$

Finally, we conclude that:

$$\begin{aligned}
|m_k(S) - \widetilde{m}_{k,\overline{\mu}}(S)| &= \left| \mathop{\mathbb{E}}_{\mathcal{P}} \left[((y - \overline{\mu}(x)) + (\overline{\mu}(x) - \mu(S)))^k - (y - \overline{\mu}(x))^k \middle| x \in S \right] \right| \\
&\leq k \mathop{\mathbb{E}}_{\mathcal{P}} [|\overline{\mu}(x) - \mu(S)||x \in S] \\
&\leq k \left(\mathop{\mathbb{E}}_{\mathcal{P}} [|\overline{\mu}(x) - \overline{\mu}(S)||x \in S] + |\overline{\mu}(S) - \mu(S)| \right) \\
&\leq k \left(\frac{1}{m} + \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S)} \right).
\end{aligned}$$

The first inequality follows from Lemma 27 with $a = (y - \overline{\mu}(x)) + (\overline{\mu}(x) - \mu(S))$ and $b = y - \overline{\mu}(x)$. The final inequality follows from (5.1) (mean consistency) together with the fact that $\overline{\mu}(x)$ falls within a bucket of width $\frac{1}{m}$ for any $x \in S$ (recall that by definition, $S = G(\overline{\mu}, \overline{m}_k, i, j)$), and so does $\overline{\mu}(S)$

Finally, because $\overline{m}_k$ is β-pseudo-moment consistent on S with respect to $\overline{\mu}$ we can invoke the triangle inequality to conclude:

$$\begin{aligned}
|\overline{m}_k(S) - m_k(S)| &\leq |\overline{m}_k(S) - \widetilde{m}_{k,\overline{\mu}}(S)| + |\widetilde{m}_{k,\overline{\mu}}(S) - m_k(S)|, \\
&\leq \frac{\beta}{\mathcal{P}_{\mathcal{X}}(S)} + k \left(\frac{1}{m} + \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S)} \right).
\end{aligned}$$ □

Lemma 26 reduces the problem of finding mean-conditioned-moment multicalibrated predictors $(\overline{\mu}, \overline{m}_k)$ to the problem of finding a pair of predictors $(\overline{\mu}, \overline{m}_k)$ satisfying mean-consistency and pseudo-moment-consistency on the sets $G(\overline{\mu}, \overline{m}_k, i, j)$. It is unclear how to do this, because these conditions have a circular dependency: pseudo-moment consistency of $\overline{m}_k$ with respect to $\overline{\mu}$ is not defined until we have fixed a mean predictor $\overline{\mu}$, because the "labels" $\widetilde{m}_{k,\overline{\mu}}(x)$ with respect to which pseudo-moment consistency is defined depend on $\overline{\mu}$. On the other hand, the sets $G(\overline{\mu}, \overline{m}_k, i, j)$ on which $\overline{\mu}$ must satisfy mean consistency are not defined until we fix the moment predictor $\overline{m}_k$. The next section is devoted to resolving this circularity and finding predictors satisfying the conditions of Lemma 26.

5.4.3. Mean-Conditioned Moment Multicalibration

We arrive at the last block upon which our main result rests: an alternating gradient descent procedure that on any distribution finds a mean multicalibrated predictor $\overline{\mu}$ together with moment predictors $\{\overline{m}_a\}_{a=2}^k$ such that each pair $(\overline{\mu}, \overline{m}_a)$ is approximately mean-conditioned moment multicalibrated on $\mathcal{G}$. We continue, for clarity's sake, to assume access to the underlying distribution $\mathcal{P}$, and postpone to Section 5.5 the details of implementing this approach with a polynomially sized sample of points. Our strategy is to obtain a set of predictors that together satisfy the hypotheses of Lemma 26: mean consistency and pseudo-moment-consistency on every set of the form $G(\overline{\mu}, \overline{m}_a, i, j)) \subseteq G \in \mathcal{G}$, $1 < a \leq k$, and $i, j \in [m]$. We have already seen in Section 5.4.1 that for a *fixed* collection of sets, a simple gradient-descent procedure can obtain mean consistency on each of the sets. Section 5.4.2 demonstrates that for a *fixed* mean predictor $\overline{\mu}$, a similar gradient descent procedure can obtain pseudo-moment-consistency with respect to $\overline{\mu}$ on each set $G(\overline{\mu}, \overline{m}_a, i, j))$. Our algorithm simply alternates between these two procedures. In rounds t, we maintain hypothesis predictors $\overline{\mu}^t, \{\overline{m}_a^t\}_{a=2}^k$. In alternating rounds, we perform updates of gradient descent using Algorithm 5 to arrive at a mean predictor $\overline{\mu}^t$ that has taken a step towards consistency on sets $G(\overline{\mu}^t, \overline{m}_a^{t-1}, i, j)$, and then using the newly updated mean predictor $\overline{\mu}^t$, run Algorithm 6 to obtain moment predictors $\overline{m}_a^t$ that obtain pseudo-moment-consistency with respect to $\overline{\mu}^t$ on all sets $G(\overline{\mu}^t, \overline{m}_a^t, i, j)$. This is coordinated via a wrapper algorithm, Algorithm 7. We prove this alternating procedure terminates after $1/\alpha^2 - 1$ many rounds and outputs predictors $\overline{\mu}, \{\overline{m}_a\}_{a=2}^k$ that are jointly mean-conditioned moment-multicalibrated.

```
Algorithm 5: MeanConsistencyUpdate($\overline{\mu}, S, \lambda$)
    $\overline{\mu}(x) = \begin{cases} \text{project}_{[0,1]}(\overline{\mu}(x) - \alpha\lambda) & \text{if } x \in S, \\ \overline{\mu}(x) & \text{otherwise.} \end{cases}$
    return $\overline{\mu}$
```

Algorithm 6: PseudoMomentConsistency$(a, \beta, \overline{\mu}, \overline{m}_a, \mathcal{G})$

```
define pseudo-moment labels $\widetilde{m}_{a,\overline{\mu}}(x) = \mathbb{E}\left[(y - \overline{\mu}(x))^a | x\right]$ for all $x$
while $\exists R = G(\overline{\mu}, \overline{m}_a, i, j)$ for some $G \in \mathcal{G}$, $i, j \in [m]$ s.t.
 $|\overline{m}_a(R) - \widetilde{m}_{a,\overline{\mu}}(R)| \geq \frac{\beta}{\mathcal{P}_{\mathcal{X}}(R)}$ do
    $\psi = \mathrm{sign}(\overline{m}_k(R) - \widetilde{m}_{k,\overline{\mu}}(R))$
    $\overline{m}_a(x) = \begin{cases} \mathrm{project}_{[0,1]}(\overline{m}_a(x) - \beta\psi) & \text{if } x \in R \\ \overline{m}_a(x) & \text{otherwise.} \end{cases}$
end
return $\overline{m}_k$
```

Algorithm 7: AlternatingGradientDescent$(\alpha, \beta, \mathcal{G})$

```
initialize $\overline{\mu}^1(x) = 0$ for all $x$
for all $1 < a \leq k$, initialize $\overline{m}_a^1(x) = 0$ for all $x$
$t = 1$
while $\exists S^t = G(\overline{\mu}^t, i)$ or $S^t = G(\overline{\mu}^t, \overline{m}_a^t, i, j)$ for some $G \in \mathcal{G}$, $i, j \in [m]$, $1 < a \leq k$
 s.t. $\left|\overline{\mu}^t(S^t) - \mu(S^t)\right| \geq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S^t)}$ do
    $\lambda^t = \mathrm{sign}(\overline{\mu}(S^t) - \mu(S^t))$
    $\overline{\mu}^{t+1} = \mathrm{MeanConsistencyUpdate}(\overline{\mu}^t, S^t, \lambda^t)$
    for $a = 2, \ldots, k$ do
        $\overline{m}_a^{t+1} = \mathrm{PseudoMomentConsistency}(a, \beta, \overline{\mu}^{t+1}, \overline{m}_a^t, \mathcal{G})$.
    end
    $t = t + 1$
end
return $(\overline{\mu}^t, \{\overline{m}_a^t\}_{a=2}^k)$
```

Theorem 21. *Let T be the final iterate t of Algorithm 7 (i.e. its output is $(\overline{\mu}^T, \{\overline{m}_a^T\}_{a=2}^k)$. Algorithm 7 has the following guarantees:*

1. ***Total Iterations**: The algorithm halts. The final iterate T is s.t. $T \leq \frac{1}{\alpha^2} - 1$. The total number of gradient descent update operations is at most $\left(\frac{1}{\alpha^2} - 1\right)\left(1 + (k-1)\left(\frac{1}{\beta^2} - 1\right)\right)$.*

2. ***Mean multicalibration**: Output $\overline{\mu}^T$ is α-mean multicalibrated with respect to $\mathcal{G}$.*

3. ***Mean Conditioned Moment multicalibration**: For every $a \in \{2, \ldots, k\}$, the pair $(\overline{\mu}^T, \overline{m}_a^T)$ is $(\alpha, \beta + a\alpha, \frac{a}{m})$-mean-conditioned moment-multicalibrated with respect to $\mathcal{G}$.*

Proof. We prove each guarantee in turn.

Total Iterations: Every step t of the while loop in Algorithm 7 performs a gradient descent update using MeanConsistencyUpdate on a pair (λ^t, S^t) such that:

$$\lambda^t \left(\overline{\mu}^t(S^t) - \mu(S^t)\right) \geq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S^t)}.$$

By Theorem 19, this process can continue for at most $T \leq \frac{1}{\alpha^2} - 1$ many iterations. Within each iteration t of the loop, the algorithm makes one call to PseudoMomentConsistency for each $1 < a \leq k$ for a total of $(k-1)$ calls per iteration. Each of these calls performs at most $\frac{1}{\beta^2} - 1$ iterations of gradient descent, by Theorem 20.

Mean multicalibration: Suppose for a contradiction that Algorithm 7 terminates at $t = T$ with output $\overline{\mu}^T$ which is not mean multicalibrated, i.e. there exists a set $S \equiv G(\overline{\mu}^T, i)$ for some $G \in \mathcal{G}$, $i \in [m]$ such that $|\overline{\mu}^t(S) - \mu(S)| \geq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S)}$. Then, by construction of the while loop in Algorithm 7, T cannot be the final iterate of t.

Mean Conditioned Moment multicalibration: The While loop in Algorithm 7 will continue as long as there exists a set $S^t \equiv G(\overline{\mu}^t, \overline{m}_a^t, i, j)$ for some $G \in \mathcal{G}, i, j \in [m]$ such that $|\overline{\mu}^t(S^t) - \mu(S^t)| \geq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S^t)}$. Hence we can conclude that at termination, $\overline{\mu}^T$ is α-mean consistent on every set $G(\overline{\mu}^T, \overline{m}_a^T, i, j)$ for some $G \in \mathcal{G}, i, j \in [m]$.

Moreover, during the final iteration, for each $1 < a \leq k$, $\overline{m}_a^T$ was constructed by running PseudoMomentConsistency$(a, \beta, \overline{\mu}^T, \overline{m}_a^{T-1}, \mathcal{G})$. Therefore, by Theorem 20 we know that $\overline{m}_a^T$ is β-pseudo-moment consistent on every set $G(\overline{\mu}^T, \overline{m}_a^T, i, j)$. To see this, note that if PseudoMomentConsistency runs for $\frac{1}{\beta^2} - 1$ many rounds, then it is β-pseudo-moment consistent on *every* set. On the other hand, the only way it can halt before that many rounds (by construction of the halting condition in its While loop) is if $\overline{m}_a^T$ is β-pseudo-moment consistent on every set $G(\overline{\mu}^T, \overline{m}_a^T, i, j)$.

Therefore, $\overline{\mu}^T$ and $\{\overline{m}_a^T\}_{a=2}^k$ jointly satisfy the conditions of Lemma 26. It follows from the Lemma that they are mean-conditioned moment-multicalibrated at the desired parameters.

□

5.5. Implementation with Finite Sample and Runtime Guarantees

In Section 5.4, we analyzed a version of our algorithm as if we had direct access to the true distribution, $\mathcal{P}$. In particular, in both Algorithm 6 and Algorithm 7, access to $\mathcal{P}$ was needed in (only) two places. First, to identify a set S^t such that $\left|\overline{\mu}^t(S^t) - \mu(S^t)\right| \geq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S^t)}$. Second, to identify a set R such that $|\overline{m}_a(R) - \widetilde{m}_{a,\overline{\mu}}(R)| \geq \frac{\beta}{\mathcal{P}_{\mathcal{X}}(R)}$. In this section, we show how to perform these operations approximately by using a small finite sample of points drawn from $\mathcal{P}$, and hence to obtain a finite sample version of our main result together with sample complexity and running time bounds.

There are two issues at play here: the first issue is purely statistical: how many *samples* are needed to execute the two checks needed to implement our algorithm? Our finite sample algorithm will essentially use a sufficiently large fresh sample of data at every iteration to guarantee uniform convergence of the quantities to be estimated over all of the sets that must be checked at that iteration. The second issue is computational: even if we have enough samples so that we can check in-sample quantities as proxies for the distributional quantities we care about, what is the running time of our algorithm? We are performing gradient descent in a potentially infinite dimensional space, and so we cannot explicitly maintain the weights $\overline{\mu}^t(x), \overline{m}_a^t(x)$ for all x. Instead, we maintain these weights *implicitly* as a weighted linear combination of the indicator functions for each of the sets S^t, R, used to perform updates (recall that we have assumed that each set $G \in \mathcal{G}$ can be represented by an indicator function computed by a polynomially sized circuit, so we have concise implicit representations of every set that our algorithm must manipulate). Ostensibly one must exhaustively enumerate the collection of sets S, R, for which our algorithm must perform some check (in fact their indicator functions), which takes time that scales with $m^2 \cdot |\mathcal{G}|$. We first focus on the statistical problem, showing that the number of *samples* needed to implement our algorithm is small, and then we observe that if we have an *agnostic learning algorithm* for $\mathcal{G}$, we can use it to replace exhaustive enumeration. In both cases—although the details differ—we handle these issues in largely the same way they were handled by Hébert-Johnson et al. [43], so many of the proofs and calculations will be deferred to the Appendix.

Finally, we remark that it is essential that we draw a fresh sample of n data points each time we try to find a set for consistency violation because ℓ, $\overline{\ell}$, as well as the collection of sets $\mathcal{S}$ that we are auditing, are not fixed *a priori* but change adaptively (i.e. as a function of the data) between rounds. Due to the adaptive nature of the statistical tests that need to be performed, we cannot simply union bound over these queries. We remark that we could have applied adaptive data analysis techniques (see e.g. [7, 20, 52]) to partially re-use the data, which would save a quadratic factor in the sample complexity (or for finite data domains, an exponential improvement in some of the existing parameters, at the cost of an additional dependence on $\log |\mathcal{X}|$ by using the private multiplicative weights algorithm of Hardt and Rothblum [40]). This idea is applied in Hébert-Johnson et al. [43]; it applies here in the same manner; interested readers can refer to Hébert-Johnson et al. [43].

5.5.1. Sample Complexity Bounds and a Finite Sample Algorithm via Exhaustive Group Enumeration

First, recall that pseudo-moment consistency is mean consistency with respect to the artificially created label $\widetilde{m}_{k,\overline{\mu}}(x) = (y - \overline{\mu}(x))^k$. To avoid needless repetition, we focus on achieving mean consistency for an arbitrary label defined by a label function $\ell(x, y)$ with a predictor $\overline{\ell}(x)$. Then, auditing for mean consistency for $\overline{\mu}$ and pseudo-moment consistency for $\overline{m}_k$ follows by setting

$$\begin{aligned}\overline{\ell}(x) = \overline{\mu}(x) \quad &\text{and} \quad \ell(x,y) = y \\ \overline{\ell}(x) = \overline{m}_k(x) \quad &\text{and} \quad \ell(x,y) = (y - \overline{\mu}(x))^k\end{aligned}$$

for mean consistency and pseudo-moment consistency respectively. For economy of notation set,

$$\overline{\ell}(S) = \mathop{\mathbb{E}}_{\mathcal{P}}[\overline{\ell}(x)|x \in S] \quad \text{and} \quad \ell(S) = \mathop{\mathbb{E}}_{\mathcal{P}}[\ell(x,y)|x \in S],$$

for all $S \subseteq \mathcal{X}$. For any set $S \subseteq \mathcal{X}$, given a dataset D, we refer to D_S as the subset of the data where the corresponding points lie in S (i.e. $D_S = \{(x,y) \in D : x \in S\}$). If dataset D_S has n' points each drawn independently from $\mathcal{P}$ conditional on $x \in S$, we can appeal to the Chernoff bound (Theorem 28) to argue that empirical averages must be close to their expectations.

We can also appeal to the Chernoff bound to argue that when we sample data points from $\mathcal{P}$, the number of points that fall into some set S (e.g. $G(\overline{\mu}, \overline{m}_k, i, j)$) scales roughly with $n\mathcal{P}_{\mathcal{X}}(S)$ (Lemma 33).

Throughout the execution of our algorithm, we need to audit the current mean and moment estimators for α-mean consistency violations. This is important, because the analysis of the running time of the algorithm (e.g. the fact that it converges after at most $T = \frac{1}{\alpha^2} - 1$ iterations) relies on making a minimum amount of progress guaranteed by α-mean *inconsistency*. In the next lemma, we provide a condition that can be checked using empirical estimates which guarantees α-mean inconsistency (on the true, unknown distribution) whenever the sample is appropriately close to the distribution; it follows from two applications of a Chernoff bound that this approximate closeness condition will occur with high probability. We encapsulate this empirical check in Algorithm 8.

```
Algorithm 8: Auditor($\ell, \overline{\ell}, \alpha, \delta, \{(x_b, y_b)\}_{b=1}^{n'}$)
if $n' > 0$ and $\left|\frac{1}{n'}\sum_{b=1}^{n'} \overline{\ell}(x_b) - \frac{1}{n'}\sum_{b=1}^{n'} \ell(x_b, y_b)\right| - 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n'}} \geq \frac{\alpha}{\frac{n'}{n} - \sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}}$ then
    $\lambda = \mathrm{sign}(\frac{1}{n'}\sum_{b=1}^{n'} \overline{\ell}(x_b) - \frac{1}{n'}\sum_{b=1}^{n'} \ell(x_b, y_b))$
    Output $YES, \lambda$ (A Consistency Violation has been found)
else
    Output $No$
end
```

Definition 19. *Fix any set $S \subseteq \mathcal{X}$. Given a set of n data points D and its associated $D_S = \{(x_b, y_b)\}_{b=1}^{n'}$, we say that D is approximately close to $\mathcal{P}$ with respect to $(S, \ell, \overline{\ell})$, if the following inequalities hold true:*

$$n' > 0$$

$$\left|\frac{n'}{n} - \mathcal{P}_{\mathcal{X}}(S)\right| \leq \sqrt{\frac{\ln(\frac{2}{\delta})}{2n}} \tag{5.2a}$$

$$\left|\frac{1}{n'}\sum_{b=1}^{n'} \overline{\ell}(x_b) - \overline{\ell}(S)\right| \leq \sqrt{\frac{\ln(\frac{2}{\delta})}{2n'}} \tag{5.2b}$$

$$\left|\frac{1}{n'}\sum_{b=1}^{n'} \ell(x_b, y_b) - \ell(S)\right| \leq \sqrt{\frac{\ln(\frac{2}{\delta})}{2n'}} \tag{5.2c}$$

Lemma 28. *Fix any set $S \subseteq \mathcal{X}$. If dataset D is approximately close to $\mathcal{P}$ with respect to $(S, \overline{\ell}, \ell)$, then we have*

$$\mathit{Auditor}(\ell, \overline{\ell}, \alpha, D_S) = (\mathit{YES}, \lambda) \implies \left|\overline{\ell}(S) - \ell(S)\right| \geq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S)} \quad \mathit{and} \quad \lambda = \mathit{sign}(\overline{\ell}(S) - \ell(S))$$

Lemma 28 implies that when we find an empirical consistency violation using Algorithm 8, it is indeed a real α-consistency violation with respect to the true distribution, allowing us to make progress — this guarantees that our algorithm will not run for too many iterations. But we need a converse condition, in order to make sure that we don't halt too early: we must show that if there are no empirical α'-consistency violations for some $\alpha' > \alpha$, then there are also no α-consistency violations with respect to the true distribution. This is what we do in Lemma 29. Observe, that without loss of generality, we can restrict attention to sets such that $\mathcal{P}_{\mathcal{X}}(S) \geq \alpha$ because any estimator in the range $[0,1]$ is trivially α-mean consistent on every set with measure $< \alpha$.

Lemma 29. *Fix any set $S \subseteq \mathcal{X}$ such that $\mathcal{P}_{\mathcal{X}}(S) \geq \alpha$. Assume n is sufficiently large such that $2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}} < \alpha$ If D is approximately close to $\mathcal{P}$ with respect to $(S, \bar{\ell}, \ell)$, we have*

$$\left|\bar{\ell}(S) - \ell(S)\right| \geq \frac{\alpha'}{\mathcal{P}_{\mathcal{X}}(S)} \implies \textit{Auditor}(\ell, \bar{\ell}, \alpha, D_S) = \textit{YES},$$

where $\alpha' = \alpha + 4\sqrt{\frac{1}{2n}\ln(\frac{2}{\delta})} + \left(\alpha - 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right)^{-2}\left(2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right)$.

The Auditor subroutine above performs a consistency check on a single set. We now use it to audit for mean consistency and pseudo-moment consistency across a collection of sets.

Algorithm 9: ConsistencyAuditor($\bar{\ell}, \ell, \alpha, \delta, D, \mathcal{S}$)

```
for $S \in \mathcal{S}$ do
    if $|D_S| > 0$ and Auditor($\ell, \bar{\ell}, \alpha, \delta, D_S$) $= YES, \lambda$ then
        return $S, \lambda$
    end
end
return $NULL$
```

Corollary 3. *Fix $\bar{\ell}, \ell, \alpha, \delta$, and a collection of sets $\mathcal{S}$. Given a set of n points D drawn i.i.d. from $\mathcal{P}$ where $\alpha > 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}$, ConsistencyAuditor($\overline{\mu}, \alpha, D, \mathcal{S}$) has the following guarantee with probability $1 - 3\delta|\mathcal{S}|$ over the randomness of D:*

1. *If ConsistencyAuditor does output some set S and λ, then*

$$\left|\bar{\ell}(S) - \ell(S)\right| \geq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S)} \quad \textit{and} \quad \lambda = \textit{sign}(\bar{\ell}(S) - \ell(S)).$$

2. *If ConsistencyAuditor outputs NULL, then for all $S \in \mathcal{S}$,*

$$\left|\overline{\ell}(S) - \ell(S)\right| \leq \frac{\alpha'}{\mathcal{P}_{\mathcal{X}}(S)},$$

where $\alpha' = \alpha + 4\sqrt{\frac{1}{2n}\ln(\frac{2}{\delta})} + \left(\alpha - 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right)^{-2}\left(2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right)$.

Thus, to detect a set S with α-mean consistency violation in line 7 of Algorithm 7, we can leverage Algorithm 9 by drawing a fresh sample of size n and setting $\overline{\ell}(x) = \overline{\mu}(x)$, $\ell(x, y) = y$, and $\mathcal{S} = \{G(\overline{\mu}, i) : G \in \mathcal{G}, i \in [m]\} \cup \{G(\overline{\mu}, \overline{m}_a, i, j) : G \in \mathcal{G}, i, j \in [m], a \in \{2, \ldots, k\}\}$. Likewise, for every $a \in \{2, \ldots, k\}$, to detect a set S with β-pseudo-moment consistency violation in line 6 of Algorithm 6, we can leverage Algorithm 9 by drawing a fresh sample of size n and setting $\overline{\ell}(x) = \overline{m}_k(x)$, $\ell(x, y) = (y - \overline{\mu}(x))^k$, and $\mathcal{S} = \{G(\overline{\mu}, \overline{m}_a, i, j) : G \in \mathcal{G}, i, j \in [m]\}$. We write out the pseudocode of this process below.

Algorithm 10: PseudoMomentConsistencyFinite($a, \beta, \delta, \overline{\mu}, \overline{m}_a, n, \mathcal{G}$)

```
$D = \{(x_b, y_b)\}_{b=1}^{n} \sim \mathcal{P}^n$
$\mathcal{S} = \{G(\overline{\mu}, \overline{m}_a, i, j) : G \in \mathcal{G}, i, j \in [m]\}$
$\overline{\ell}(x) = \overline{m}_a(x)$
$\ell(x, y) = (y - \overline{\mu}(x))^a$
$R, \psi = \text{ConsistencyAuditor}(\overline{\ell}, \ell, \beta, \delta, D, \mathcal{S})$
while $R, \psi \neq NULL$ do
    $\overline{m}_a(x) = \begin{cases} \text{project}_{[0,1]}(\overline{m}_a(x) - \beta\psi) & \text{if } x \in R \\ \overline{m}_a(x) & \text{otherwise.} \end{cases}$
    $D = \{(x_b, y_b)\}_{b=1}^{n} \sim \mathcal{P}^n$
    $\mathcal{S} = \{G(\overline{\mu}, \overline{m}_a, i, j) : G \in \mathcal{G}, i, j \in [m]\}$
    $\overline{\ell}(x) = \overline{m}_a(x)$
    $R, \psi = \text{ConsistencyAuditor}(\overline{\ell}, \ell, \beta, \delta, D, \mathcal{S})$
end
return $\overline{m}_k$
```

```
Algorithm 11: AlternatingGradientDescentFinite($\alpha, \beta, \delta, n, \mathcal{G}$)
Initialize $\overline{\mu}^1(x) = 0$ for all $x$
For all $1 < a \leq k$, initialize $\overline{m}_a^1(x) = 0$ for all $x$
$t = 1$
$\overline{\ell}^t(x) = \overline{\mu}(x)$
$\ell(x, y) = y$
$D^t = \{(x_b, y_b)\}_{b=1}^n \sim \mathcal{P}^n$
$\mathcal{S}^t = \{G(\overline{\mu}^t, i) : G \in \mathcal{G}, i \in [m]\} \cup \{G(\overline{\mu}^t, \overline{m}_a^t, i, j) : G \in \mathcal{G}, i, j \in [m], a \in \{2, \ldots, k\}\}$
$S^t, \lambda^t$ = ConsistencyAuditor($\overline{\ell}^t, \ell, \alpha, \delta, D, \mathcal{S}$)
while $S^t, \lambda^t \neq NULL$ do
    $\overline{\mu}^{t+1}$ = MeanConsistencyUpdate($\overline{\mu}^t, S^t, \lambda^t$)
    for $a = 2, \ldots, k$ do
        $\overline{m}_a^{t+1}$ = PseudoMomentConsistencyFinite($a, \beta, \delta, \overline{\mu}^{t+1}, \overline{m}_a^t, n, \mathcal{G}$).
    end
    $t = t + 1$
    $\overline{\ell}^t(x) = \overline{\mu}(x)$
    $D^t = \{(x_b, y_b)\}_{b=1}^n \sim \mathcal{P}^n$
    $\mathcal{S}^t = \{G(\overline{\mu}^t, i) : G \in \mathcal{G}, i \in [m]\} \cup \{G(\overline{\mu}^t, \overline{m}_a^t, i, j) : G \in \mathcal{G}, i, j \in [m], a \in \{2, \ldots, k\}\}$
    $S^t, \lambda^t$ = ConsistencyAuditor($\overline{\ell}^t, \ell, \alpha, \delta, D^t, \mathcal{S}^t$)
end
return $(\overline{\mu}^t, \{\overline{m}_a^t\}_{a=2}^k)$
```

Theorem 22. *Let T be the final iterate of Algorithm 11. If $2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}} \leq \alpha$ and $2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}} \leq \beta$, we have the following guarantees:*

1. ***Total Iterations**: With probability $1 - 3\delta|\mathcal{G}|Q_\alpha\left((m^2 + m) + m^2 Q_\beta\right)$ over the randomness of our samples, the final iterate T is s.t. $T \leq \frac{1}{\alpha^2} - 1$ and the total number of*

gradient descent update operations will be at most Q, where

$$Q_\alpha = \frac{1}{\alpha^2} - 1, Q_\beta = (k-1)\left(\frac{1}{\beta^2} - 1\right), Q = Q_\alpha(1+Q_\beta).$$

In particular, the algorithm uses at most nQ samples from $\mathcal{P}$.

2. ***Mean multicalibration**: With probability $1 - 3\delta(m^2+m)|\mathcal{G}|$, output $\overline{\mu}^T$ is α'-mean multicalibrated with respect to $\mathcal{G}$ where*

$$\alpha' = \alpha + 4\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}} + \left(\alpha - 2\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}\right)^{-2}\left(2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right).$$

3. ***Mean Conditioned Moment multicalibration**: With probability $1 - 3\delta|\mathcal{G}|(km^2 + m)$, for any $a \in \{2, \ldots, k\}$, pair $(\overline{\mu}^T, \overline{m}_a^T)$ is $(\alpha', a\alpha' + \beta', \frac{a}{m})$-mean-conditioned-moment multicalibrated where*

$$\alpha' = \alpha + 4\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}} + \left(\alpha - 2\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}\right)^{-2}\left(2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right)$$

$$\beta' = \beta + 4\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}} + \left(\beta - 2\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}\right)^{-2}\left(2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right)$$

The following corollary derives the sample complexity of our algorithm, implied by Theorem 22, for a target set of parameters. Observe that the sample complexity is polynomial in $k, m, 1/\alpha, 1/\beta, 1/\epsilon, \log(1/\delta)$, and $\log|\mathcal{G}|$.

Corollary 4. *Fix target parameters α', β', δ' and $\epsilon > 0$ such that $\epsilon < \alpha'$ and $\epsilon < \beta'$. Define*

$$\overline{Q} = \frac{6|\mathcal{G}|km^2}{\left(\frac{\alpha'-\epsilon}{6+\frac{2}{\epsilon^2}}\right)^2 \left(\frac{\beta'-\epsilon}{6+\frac{2}{\epsilon^2}}\right)^2}, \quad \delta = \frac{\delta'}{\max(3|\mathcal{G}|(km^2+m), \overline{Q})},$$

$$n_\alpha = \frac{\ln(\frac{2\overline{Q}}{\delta})}{2\left(\frac{\alpha'-\epsilon}{6+\frac{2}{\epsilon^2}}\right)^2}, \quad n_\beta = \frac{\ln(\frac{2\overline{Q}}{\delta})}{2\left(\frac{\beta'-\epsilon}{6+\frac{2}{\epsilon^2}}\right)^2}.$$

Then, AlternatingGradientDescentFinite$(\alpha, \beta, \delta, n, \mathcal{G})$ where

$$\alpha = 2\sqrt{\frac{\ln(\frac{2\overline{Q}}{\delta})}{2n_\alpha}} + \epsilon, \beta = 2\sqrt{\frac{\ln(\frac{2\overline{Q}}{\delta})}{2n_\beta}} + \epsilon,$$

$$n = \max\left(\frac{\ln(\frac{2\overline{Q}}{\delta})}{\ln(\frac{2}{\delta})}n_\alpha, \frac{\ln(\frac{2\overline{Q}}{\delta})}{\ln(\frac{2}{\delta})}n_\beta, \frac{2\ln(\frac{2}{\delta})}{\alpha^2}, \frac{2\ln(\frac{2}{\delta})}{\beta^2}\right)$$

has the following guarantees with probability $1-\delta'$:

1. *The total number of gradient descent updates will be at most Q, where Q is as defined in Theorem 22.*

2. *$\overline{\mu}^T$ is α'-mean-multicalibrated.*

3. *For every $a \in \{2, \ldots, k\}$, $(\overline{\mu}^T, \overline{m}_a^T)$ is $(\alpha', a\alpha' + \beta', \frac{a}{m})$-mean-conditioned-moment multicalibrated.*

Finally, we state the running time of the algorithm in the following Theorem.

Theorem 23. *With probability $1 - 3\delta|\mathcal{G}|Q_\alpha\left((m^2+m) + m^2 Q_\beta\right)$, the running time of Algorithm 11 is $O\left(Q|\mathcal{G}|m^2 n\right) = O\left(\frac{k|\mathcal{G}|m^2 n}{\alpha^2\beta^2}\right)$ where Q_α, Q_β, and Q are as defined in Theorem 22.*

5.5.2. Oracle Efficient Implementation

In Section 5.5.1 we analyzed an algorithm that had favorable sample-complexity bounds, but was computationally expensive when $\mathcal{G}$ was large: although it ran for only a small number of iterations, each iteration required a complete enumeration of every set in $\mathcal{G}$. In this section, we show how to replace this expensive step with a call to an algorithm which can solve learning problems over $\mathcal{G}$, if one is available. Because the remaining portion of the algorithm is computationally efficient — even if $\mathcal{G}$ is very large — this yields what is sometimes known as an "oracle efficient algorithm". Similar reductions have been given in Hébert-Johnson et al. [43], Kearns et al. [59] and Kim et al. [61].

Definition 20. *For some $\rho \in [0, 1]$ and non-increasing function $p : \mathbb{N} \to [0, 1]$, A is a (ρ, p)-agnostic learning oracle for hypothesis class $\mathcal{H} \subseteq 2^{\mathcal{X}}$ with respect to a label function $r(x, y) \in [-1, 1]$, if for any distribution $\mathcal{P}$, given n random samples from $\mathcal{P}$, it outputs $f : \mathcal{X} \to \{0, 1\}$ such that with probability $1 - p(n)$,*

$$\mathop{\mathbb{E}}_{(x,y)\sim\mathcal{P}}[f(x) \cdot r(x, y)] + \rho \geq \sup_{h\in\mathcal{H}} \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{P}}[h(x) \cdot r(x, y)].$$

We write $\tau(n)$ to denote the running time of the oracle A when n data points are used, which we assume is at least $\Omega(n)$.

Remark 7. *A more common definition of an agnostic learning oracle would use hypotheses with range $\{-1, 1\}$ rather than $\{0, 1\}$. But this definition will be more convenient for us, and is equivalent (up to a constant factor in the parameters) via a linear transformation.*

We will use a learning algorithm for any class $\mathcal{H}$ such that $\mathcal{G} \subseteq \mathcal{H}$ to replace the set enumeration steps of our algorithm. In particular, to find a set of the form $G(\overline{\mu}, i)$ on which our existing predictor $\overline{\mu}$ fails to be mean consistent, we run our learning algorithm on the subset of our sample that intersects with $\mathcal{X}(\overline{\mu}, i)$, labelled with the positive and negative *residuals* of our predictor — i.e. on the labels $r^+_R(x, y) = \overline{\mu}(x) - y$ and $r^-_R = y - \overline{\mu}(x)$. Similarly, to find a set of the form $G(\overline{\mu}, \overline{m}_a, i, j)$, we run our learning algorithm on the sets $\mathcal{X}(\overline{\mu}, \overline{m}_a, i, j)$ labeled with both the positive and negative residuals. Finding sets on which we fail to be moment pseudo-consistent with respect to $\overline{\mu}$ is similar. All in all, this requires $O(k \cdot m^2)$ runs of our learning algorithm per gradient descent step, replacing the complete enumeration of the collection of sets $\mathcal{G}$. We make this process more precise in Algorithm 13 and state the guarantees in Theorem 24. We also include the pseudocode for the correspondingly updated AlternatingGradientDescentFinite using Algorithm 13 as the auditing subroutine in the appendix – see Algorithm 15.

Algorithm 12: LearningOracleConsistencyAuditor($\overline{\ell}, \ell, \alpha, \delta, D, R, A$)

```
$r_R^+(x,y) = \begin{cases} \overline{\ell}(x) - \ell(x,y) & \text{if } x \in R \\ 0 & \text{otherwise} \end{cases}$   and   $D_R^+ = \{(x_b, r_R^+(x,y))\}_{b=1}^n$
$r_R^-(x,y) = \begin{cases} \ell(x,y) - \overline{\ell}(x) & \text{if } x \in R \\ 0 & \text{otherwise} \end{cases}$   and   $D_R^- = \{(x_b, r_R^-(x,y))\}_{b=1}^n$
$\chi_{S^+} = A(D_R^+, \mathcal{H})$
$\chi_{S^-} = A(D_R^-, \mathcal{H})$
return $(S^+, S^-)$
```

Algorithm 13: LearningOracleConsistencyAuditorWrapper($\overline{\ell}, \ell, \alpha, \delta, D, D^{\text{check}}, \mathbb{R}, A$)

```
$\mathcal{V} = \{\}$ for $R \in \mathbb{R}$ do
    if $|D_R| > 0$ then
        $S^+, S^-$ = LearningOracleConsistencyAuditor($\overline{\ell}, \ell, \alpha, \delta, D, D^{check}, R, A$)
        $\mathcal{V} = \mathcal{V} \cup \{(S^+ \cap R), (S^- \cap R)\}$
    end
end
return ConsistencyAuditor($\overline{\ell}, \ell, \alpha, D^{\text{check}}, \mathcal{V}$)
```

First we observe that the objective of the agnostic learning oracle on the sets we run it on corresponds directly to the (positive and negative) violation of mean consistency on these sets, weighted by the measure of the sets.

Lemma 30. *For each $R \in \mathbb{R}$ and any χ_S:*

$$\mathop{\mathbb{E}}_{(x,y)} \left[\chi_S(x) \cdot r_R^+(x,y)\right] = \mathcal{P}_{\mathcal{X}}(R \cap S) \left(\overline{\ell}(R \cap S) - \ell(R \cap S)\right)$$

$$\mathop{\mathbb{E}}_{(x,y)}\left[\chi_S(x)\cdot r_R^-(x,y)\right]=\mathcal{P}_{\mathcal{X}}(R\cap S)\left(\ell(R\cap S)-\bar{\ell}(R\cap S)\right)$$

Using this, we can show that our learning oracle based consistency auditor has comparable guarantees to the consistency auditor that operated via set enumeration:

Theorem 24. *Assume n is sufficiently large such that $\alpha > 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}$. Algorithm 13 has the following guarantees:*

1. *If it returns some S and λ, then with probability $1-3\delta|\mathbb{R}|$ over the randomness of D^{check},*

$$|\bar{\ell}(S)-\ell(S)|\geq\frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S)}.$$

2. *If it returns $NULL$, then with probability $1-|\mathbb{R}|(3\delta+2p(n))$ over the randomness of D and D^{check}, for all $\chi_S\in\mathcal{H}$ and $R\in\mathbb{R}$,*

$$|\bar{\ell}(R\cap S)-\ell(R\cap S)|\leq\frac{\alpha'+\rho}{\mathcal{P}_{\mathcal{X}}(R\cap S)},$$

where α' is as defined in Corollary 3.

Observe that when $\mathcal{G}\subseteq\mathcal{H}$ and $\mathbb{R}=\{\mathcal{X}(\overline{\mu},\overline{m}_a,i,j):i,j\in[m]\}$, then, the collection of intersections $R\cap S$ over all $\chi_S\in\mathcal{H}$ and $R\in\mathbb{R}$ contains $\{G(\overline{\mu},\overline{m}_a,i,j):G\in\mathcal{G},i,j\in[m]\}$. The same observation applies when $\mathbb{R}=\{\mathcal{X}(\overline{\mu},i):i\in[m]\}\cup\{\mathcal{X}(\overline{\mu},\overline{m}_a,i,j):i,j\in[m]\}$ – the collection of intersections includes $\{G(\overline{\mu},i):G\in\mathcal{G},i\in[m]\}\cup\{\mathcal{X}(\overline{\mu},\overline{m}_a,i,j):G\in\mathcal{G},i,j\in[m]\}$.

We now present the guarantees of a version of AlternatingGradientDescent that uses Algorithm 13 as the auditor. Its pseudo-code can be found as Algorithm 15 in the appendix. We elide the proof as it is almost identical to that of Theorem 21 and Theorem 22.

Theorem 25. *Assume $\mathcal{G}\subseteq\mathcal{H}$. Let T be the final iterate of Algorithm 15. If $2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\leq\alpha$, $2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\leq\beta$, and $\mathcal{G}\subseteq\mathcal{H}$, we have the following guarantees:*

1. ***Total Iterations**: With probability $1 - 3\delta Q_\alpha \left((m^2+m) + m^2 Q_\beta\right)$ over the randomness of our samples, the final iterate T is such that $T \le \frac{1}{\alpha^2} - 1$ and the total number of gradient descent update operations will be at most Q, where Q_α, Q_β, and Q are all as defined in Theorem 22.*

   In particular, the algorithm uses at most $O(nQ)$ samples from $\mathcal{P}$.

2. ***Mean multicalibration**: With probability $1 - (m^2+m)(3\delta + 2p(n))$, output $\overline{\mu}^T$ is α''-mean multicalibrated with respect to $\mathcal{G}$ where $\alpha'' = \alpha' + \rho$ and α' is as defined in Theorem 22.*

3. ***Mean Conditioned Moment multicalibration**: With probability $1-(km^2+m)(3\delta+2p(n))$, for any $a \in \{2, \ldots, k\}$, pair $(\overline{\mu}^T, \overline{m}_a^T)$ is $(\alpha'', a\alpha'' + \beta'', \frac{a}{m})$-mean-conditioned-moment calibrated where $\beta'' = \beta' + \rho$ and β' is as defined in Theorem 22.*

Finally, we state the running time of Algorithm 15.

Theorem 26. *With probability at least $1 - 3\delta Q_\alpha \left((m^2+m) + m^2 Q_\beta\right)$, the running time of Algorithm 15 is bounded by $O(Qm^2\tau(n))$, where Q is the total number of gradient descent operations as defined in Theorem 22.*

5.6. Marginal Prediction Intervals

We now present an application of our results. Given subgroups of interest $\mathcal{G}$, we have shown how to to construct a multicalibrated mean predictor $\overline{\mu}$ and moment predictors $(\overline{m}_a)_{a=2}^k$ that are simultaneously mean-conditioned moment-multicalibrated. A key question is whether we can use mean-conditioned moment multicalibrated predictors in applications in which we would use real distributional moments, were they available.

In this section, we show that the answer is *yes* in an important application. Mean-conditioned moment multicalibrated predictors can be used in tail bounds just as real moments could be to compute prediction intervals. Where real moments would yield prediction intervals conditioned on an individual vector of features x, mean-conditioned moment-multicalibrated predictors when used in the same computations yield marginal prediction intervals that are simultaneously valid for every sufficiently large subgroup. In particular, given a coverage failure probability δ and a group size γ we show how to construct just from mean and moment predictions, for every $x \in X$, an interval $I(x, \gamma)$ such that for every $G \in \mathcal{G}$ and for every pair of predictions i, j such that $G(\overline{\mu}, \overline{m}_a, i, j)$ has mass at least γ we have: $\Pr_{(x,y)}[y \in I(x, \gamma) | x \in G(\overline{\mu}, \overline{m}_a, i, j)] \geq 1 - \delta$.

Recall the following tail inequality (a simple consequence of Markov's inequality: when $k = 2$, it is known as Chebyshev's inequality):

Lemma 31. *Let X be a random variable with mean μ. Then for even k, $t > 0$:*

$$\Pr[|X - \mu| \geq t] \leq \frac{\mathbb{E}[(X - \mu)^k]}{t^k}.$$

Suppose we knew the *real* moments $m_k(x)$ of the distribution on y conditional on features x: A direct application of the above lemma would allow us to conclude that for any even moment k:

$$\Pr\left[y \notin \left[\mu(x) - \left(\frac{m_k(x)}{\delta} \right)^{\frac{1}{k}}, \mu(x) + \left(\frac{m_k(x)}{\delta} \right)^{\frac{1}{k}} \right] \Bigg| x \right] \leq \delta.$$

Bounds of this form are simple, but also strong: there is always an integer moment k such that the above bound is at least as tight as a generalized Chernoff bound[18] [78], and only the first $k \leq O(\log(1/\delta))$ moments are necessary to match Chernoff bounds at coverage probability $1 - \delta$ [83].

If we had *exactly* mean-conditioned moment-calibrated predictors $(\overline{\mu}, \overline{m}_k)$ for some k even, over a set of groups G, we would obtain *exactly* the same bound using these predictors as

[18] Chernoff's bound is $\Pr[X \geq t] \leq \inf_{\theta \geq 0} M_X(\theta) e^{-\theta t}$, where $M_X(\theta)$ is the moment generating function for X

a marginal prediction interval: i.e. we would obtain for every $G \in \mathcal{G}$, and every i, j:

$$\Pr_{(x,y)}\left[y \notin \left[\overline{\mu}(x) - \left(\frac{\overline{m}_k(x)}{\delta}\right)^{1/k}, \overline{\mu}(x) + \left(\frac{\overline{m}_k(x)}{\delta}\right)^{1/k} \right] \middle| x \in G(\overline{\mu}, \overline{m}_k, i, j) \right] \leq \delta.$$

This is because mean-conditioned moment-multicalibrated predictors actually do provide real distributional moments, over the selection of a random point within any set $G(\overline{\mu}, \overline{m}_k, i, j)$. Of course we only have approximately mean-conditioned-moment multicalibrated predictors. Given $(\alpha, \beta, \epsilon)$-mean-conditioned-moment multicalibrated predictors $(\overline{\mu}, \overline{m}_k)$, k even, with respect to some collection of groups $\mathcal{G}$, we can endow our predictions with (marginal) prediction intervals that have coverage probability $1 - \delta$ as follows. The *width* of our prediction interval for a point x will be:

$$\Delta_{\gamma,k}(x) = \frac{\alpha}{\gamma} + \epsilon + \frac{1}{m} + \left(\frac{\overline{m}_k(x) + \epsilon + \frac{1}{m} + \frac{\beta}{\gamma}}{\delta}\right)^{\frac{1}{k}},$$

Our prediction interval for x will be centered at its predicted mean, and is defined as follows:

$$I_{\gamma,k}(x) = [\overline{\mu}(x) - \Delta_{\gamma,k}(x), \overline{\mu}(x) + \Delta_{\gamma,k}(x)].$$

These are valid marginal prediction intervals as averaged over *any* set of the form $G(\overline{\mu}, \overline{m}_k, i, j)$ that has measure larger than γ. Note that all of the approximation terms $\alpha, \beta, \epsilon, 1/m$ are terms that we can drive to zero at polynomial cost in running time and sample complexity.

Theorem 27. *Assume that $\overline{\mu}, \overline{m}_k$ is $(\alpha, \beta, \epsilon)$-mean-conditioned moment multicalibrated with respect to $\mathcal{G}$, with k even. Then for any group $G \in \mathcal{G}$ and any set $G(\overline{\mu}, \overline{m}_k, i, j)$ such that $\mathcal{P}_{\mathcal{X}}[G(\overline{\mu}, \overline{m}_k, i, j)] \geq \gamma$, we have:*

$$\mathcal{P}[y \notin I_{\gamma,k}(x) | x \in G(\overline{\mu}, \overline{m}_k, i, j)] \leq \delta$$

Proof. To see this note that:

$$
\begin{aligned}
&\mathcal{P}[y \notin I_{\gamma,k}(x) | x \in G(\overline{\mu}, \overline{m}_k, i, j)] \\
&= \mathcal{P}\left[|y - \overline{\mu}(x)| \geq \frac{\alpha}{\gamma} + \frac{1}{m} + \epsilon + \left(\frac{\overline{m}_k(x) + \frac{1}{m} + \epsilon + \frac{\beta}{\gamma}}{\delta} \right)^{\frac{1}{k}} \middle| x \in G(\overline{\mu}, \overline{m}_k, i, j) \right] \\
&\leq \mathcal{P}\left[|y - \overline{\mu}(G(\overline{\mu}, \overline{m}_k, i, j))| \geq \frac{\alpha}{\gamma} + \epsilon + \left(\frac{\overline{m}_k(G(\overline{\mu}, \overline{m}_k, i, j)) + \frac{\beta}{\gamma} + \epsilon}{\delta} \right)^{\frac{1}{k}} \middle| x \in G(\overline{\mu}, \overline{m}_k, i, j) \right] \\
&\leq \mathcal{P}\left[|y - \mu(G(\overline{\mu}, \overline{m}_k, i, j))| \geq \left(\frac{m_k(G(\overline{\mu}, \overline{m}_k, i, j))}{\delta} \right)^{\frac{1}{k}} \middle| x \in G(\overline{\mu}, \overline{m}_k, i, j) \right] \\
&\leq \delta
\end{aligned}
$$

Here, the first inequality follows from the fact that all $x \in G(\overline{\mu}, \overline{m}_k, i, j)$ are (by definition) such that $|\overline{\mu}(x) - \frac{i}{m}| \leq \frac{1}{2m}$ and $|\overline{m}_k(x) - \frac{j}{m}| \leq \frac{1}{2m}$, and hence $|\overline{\mu}(x) - \overline{\mu}(G(\overline{\mu}, \overline{m}_k, i, j))| \leq \frac{1}{m}$ and $|\overline{m}_k(x) - \overline{m}_k(G(\overline{\mu}, \overline{m}_k, i, j)))| \leq \frac{1}{m}$. The second inequality follows from the definition of (α, β)-mean conditioned moment multicalibration and the fact that $\mathcal{P}[G(\overline{\mu}, \overline{m}_k, i, j)] \geq \gamma$. Finally, once we have replaced our mean and moment estimates with the true mean and moment of $G(\overline{\mu}, \overline{m}_k, i, j)$, the final inequality follows as an application of Lemma 31.

□

This theorem shows how—given $(\alpha, \beta, \epsilon)$ mean-conditioned moment-multicalibrated predictors $\overline{\mu}, \overline{m}_k$—we can construct prediction intervals for any set $G(\overline{\mu}, \overline{m}_k, i, j)$ with probability larger than γ.[19] However, we have more information available to us: We have mean conditioned moment-calibrated predictors for all moments 2 thru k, $(\overline{m}_a)_{a=2}^{k}$. A straightforward valid solution is to pick some even moment a s.t. $1 < a \leq k$, and then construct prediction

[19] We showed this just for *even* moments k — but a version of Lemma 31 also holds for k odd, i.e. for any r.v. X with mean μ, any number k, and any $t > 0$, we have $\Pr[|X - \mu| \geq t] \leq \frac{\mathbb{E}[|(X-\mu)^k|]}{t^k}$. We can use this to construct valid confidence intervals using "absolute central moments" of any degree, even or odd. Note also that our algorithms and analysis apply identically if the goal was to provide mean-conditioned, multicalibrated predictors of absolute central moments (i.e. the analog of Definition 17 but where instead of $m_k(\cdot)$, we calibrate our predictor to the analogous absolute central moment).

intervals as above. We could optimize our choice of a so as to minimize e.g. the expected width of the prediction intervals over a random choice of x. But this leads to the question of whether we can do better by using more than one moment estimator at a time.

In Appendix 5.D, we show that this problem reduces to the venerable submodular-cost set-cover problem. Known approximation guarantees for this problem are relatively weak in this context (scaling with $\log |\mathcal{X}|$, which will typically be linear in the *dimension* of the data). We leave the question of how to optimally use multiple mean-conditioned moment multicalibrated predictors—taking advantage of multiple moments simultaneously—to future research.

Appendix

5.A. Details and Proofs from Section 5.4.1

Lemma 25. *For any initial mean predictor $\overline{\mu}^1 \in \mathcal{X} \to [0,1]$ and any sequence of $(S^t, \lambda^t)_{t=1}^T$, Algorithm 3 satisfies:*

$$\sum_{t=1}^{T} \lambda^t \mathcal{P}_{\mathcal{X}}(S^t)\left(\overline{\mu}^t(S^t) - \mu(S^t)\right) \leq \frac{1}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T} \mathcal{P}_{\mathcal{X}}(S^t).$$

Proof. Because $\overline{\mu}^{t+1}(x) = \overline{\mu}^t(x)$ for $x \notin S^t$, we can lower bound the "progress" made towards μ at each round $t \in [T]$ as:

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\mathcal{P}}\left[(\overline{\mu}^t(x) - \mu(x))^2 - (\overline{\mu}^{t+1}(x) - \mu(x))^2\right] \\
&= \mathcal{P}_{\mathcal{X}}(S^t)\mathop{\mathbb{E}}_{\mathcal{P}}\left[(\overline{\mu}^t(x) - \mu(x))^2 - (\overline{\mu}^{t+1}(x) - \mu(x))^2 | x \in S^t\right] \\
&\geq \mathcal{P}_{\mathcal{X}}(S^t)\mathop{\mathbb{E}}_{\mathcal{P}}\left[(\overline{\mu}^t(x) - \mu(x))^2 - (\overline{\mu}^t(x) - \eta\lambda^t - \mu(x))^2 | x \in S^t\right] \\
&= \mathcal{P}_{\mathcal{X}}(S^t)\mathop{\mathbb{E}}_{\mathcal{P}}\left[(\overline{\mu}^t(x) - \mu(x))^2 - \left((\overline{\mu}^t(x) - \mu(x))^2 - 2\eta\lambda^t(\overline{\mu}^t(x) - \mu(x)) + (\eta\lambda^t)^2\right) | x \in S^t\right] \\
&= \mathcal{P}_{\mathcal{X}}(S^t)\mathop{\mathbb{E}}_{\mathcal{P}}\left[2\eta\lambda^t(\overline{\mu}^t(x) - \mu(x)) | x \in S^t\right] - \mathcal{P}_{\mathcal{X}}(S^t)(\eta\lambda^t)^2 \\
&= 2\eta\lambda^t\mathcal{P}_{\mathcal{X}}(S^t)\left(\overline{\mu}^t(S^t) - \mu(S^t)\right) - \mathcal{P}_{\mathcal{X}}(S^t)(\eta\lambda^t)^2.
\end{aligned}$$

The inequality would be an equality if we did not project $\overline{\mu}^t$ into the range $[0,1]$. Performing the projection only decreases its ℓ_2 distance to μ, which yields the inequality. Rearranging terms and observing that $(\lambda^t)^2 = 1$ yields

$$\lambda^t\mathcal{P}_{\mathcal{X}}(S^t)\left(\overline{\mu}^t(S^t) - \mu(S^t)\right) \leq \frac{1}{2\eta}\mathop{\mathbb{E}}_{\mathcal{P}}\left[(\overline{\mu}^t(x) - \mu(x))^2 - (\overline{\mu}^{t+1}(x) - \mu(x))^2\right] + \frac{\eta\mathcal{P}_{\mathcal{X}}(S^t)}{2}$$

Therefore we have that

$$\begin{aligned}
&\sum_{t=1}^{T} \lambda^t \mathcal{P}_{\mathcal{X}}(S^t)\left(\overline{\mu}^t(S^t) - \mu(S^t)\right) \\
\leq& \sum_{t=1}^{T} \left(\frac{1}{2\eta} \mathop{\mathbb{E}}_{\mathcal{P}} \left[(\overline{\mu}^t(x) - \mu(x))^2 - (\overline{\mu}^{t+1}(x) - \mu(x))^2 \right] + \frac{\eta \mathcal{P}_{\mathcal{X}}(S^t)}{2} \right) \\
=& \frac{1}{2\eta} \mathop{\mathbb{E}}_{\mathcal{P}} \left[(\overline{\mu}^1(x) - \mu(x))^2 - (\overline{\mu}^{T+1}(x) - \mu(x))^2 \right] + \frac{\eta}{2} \sum_{t=1}^{T} \mathcal{P}_{\mathcal{X}}(S^t) \\
\leq& \frac{1}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \mathcal{P}_{\mathcal{X}}(S^t)
\end{aligned}$$

as desired. The last inequality follows because $\overline{\mu}^1(x), \mu(x)$, and $\overline{\mu}^{T+1}(x)$ all fall in $[0, 1]$. □

Theorem 19. *Set $T = \frac{1}{\alpha^2} - 1$ and $\eta = \alpha = \frac{1}{\sqrt{T+1}}$ in Algorithm 3. Assume that for every $t \in [T]$,*

$$\lambda^t \left(\overline{\mu}^t(S^t) - \mu(S^t)\right) \geq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S^t)},$$

Then, for every $S \subseteq \mathcal{X}$, we have

$$\left|\overline{\mu}^{T+1}(S) - \mu(S)\right| \leq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S)}.$$

Proof. Fix any set $S \subseteq \mathcal{X}$ and imagine extending the sequence by setting $S^{T+1} = S$ and setting $\lambda^{T+1} = \text{sign}(\overline{\mu}^{T+1}(S) - \mu(S))$. By Lemma 25, we would then have:

$$\begin{aligned}
\sum_{t=1}^{T+1} \lambda^t \mathcal{P}_{\mathcal{X}}(S^t)\left(\overline{\mu}^t(S^t) - \mu(S^t)\right) &\leq \frac{1}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T+1} \mathcal{P}_{\mathcal{X}}(S^t) \\
&\leq \frac{1}{2\eta} + \frac{\eta(T+1)}{2} \\
&\leq \sqrt{T+1} && \text{(substituting } \eta = \frac{1}{\sqrt{T+1}}\text{)}
\end{aligned}$$

We can then peel off the last term in the sum corresponding to $S^{T+1} = S$ to obtain:

$$\begin{aligned}
\mathcal{P}_{\mathcal{X}}(S) \left|\overline{\mu}^{T+1}(S) - \mu(S)\right| &\leq \sqrt{T+1} - \sum_{t=1}^{T} \lambda^t \mathcal{P}_{\mathcal{X}}(S^t) \left(\overline{\mu}^t(S^t) - \mu(S^t)\right) \\
&\leq \sqrt{T+1} - \alpha T && \text{(by assumption)} \\
&= \alpha && \text{(since } T = \frac{1}{\alpha^2} - 1\text{)}
\end{aligned}$$

which completes the proof. □

5.B. Details and Proofs from Section 5.4.2

For intuition, we can think of the pseudo-moment calibration algorithm as playing the following zero sum game using projected online gradient descent against an adversary who plays best responses. Recall that $\overline{\mu}$ is a fixed quantity so that $\widetilde{m}_{k,\overline{\mu}}$ is well defined.

$$\min_{\overline{m}_k} \max_{\substack{R \subseteq \mathcal{X} \\ \psi \in \{-1,1\}}} \psi \mathcal{P}_{\mathcal{X}}(R) \left(\overline{m}_k(R) - \widetilde{m}_{k,\overline{\mu}}(R)\right).$$

Lemma 32. *For any arbitrary $\overline{m}_k{}^1 : \mathcal{X} \to [0,1]$ and any sequence of $(R^t, \psi^t)_{t=1}^T$, we have that*

$$\sum_{t=1}^{T} \psi^t \mathcal{P}_{\mathcal{X}}(R^t) \left(\overline{m}_k{}^t(R) - \widetilde{m}_{k,\overline{\mu}}(R)\right) \leq \frac{1}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \mathcal{P}_{\mathcal{X}}(R^t)$$

Proof.

$$
\begin{aligned}
&\mathop{\mathbb{E}}_{\mathcal{P}}\left[(\overline{m}_k{}^t(x) - \widetilde{m}_{k,\overline{\mu}}(x))^2 - (\overline{m}_k{}^{t+1}(x) - \widetilde{m}_{k,\overline{\mu}}(x))^2\right] \\
=&\mathcal{P}_{\mathcal{X}}(R^t) \mathop{\mathbb{E}}_{\mathcal{P}}\left[(\overline{m}_k{}^t(x) - \widetilde{m}_{k,\overline{\mu}}(x_i))^2 - (\overline{m}_k{}^{t+1}(x) - \widetilde{m}_{k,\overline{\mu}}(x))^2 | x \in R^t\right] \\
\geq&\mathcal{P}_{\mathcal{X}}(R^t) \mathop{\mathbb{E}}_{\mathcal{P}}\left[(\overline{m}_k{}^t(x) - \widetilde{m}_{k,\overline{\mu}}(x))^2 - (\overline{m}_k{}^t(x) - \eta\psi^t - \widetilde{m}_{k,\overline{\mu}}(x))^2 | x \in R^t\right] \\
=&\mathcal{P}_{\mathcal{X}}(R^t) \mathop{\mathbb{E}}_{\mathcal{P}}\Bigg[(\overline{m}_k{}^t(x) - \widetilde{m}_{k,\overline{\mu}}(x))^2 \\
&\quad - \left((\overline{m}_k{}^t(x) - \widetilde{m}_{k,\overline{\mu}}(x))^2 - 2\eta\psi^t(\overline{m}_k{}^t(x) - \widetilde{m}_{k,\overline{\mu}}(x)) + (\eta\psi^t)^2\right) | x \in R^t \Bigg] \\
=&2\eta\psi^t\mathcal{P}_{\mathcal{X}}(R^t) \mathop{\mathbb{E}}_{\mathcal{P}}\left[\overline{m}_k{}^t(x) - \widetilde{m}_{k,\overline{\mu}}(x)) | x \in R^t\right] - \mathcal{P}_{\mathcal{X}}(R^t)(\eta\psi^t)^2
\end{aligned}
$$

Here the inequality comes from the fact that projection can only make the ℓ_2 norm smaller. Rearranging terms and observing that $(\psi^t)^2 = 1$ yields

$$
\begin{aligned}
&\psi^t\mathcal{P}_{\mathcal{X}}(R^t) \mathop{\mathbb{E}}_{\mathcal{P}}\left[\overline{m}_k{}^t(x) - \widetilde{m}_{k,\overline{\mu}}(x) | x \in R^t\right] \\
&\leq \frac{1}{2\eta} \mathop{\mathbb{E}}_{\mathcal{P}}\left[(\overline{m}_k{}^t(x) - \widetilde{m}_{k,\overline{\mu}}(x))^2 - (\overline{m}_k{}^{t+1}(x) - \widetilde{m}_{k,\overline{\mu}}(x_i))^2\right] + \frac{\eta\mathcal{P}_{\mathcal{X}}(R^t)}{2}.
\end{aligned}
$$

Plugging this inequality back into the regret, we get

$$
\begin{aligned}
&\sum_{t=1}^{T} \psi^t\mathcal{P}_{\mathcal{X}}(R^t)\left(\overline{m}_k{}^t(R) - \widetilde{m}_{k,\overline{\mu}}(R)\right) \\
&= \sum_{t=1}^{T} \psi^t\mathcal{P}_{\mathcal{X}}(R^t) \mathop{\mathbb{E}}_{\mathcal{P}}\left[\overline{m}_k{}^t(x) - \widetilde{m}_{k,\overline{\mu}}(x) | x \in R^t\right] \\
&\leq \sum_{t=1}^{T} \left(\frac{1}{2\eta} \mathop{\mathbb{E}}_{\mathcal{P}}\left[(\overline{m}_k{}^t(x) - \widetilde{m}_{k,\overline{\mu}}(x))^2 - (\overline{m}_k{}^{t+1}(x) - \widetilde{m}_{k,\overline{\mu}}(x_i))^2\right] + \frac{\eta\mathcal{P}_{\mathcal{X}}(R^t)}{2}\right) \\
&= \frac{1}{2\eta}\left(\sum_{x\in\mathcal{X}} \mathop{\mathbb{E}}_{\mathcal{P}}\left[(\overline{m}_k{}^1(x) - \widetilde{m}_{k,\overline{\mu}}(x))^2 - (\overline{m}_k{}^{T+1}(x) - \widetilde{m}_{k,\overline{\mu}}(x_i))^2\right]\right) + \frac{\eta}{2}\sum_{t=1}^{T}\mathcal{P}_{\mathcal{X}}(R^t) \\
&\leq \frac{1}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\mathcal{P}_{\mathcal{X}}(R^t)
\end{aligned}
$$

as desired. The last inequality follows because $\overline{m}_k{}^1(x), \widetilde{m}_{k,\overline{\mu}}(x)$, and $\overline{m}_k{}^{T+1}(x)$ all fall in $[0,1]$. □

Theorem 20. *Let $T = \frac{1}{\beta^2} - 1$ and $\eta = \frac{1}{\sqrt{T+1}} = \beta$ in Algorithm 4, and fix any mean predictor $\overline{\mu}$, which defines the function $\widetilde{m}_{k,\overline{\mu}}(x)$. Assume that for every $t \in [T]$,*

$$\left|\overline{m}_k{}^t(R^t) - \widetilde{m}_{k,\overline{\mu}}(R^t)\right| \geq \frac{\beta}{\mathcal{P}_{\mathcal{X}}(R^t)},$$

Then, for every $R \subseteq \mathcal{X}$, we have

$$|\overline{m}_k(R) - \widetilde{m}_{k,\overline{\mu}}(R)| \leq \frac{\beta}{\mathcal{P}_{\mathcal{X}}(R)}.$$

i.e. $\overline{m}_k$ is β-pseudo-moment-consistent on every set R.

Proof. Set $R^{T+1} = R$. From Lemma 32, we get

$$\begin{aligned}\sum_{t=1}^{T+1} \psi^t \mathcal{P}_{\mathcal{X}}(R^t)\left(\overline{m}_k{}^t(R^t) - \widetilde{m}_{k,\overline{\mu}}(R^t)\right) &\leq \frac{1}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T+1} \mathcal{P}_{\mathcal{X}}(R^t) \leq \sqrt{T+1} \\ \Longrightarrow \mathcal{P}_{\mathcal{X}}(R^{T+1})\left|\overline{m}_k{}^{T+1}(R^{T+1}) - \widetilde{m}_{k,\overline{\mu}}(R^{T+1})\right| &\leq \sqrt{T+1} - \beta T = \beta\end{aligned}$$

□

5.C. Details and Proofs from Section 5.5

Theorem 28 (Chernoff Bound)**.** *Fix distribution $\mathcal{P}$ and some function $f(x,y) \in [0,1]$. Let $\{(x_b, y_b)\}_{b=1}^n$ be n points sampled i.i.d. from $\mathcal{P}$. Then, we have for any $\delta \in [0,1]$,*

$$\Pr_{\{(x_b,y_b)\}_{b=1}^n \sim \mathcal{P}^n}\left[\left|\frac{1}{n}\sum_{b=1}^n f(x_b, y_b) - \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{P}}[f(x,y)]\right| \geq \sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right] \leq \delta.$$

Lemma 33. *For any set $S \subseteq \mathcal{X}$,*

$$\Pr_{D\sim\mathcal{P}^n}\left[\left|\frac{1}{n}\sum_{b=1}^{n}\mathbb{1}(x_b \in S) - \mathcal{P}_{\mathcal{X}}(S)\right| > \sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right] \leq \delta$$

Proof. We apply a Chernoff bound (Theorem 28) with $f(x,y) = \mathbb{1}(x \in S)$. Observe that $\mathbb{E}[f(x,y)] = \mathcal{P}_{\mathcal{X}}(S)$. □

Lemma 28. *Fix any set $S \subseteq \mathcal{X}$. If dataset D is approximately close to $\mathcal{P}$ with respect to $(S, \bar{\ell}, \ell)$, then we have*

$$Auditor(\ell, \bar{\ell}, \alpha, D_S) = (YES, \lambda) \Longrightarrow \left|\bar{\ell}(S) - \ell(S)\right| \geq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S)} \quad and \quad \lambda = sign(\bar{\ell}(S) - \ell(S))$$

Proof. To see this, observe that

$$\begin{aligned}
&\left|\bar{\ell}(S) - \ell(S)\right| \\
&\geq \left|\frac{1}{n'}\sum_{b=1}^{n'}\bar{\ell}(x_b) - \frac{1}{n'}\sum_{b=1}^{n'}\ell(x_b, y_b)\right| - 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n'}} \\
&\geq \frac{\alpha}{\frac{n'}{n} - \sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}} \\
&\geq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S)}
\end{aligned}$$

Here, the first inequality follows from the (5.2b) and (5.2c), the second from the condition of Algorithm 8, and the last inequality follows from (5.2a).

Finally, if $\frac{1}{n'}\sum_{b=1}^{n'}\bar{\ell}(x_b) \geq \frac{1}{n'}\sum_{b=1}^{n'}\ell(x_b, y_b)$, then

$$\bar{\ell}(S) \geq \frac{1}{n'}\sum_{b=1}^{n'}\bar{\ell}(x_b) - \sqrt{\frac{\ln(\frac{2}{\delta})}{2n'}} \geq \frac{1}{n'}\sum_{b=1}^{n'}\ell(x_b, y_b) + \sqrt{\frac{\ln(\frac{2}{\delta})}{2n'}} \geq \ell(S).$$

The same argument applies when $\frac{1}{n'}\sum_{b=1}^{n'}\bar{\ell}(x_b) < \frac{1}{n'}\sum_{b=1}^{n'}\ell(x_b, y_b)$. Therefore, $\text{sign}(\frac{1}{n'}\sum_{b=1}^{n'}\bar{\ell}(x_b) - \frac{1}{n'}\sum_{b=1}^{n'}\ell(x_b, y_b)) = \text{sign}(\bar{\ell}(S) - \ell(S))$.

□

Lemma 29. *Fix any set $S \subseteq \mathcal{X}$ such that $\mathcal{P}_{\mathcal{X}}(S) \geq \alpha$. Assume n is sufficiently large such that $2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}} < \alpha$ If D is approximately close to $\mathcal{P}$ with respect to $(S, \overline{\ell}, \ell)$, we have*

$$\left|\overline{\ell}(S) - \ell(S)\right| \geq \frac{\alpha'}{\mathcal{P}_{\mathcal{X}}(S)} \Longrightarrow \mathit{Auditor}(\ell, \overline{\ell}, \alpha, D_S) = \mathit{YES},$$

where $\alpha' = \alpha + 4\sqrt{\frac{1}{2n}\ln(\frac{2}{\delta})} + \left(\alpha - 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right)^{-2}\left(2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right)$.

Proof. The pre-condition implies that

$$\begin{aligned}
&\left|\frac{1}{n'}\sum_{b=1}^{n'}\overline{\ell}(x_b) - \frac{1}{n'}\sum_{b=1}^{n'}\ell(x_b, y_b)\right| - 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n'}} \\
&\geq \left|\overline{\ell}(S) - \ell(S)\right| - 4\sqrt{\frac{\ln(\frac{2}{\delta})}{2n'}} \\
&\geq \frac{\alpha'}{\mathcal{P}_{\mathcal{X}}(S)} - 4\sqrt{\frac{\ln(\frac{2}{\delta})}{2n'}}.
\end{aligned}$$

Therefore it is sufficient to show that $\frac{\alpha'}{\mathcal{P}_{\mathcal{X}}(S)} - 4\sqrt{\frac{\ln(\frac{2}{\delta})}{2n'}} \geq \frac{\alpha}{\frac{n'}{n} - \sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}}$.

$$
\begin{aligned}
\frac{\alpha'}{\mathcal{P}_{\mathcal{X}}(S)} &= \frac{\alpha + 4\sqrt{\frac{1}{2n}\ln(\frac{2}{\delta})} + \left(\alpha - 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right)^{-2}\left(2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right)}{\mathcal{P}_{\mathcal{X}}(S)} \\
&\geq \frac{\alpha + 4\sqrt{\frac{1}{2n}\ln(\frac{2}{\delta})}}{\mathcal{P}_{\mathcal{X}}(S) - 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}} && (5.3) \\
&\geq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S) - 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}} + \frac{4\sqrt{\frac{1}{2n}\ln(\frac{2}{\delta})}}{\mathcal{P}_{\mathcal{X}}(S) - \sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}} \\
&\geq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S) - 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}} + 4\sqrt{\frac{\ln(\frac{2}{\delta})}{2n(\mathcal{P}_{\mathcal{X}}(S) - \sqrt{\frac{\ln(\frac{2}{\delta})}{2n}})}} && \forall x \in [0,1] : x \leq \sqrt{x} \\
&\geq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S) - 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}} + 4\sqrt{\frac{\ln(\frac{2}{\delta})}{2n'}} && \text{by (5.2a)} \\
&\geq \frac{\alpha}{\frac{n'}{n} - \sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}} + 4\sqrt{\frac{\ln(\frac{2}{\delta})}{2n'}} && \text{by (5.2a)}
\end{aligned}
$$

Inequality (5.3) comes from Lemma 34, where we plug in $x = \mathcal{P}_{\mathcal{X}}(S)$, $c = \alpha + 4\sqrt{\frac{1}{2n}\ln(\frac{2}{\delta})}$ and $\epsilon = 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}$. □

Lemma 34. *For any* $0 < \epsilon \leq \alpha \leq x \leq 1$ *and* $0 < c \leq 1$*,*

$$\frac{c + \frac{\epsilon}{(\alpha-\epsilon)^2}}{x} \geq \frac{c}{x - \epsilon}$$

Proof. Because

$$\frac{c + \frac{\epsilon}{(\alpha-\epsilon)^2}}{x} \geq \frac{c}{x} + \frac{\epsilon}{(\alpha - \epsilon)^2},$$

it is sufficient to show that

$$\frac{c}{x} + \frac{\epsilon}{(\alpha - \epsilon)^2} \geq \frac{c}{x - \epsilon}.$$

Because $f(x) = \frac{c}{x}$ is convex, it's easy to see that:

$$f(x-\epsilon) + \epsilon f'(x-\epsilon) \leq f(x)$$
$$\frac{c}{x-\epsilon} - \frac{c\epsilon}{(x-\epsilon)^2} \leq \frac{c}{x}.$$

Now, because $\epsilon \leq \alpha \leq x$ and $0 < c \leq 1$, we have

$$\frac{c}{x-\epsilon} - \frac{\epsilon}{(\alpha-\epsilon)^2} \leq \frac{c}{x}. \qquad \square$$

Corollary 3. *Fix $\bar{\ell}, \ell, \alpha, \delta$, and a collection of sets $\mathcal{S}$. Given a set of n points D drawn i.i.d. from $\mathcal{P}$ where $\alpha > 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}$, ConsistencyAuditor$(\overline{\mu}, \alpha, D, \mathcal{S})$ has the following guarantee with probability $1 - 3\delta|\mathcal{S}|$ over the randomness of D:*

1. *If ConsistencyAuditor does output some set S and λ, then*

$$\left|\bar{\ell}(S) - \ell(S)\right| \geq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S)} \quad \text{and} \quad \lambda = \mathit{sign}(\bar{\ell}(S) - \ell(S)).$$

2. *If ConsistencyAuditor outputs NULL, then for all $S \in \mathcal{S}$,*

$$\left|\bar{\ell}(S) - \ell(S)\right| \leq \frac{\alpha'}{\mathcal{P}_{\mathcal{X}}(S)},$$

where $\alpha' = \alpha + 4\sqrt{\frac{1}{2n}\ln(\frac{2}{\delta})} + \left(\alpha - 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right)^{-2}\left(2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right).$

Proof. For each set $S \in \mathcal{S}$, we write $D_S = \{(x_b^S, y_b^S)\}_{b=1}^{n'_S}$ to denote the points from D that fall in S.

First, by union bounding the failure probabilities of Lemma 33 over every $S \in \mathcal{S}$, we have with probability $1 - \delta|\mathcal{S}|$,

$$\left|\frac{n'_S}{n} - \mathcal{P}_{\mathcal{X}}(S)\right| > \sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}.$$

We apply the Chernoff bound again for every set S where $n'_S > 0$ and take the union bound to argue that with probability at least $1 - 2|\mathcal{S}|\delta$, for all such sets S where $n'_S > 0$,

$$\left| \frac{1}{n'_S} \sum_{b=1}^{n'_S} \bar{\ell}(x_b^S) - \bar{\ell}(S) \right| \leq \sqrt{\frac{\ln(\frac{2}{\delta})}{2n'_S}}$$
$$\left| \frac{1}{n'_S} \sum_{b=1}^{n'_S} \ell(x_b^S, y_b^S) - \ell(S) \right| \leq \sqrt{\frac{\ln(\frac{2}{\delta})}{2n'_S}}.$$

Observe that despite the fact that n'_S is not fixed before we draw the sample, we can still apply a Chernoff bound here because for *every* realized value of n'_S, the distribution, conditional on the value of n'_S, of points (x, y) such that $(x, y) \in S$ remains a product distribution, with individual such points distributed as $\mathcal{P}|x \in S$. Now, we go through each scenario:

1. **ConsistencyAuditor outputs some set S and λ**: In this case, S would have been returned only if $n'_S > 0$ due to the if condition in Algorithm 8. Therefore, D must be approximately close to $\mathcal{P}$ with respect to $(S, \bar{\ell}, \ell)$. By Lemma 29, we have

$$|\bar{\ell}(S) - \ell(S)| \geq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S)} \quad \text{and} \quad \lambda = \text{sign}(\bar{\ell}(S) - \ell(S))$$

2. **ConsistencyAuditor outputs** *NULL*: For any set S, $\mathcal{P}_{\mathcal{X}}(S) < \alpha$ directly implies that
$$|\bar{\ell}(S) - \ell(S)| \leq 1 < \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S)} \leq \frac{\alpha'}{\mathcal{P}_{\mathcal{X}}(S)}.$$

   Therefore, we focus only on sets S where $\mathcal{P}_{\mathcal{X}}(S) \geq \alpha$. For these sets, we have $n'_S > 0$ because

$$\frac{n'_S}{n} \geq \mathcal{P}_{\mathcal{X}}(S) - \sqrt{\frac{\ln(\frac{2}{\delta})}{2n}} \geq \alpha - \sqrt{\frac{\ln(\frac{2}{\delta})}{2n}} > \sqrt{\frac{\ln(\frac{2}{\delta})}{2n}} > 0,$$

   as we assumed $\alpha > 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}$. Therefore, for every set S where $\mathcal{P}_{\mathcal{X}}(S) \geq \alpha$, we must have that D must be approximately close to $\mathcal{P}$ with respect to $(S, \bar{\ell}, \ell)$. Thus, by applying Lemma 29 to these sets S, we have

$$\left|\bar{\ell}(S) - \ell(S)\right| \leq \frac{\alpha'}{\mathcal{P}_{\mathcal{X}}(S)},$$

where $\alpha' = \alpha + 4\sqrt{\frac{1}{2n}\ln(\frac{2}{\delta})} + \left(\alpha - 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right)^{-2}\left(2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right)$.

□

Theorem 22. *Let T be the final iterate of Algorithm 11. If $2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}} \leq \alpha$ and $2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}} \leq \beta$, we have the following guarantees:*

1. ***Total Iterations**: With probability $1 - 3\delta|\mathcal{G}|Q_\alpha\left((m^2+m) + m^2 Q_\beta\right)$ over the randomness of our samples, the final iterate T is s.t. $T \leq \frac{1}{\alpha^2} - 1$ and the total number of gradient descent update operations will be at most Q, where*

$$Q_\alpha = \frac{1}{\alpha^2} - 1, Q_\beta = (k-1)\left(\frac{1}{\beta^2} - 1\right), Q = Q_\alpha(1 + Q_\beta).$$

In particular, the algorithm uses at most nQ samples from $\mathcal{P}$.

2. ***Mean multicalibration**: With probability $1 - 3\delta(m^2+m)|\mathcal{G}|$, output $\overline{\mu}^T$ is α'-mean multicalibrated with respect to $\mathcal{G}$ where*

$$\alpha' = \alpha + 4\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}} + \left(\alpha - 2\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}\right)^{-2}\left(2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right).$$

3. ***Mean Conditioned Moment multicalibration**: With probability $1 - 3\delta|\mathcal{G}|(km^2 + m)$, for any $a \in \{2, \ldots, k\}$, pair $(\overline{\mu}^T, \overline{m}_a^T)$ is $(\alpha', a\alpha' + \beta', \frac{a}{m})$-mean-conditioned-*

moment multicalibrated where

$$\alpha' = \alpha + 4\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}} + \left(\alpha - 2\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}\right)^{-2}\left(2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right)$$

$$\beta' = \beta + 4\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}} + \left(\beta - 2\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}\right)^{-2}\left(2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right)$$

Proof. We prove each guarantee in turn.

Total Iterations: As argued in Theorem 21, if the auditor can successfully find a set S on which there is α-mean inconsistency and β-pseudo-moment inconsistency respectively in AlternatingGradientDescentFinite (Algorithm 11) and PseudoMomentConsistencyFinite (Algorithm 10), Theorem 19 guarantees that T will be at most $\frac{1}{\alpha^2} - 1$ and Theorem 20 guarantees that the total number of gradient descent operations in each PseudoMomentconsistencyFinite will be at most $\frac{1}{\beta^2} - 1$. Then, because in each iteration of AlternatingGradientDescentFinite, there are $k-1$ calls to PseudoMomentconsistencyFinite, the total number of number of gradient descent operations will be at most $Q = Q_\alpha(1 + Q_\beta)$ where $Q_\alpha = \frac{1}{\alpha^2} - 1$ and $Q_\beta = (k-1)(\frac{1}{\beta^2} - 1)$.

Therefore, it is sufficient for us to show that there is α-mean inconsistency and β-pseudo-moment inconsistency on every S^t and R returned by ConsistencyAuditor (Algorithm 9) for AlternatingGradientDescentFinite and PseudoMomentconsistencyFinite respectively.

For AlternatingGradientDescentFinite, because we set $\overline{\ell}^t(x) = \overline{\mu}^t(x)$, $\ell(x, y) = y$, and $\mathcal{S}^t = \{G(\overline{\mu}, i) : G \in \mathcal{G}, i \in [m]\} \cup \{G(\overline{\mu}, \overline{m}_a, i, j) : G \in \mathcal{G}, i, j \in [m], a \in \{2, \ldots, k\}\}$, Corollary 3 guarantees that with probability $1 - 3\delta|\mathcal{G}|(m^2 + m)$, $\overline{\mu}^t$ is α-mean inconsistent on S^t with n $\lambda^t = \text{sign}(\overline{\mu}^t(S^t) - \mu(S^t))$ as desired. Because T is at most Q_α, by a union bound, $\overline{\mu}^t$ is α-mean-inconsistent on S^t for every $t \in [T]$ with probability $1 - 3\delta|\mathcal{G}|(m^2 + m)Q_\alpha$.

Likewise, for PseudoMomentconsistencyFinite, we set $\overline{\ell}(x) = \overline{m}_k(x)$, $\ell(x, y) = (y - \overline{\mu}(x))^a$, and $\mathcal{S} = \{G(\overline{\mu}, \overline{m}_a, i, j) : G \in \mathcal{G}, i, j \in [m], a \in \{2, \ldots, k\}\}$. Hence, by union bounding over every $a \in \{2, \ldots, k\}$, Corollary 3 promises us that with probability $1 - 3\delta|\mathcal{G}|m^2 Q_\alpha Q_\beta$, $\overline{m}_a$ is β-pseudo-moment inconsistent on R throughout every iteration of PseudoMomentConsistencyFinite for every $a \in \{2, \ldots, k\}$ and $\psi = \text{sign}(\overline{m}_a(S) - \widetilde{m}_{a,\overline{\mu}}(S)$ as desired. Note that there are a total of Q_β calls to ConsistencyAuditor from each PseudoMomentConsistencyFinite, which is invoked a total of Q_α many times.

Mean Multi-Calibration: Our algorithm halts only if ConsistencyAuditor doesn't find S in AlternatingGradientDescentFinite. Corollary 3 promises us that with probability $1 - 3\delta(m^2 + m)|\mathcal{G}|$, $\overline{\mu}^T$ must be α'-mean-consistent on every set $S \in \mathcal{S}^T$. Because $\mathcal{S}$ includes $\{G(\overline{\mu}^T, i) : G \in \mathcal{G}, i \in [m]\}$, it must be that $\overline{\mu}^T$ is α'-mean multi-calibrated with respect to $\mathcal{G}$.

Mean Conditioned Moment Multi-Calibration: In the last round T, consider each $\overline{m}_a^T$ for $a \in \{2, \ldots, k\}$. PseudoMomentConsistencyFinite returns $\overline{m}_a^T$ only if ConsistencyAuditor doesn't return any R. Corollary 3 guarantees us that with probability $1 - 3\delta m^2|\mathcal{G}|$, $\overline{m}_a^T$ must be β'-pseudo-moment-consistent. Because $\overline{\mu}^T$ is α'-mean consistent and $\overline{m}_a^T$ is pseudo-moment-consistent on $\{G(\overline{\mu}, \overline{m}_a, i, j) : G \in \mathcal{G}, i, j \in [m], a \in \{2, \ldots, k\}\}$, Lemma 26 tells us that $(\overline{\mu}^T, \overline{m}_a^T)$ must be $(\alpha', a\alpha + \beta', \frac{a}{m})$-mean-conditioned-moment multicalibrated. By union bounding over each $a \in \{2, \ldots, k\}$ the total failure probability is $1 - 3\delta|\mathcal{G}|(km^2 + m)$. □

Corollary 4. *Fix target parameters α', β', δ' and $\epsilon > 0$ such that $\epsilon < \alpha'$ and $\epsilon < \beta'$. Define*

$$\overline{Q} = \frac{6|\mathcal{G}|km^2}{\left(\frac{\alpha' - \epsilon}{6 + \frac{2}{\epsilon^2}}\right)^2 \left(\frac{\beta' - \epsilon}{6 + \frac{2}{\epsilon^2}}\right)^2}, \quad \delta = \frac{\delta'}{\max(3|\mathcal{G}|(km^2 + m), \overline{Q})},$$

$$n_\alpha = \frac{\ln(\frac{2\overline{Q}}{\delta})}{2\left(\frac{\alpha' - \epsilon}{6 + \frac{2}{\epsilon^2}}\right)^2}, \quad n_\beta = \frac{\ln(\frac{2\overline{Q}}{\delta})}{2\left(\frac{\beta' - \epsilon}{6 + \frac{2}{\epsilon^2}}\right)^2}.$$

Then, AlternatingGradientDescentFinite$(\alpha, \beta, \delta, n, \mathcal{G})$ *where*

$$\alpha = 2\sqrt{\frac{\ln(\frac{2\overline{Q}}{\delta})}{2n_\alpha}} + \epsilon, \beta = 2\sqrt{\frac{\ln(\frac{2\overline{Q}}{\delta})}{2n_\beta}} + \epsilon,$$

$$n = \max\left(\frac{\ln(\frac{2\overline{Q}}{\delta})}{\ln(\frac{2}{\delta})}n_\alpha, \frac{\ln(\frac{2\overline{Q}}{\delta})}{\ln(\frac{2}{\delta})}n_\beta, \frac{2\ln(\frac{2}{\delta})}{\alpha^2}, \frac{2\ln(\frac{2}{\delta})}{\beta^2}\right)$$

has the following guarantees with probability $1 - \delta'$*:*

1. *The total number of gradient descent updates will be at most* Q*, where* Q *is as defined in Theorem 22.*

2. $\overline{\mu}^T$ *is* α'*-mean-multicalibrated.*

3. *For every* $a \in \{2, \dots, k\}$*,* $(\overline{\mu}^T, \overline{m}_a^T)$ *is* $(\alpha', a\alpha' + \beta', \frac{a}{m})$*-mean-conditioned-moment multicalibrated.*

Proof. Note that by construction, we have

$$\alpha > 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}} \quad \text{and} \quad \beta > 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}.$$

Therefore, in Theorem 22, the level of mean calibration for $\overline{\mu}^T$ will be

$$\begin{aligned}
&\alpha + 4\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}} + \left(\alpha - 2\sqrt{\frac{\ln\left(\frac{2}{\delta}\right)}{2n}}\right)^{-2} \left(2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}\right) \\
&\leq \alpha + 4\sqrt{\frac{\ln\left(\frac{2\overline{Q}}{\delta}\right)}{2n_\alpha}} + \left(\alpha - 2\sqrt{\frac{\ln\left(\frac{2\overline{Q}}{\delta}\right)}{2n_\alpha}}\right)^{-2} \left(2\sqrt{\frac{\ln(\frac{2\overline{Q}}{\delta})}{2n_\alpha}}\right) \\
&= 2\sqrt{\frac{\ln(\frac{2\overline{Q}}{\delta})}{2n_\alpha}} + \epsilon + 4\sqrt{\frac{\ln\left(\frac{2\overline{Q}}{\delta}\right)}{2n_\alpha}} + \frac{\left(2\sqrt{\frac{\ln(\frac{2\overline{Q}}{\delta})}{2n_\alpha}}\right)}{\epsilon^2} \\
&= \sqrt{\frac{\ln(\frac{2\overline{Q}}{\delta})}{2n_\alpha}} \left(6 + \frac{2}{\epsilon^2}\right) + \epsilon \\
&= \alpha',
\end{aligned}$$

where the first inequality follows because $n \geq \frac{\ln(\frac{2\overline{Q}}{\delta})}{\ln(\frac{2}{\delta})} n_\alpha$ and the last equality from the definition of n_α.

Applying the same analysis, we can show that we satisfy pseudo-moment-consistency at level β'. Therefore, for any $a \in \{2, \ldots, k\}$, $(\overline{\mu}^t, \overline{m}_a^T)$ satisfy $(\alpha', a\alpha' + \beta', \frac{a}{m})$-mean-conditioned-moment multicalibration.

The failure probabilities for mean muticalibration and that of mean-conditioned-moment multicalibration are both less than δ', as $3\delta(m^2 + m)|\mathcal{G}| \leq 3\delta|\mathcal{G}|(km^2 + m) \leq \delta'$ and $3\delta|\mathcal{G}|(km^2 + m) \leq \delta'$.

The failure probability for termination is

$$\begin{aligned}
&3\delta|\mathcal{G}|\left(\frac{1}{\alpha^2}-1\right)\left((m^2+m)+m^2(k-1)\left(\frac{1}{\beta^2}-1\right)\right)\\
&\leq 3\delta|\mathcal{G}|\left(\frac{1}{\alpha^2}\right)\frac{2km^2}{\beta^2}\\
&= 6|\mathcal{G}|km^2\delta\cdot\frac{1}{\left(\sqrt{\frac{\ln(\frac{2\overline{Q}}{\delta})}{2n_\alpha}}+\epsilon\right)^2\left(\sqrt{\frac{\ln(\frac{2\overline{Q}}{\delta})}{2n_\beta}}+\epsilon\right)^2}\\
&\leq 6|\mathcal{G}|km^2\delta\cdot\frac{1}{\frac{\ln(\frac{2\overline{Q}}{\delta})}{2n_\alpha}\frac{\ln(\frac{2\overline{Q}}{\delta})}{2n_\beta}}\\
&\leq 24|\mathcal{G}|km^2\delta\cdot\frac{1}{\left(\ln(\frac{2\overline{Q}}{\delta})\right)^2}n_\alpha n_\beta\\
&= 24|\mathcal{G}|km^2\delta\cdot\frac{1}{\left(\ln(\frac{2\overline{Q}}{\delta})\right)^2}\frac{\ln(\frac{2\overline{Q}}{\delta})}{2\left(\frac{\alpha'-\epsilon}{6+\frac{2}{\epsilon^2}}\right)^2}\frac{\ln(\frac{2\overline{Q}}{\delta})}{2\left(\frac{\beta'-\epsilon}{6+\frac{2}{\epsilon^2}}\right)^2}\\
&\leq 6|\mathcal{G}|km^2\delta\cdot\frac{1}{\left(\frac{\alpha'-\epsilon}{6+\frac{2}{\epsilon^2}}\right)^2\left(\frac{\beta'-\epsilon}{6+\frac{2}{\epsilon^2}}\right)^2}\\
&= \overline{Q}\delta\\
&\leq \delta'
\end{aligned}$$

□

Theorem 23. *With probability* $1-3\delta|\mathcal{G}|Q_\alpha\left((m^2+m)+m^2Q_\beta\right)$*, the running time of Algorithm 11 is* $O\left(Q|\mathcal{G}|m^2n\right)=O\left(\frac{k|\mathcal{G}|m^2n}{\alpha^2\beta^2}\right)$ *where* Q_α, Q_β*, and* Q *are as defined in Theorem 22.*

Proof. Theorem 22 tells us that except with probability $1-3\delta|\mathcal{G}|Q_\alpha\left((m^2+m)+m^2Q_\beta\right)$, the algorithm will halt after at most Q many gradient descent updates. For each gradient descent update, it must have been that Algorithm 9 was invoked with either $\mathcal{S}=\{G(\overline{\mu},i):G\in\mathcal{G},i\in[m]\}\cup\{G(\overline{\mu},\overline{m}_a,i,j):G\in\mathcal{G},i,j\in[m],a\in\{2,\ldots,k\}\}$ or $\mathcal{S}=\{G(\overline{\mu},\overline{m}_a,i,j):G\in\mathcal{G},i,j\in[m],a\in\{2,\ldots,k\}\}$. Note that Algorithm 9 needs to iterate through each set

S in $\mathcal{S}$, whose size is at most $O(|\mathcal{G}|m^2)$ in either case. And processing each set S through Algorithm 8 requires finding the average of at most $O(n)$ elements twice. Therefore, the algorithm will take time $O\left(Q|\mathcal{G}|m^2n\right)$ with probability $1-3\delta|\mathcal{G}|Q_\alpha\left((m^2+m)+m^2Q_\beta\right)Q$.

□

```
Algorithm 14: PseudoMomentConsistencyWithOracle($a, \beta, \delta, \overline{\mu}, \overline{m}_a, n, \mathcal{G}$)
    $D = \{(x_b, y_b)\}_{b=1}^n \sim \mathcal{P}^n$
    $D^{\text{check}} \sim \mathcal{P}^n$
    $\mathcal{S} = \{\mathcal{X}(\overline{\mu}, \overline{m}_a, i, j) : i, j \in [m]\}$
    $\overline{\ell}(x) = \overline{m}_a(x)$
    $\ell(x, y) = (y - \overline{\mu}(x))^a$
    $R, \psi = $ LearningOracleConsistencyAuditorWrapper($\overline{\ell}, \ell, \beta, \delta, D, D^{\text{check}}, \mathcal{S}, A$)
    while $R, \psi \neq NULL$ do
        $\overline{m}_a(x) = \begin{cases} \text{project}_{[0,1]}(\overline{m}_a(x) - \beta\psi) & \text{if } x \in R \\ \overline{m}_a(x) & \text{otherwise.} \end{cases}$
        $D = \{(x_b, y_b)\}_{b=1}^n \sim \mathcal{P}^n$
        $\mathcal{S} = \{\mathcal{X}(\overline{\mu}, \overline{m}_a, i, j) : i, j \in [m]\}$
        $\overline{\ell}(x) = \overline{m}_a(x)$
        $D = \{(x_b, y_b)\}_{b=1}^n \sim \mathcal{P}^n$
        $D^{\text{check}} \sim \mathcal{P}^n$
        $R, \psi = $ LearningOracleConsistencyAuditorWrapper($\overline{\ell}, \ell, \beta, \delta, D, D^{\text{check}}, \mathcal{S}, A$)
    end
    return $\overline{m}_k$
```

Lemma 30. *For each $R \in \mathbb{R}$ and any χ_S:*

$$\mathop{\mathbb{E}}_{(x,y)}\left[\chi_S(x) \cdot r_R^+(x, y)\right] = \mathcal{P}_\mathcal{X}(R \cap S)\left(\overline{\ell}(R \cap S) - \ell(R \cap S)\right)$$

$$\mathop{\mathbb{E}}_{(x,y)}\left[\chi_S(x) \cdot r_R^-(x, y)\right] = \mathcal{P}_\mathcal{X}(R \cap S)\left(\ell(R \cap S) - \overline{\ell}(R \cap S)\right)$$

Algorithm 15: AlternatingGradientDescentWithOracle($\alpha, \beta, \delta, n, \mathcal{G}, A$)

```
Initialize $\overline{\mu}^1(x) = 0$ for all $x$
For all $1 < a \leq k$, initialize $\overline{m}_a^1(x) = 0$ for all $x$
$t = 1$
$\overline{\ell}^t(x) = \overline{\mu}(x)$
$\ell(x, y) = y$
$D^t =\sim \mathcal{P}^n$
$D^{\text{check}^t} \sim \mathcal{P}^n$
$\mathcal{S}^t = \{\mathcal{X}(\overline{\mu}^t, i) : i \in [m]\} \cup \{\mathcal{X}(\overline{\mu}^t, \overline{m}_a^t, i, j) : i, j \in [m], a \in \{2, \ldots, k\}\}$
$S^t, \lambda^t =$ LearningOracleConsistencyAuditorWrapper($\overline{\ell}^t, \ell, \beta, \delta, D^t, D^{\text{check}^t}, \mathcal{S}^t, A$)
while $S^t, \lambda^t \neq NULL$ do
    $\overline{\mu}^{t+1} =$ MeanConsistencyUpdate($\overline{\mu}^t, S^t, \lambda^t$)
    for $a = 2, \ldots, k$ do
        $\overline{m}_a^{t+1} =$ PseudoMomentConsistencyFinite($a, \beta, \delta, \overline{\mu}^{t+1}, \overline{m}_a^t, n, \mathcal{G}$).
    end
    $t = t + 1$
    $\overline{\ell}^t(x) = \overline{\mu}(x)$
    $D^t \sim \mathcal{P}^n$
    $D^{\text{check}^t} \sim \mathcal{P}^n$
    $\mathcal{S}^t = \{\mathcal{X}(\overline{\mu}^t, i) : i \in [m]\} \cup \{\mathcal{X}(\overline{\mu}^t, \overline{m}_a^t, i, j) : i, j \in [m], a \in \{2, \ldots, k\}\}$
    $S^t, \lambda^t =$ LearningOracleConsistencyAuditorWrapper($\overline{\ell}^t, \ell, \beta, \delta, D^t, D^{\text{check}^t}, \mathcal{S}^t, A$)
end
return $(\overline{\mu}^t, \{\overline{m}_a^t\}_{a=2}^k)$
```

Proof.

$$\begin{aligned}\mathop{\mathbb{E}}_{(x,y)}[\chi_S(x)\cdot r_R^+(x,y)] &= \sum_{x,y}\mathcal{P}(x,y)\chi_S(x) r_R^+(x,y)\\ &= \sum_{x,y:x\in S,x\in R}\mathcal{P}(x,y)(\bar{\ell}(x)-\ell(x,y))\\ &= \mathcal{P}_{\mathcal{X}}(R\cap S)\left(\bar{\ell}(R\cap S)-\ell(R\cap S)\right)\end{aligned}$$

The same argument applies for r_R^- as well. □

Theorem 24. *Assume n is sufficiently large such that $\alpha > 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}$. Algorithm 13 has the following guarantees:*

1. *If it returns some S and λ, then with probability $1-3\delta|\mathbb{R}|$ over the randomness of D^{check},*

$$|\bar{\ell}(S)-\ell(S)| \geq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S)}.$$

2. *If it returns $NULL$, then with probability $1-|\mathbb{R}|(3\delta+2p(n))$ over the randomness of D and D^{check}, for all $\chi_S\in\mathcal{H}$ and $R\in\mathbb{R}$,*

$$|\bar{\ell}(R\cap S)-\ell(R\cap S)| \leq \frac{\alpha'+\rho}{\mathcal{P}_{\mathcal{X}}(R\cap S)},$$

   where α' is as defined in Corollary 3.

Proof. With probability at least $1-3|\mathbb{R}|\delta$ (since $|\mathbb{R}|\geq|\mathcal{V}|$), Corollary 3 gives us the following guarantees:

1. If S,λ is returned, then

$$\left|\bar{\ell}(S)-\ell(S)\right| \geq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(S)} \quad \text{and} \quad \lambda = \text{sign}(\bar{\ell}(S)-\ell(S)).$$

2. If $NULL$ is returned, then for all $V \in \mathcal{V}$,

$$|\bar{\ell}(V) - \ell(V)| \leq \frac{\alpha'}{\mathcal{P}_{\mathcal{X}}(V)} \tag{5.4}$$

Whenever the distributional closeness conditions of Definition 19 hold (which occur with the same $1 - 3|\mathbb{R}|\delta$ success probability of Corollary 3), and $\alpha > 2\sqrt{\frac{\ln(\frac{2}{\delta})}{2n}}$, it must be that if $|D_R| = 0$ then $\mathcal{P}_{\mathcal{X}}(R) \leq \alpha$. And for any such R we have that $\mathcal{P}_{\mathcal{X}}(R \cap S') \leq \alpha$ for any other set S', which implies that we trivially satisfy $(\alpha' + \rho)$-mean consistency for $R \cap S'$. More precisely, if $\mathcal{P}_{\mathcal{X}}(R) \leq \alpha$, then

$$\sup_{\chi_{S'} \in \mathcal{H}} |\bar{\ell}(R \cap S') - \ell(R \cap S')| \leq 1 \leq \frac{\alpha}{\mathcal{P}_{\mathcal{X}}(R \cap S')} \leq \frac{\alpha' + \rho}{\mathcal{P}_{\mathcal{X}}(R \cap S')}.$$

We can therefore restrict our attention to those $R \in \mathbb{R}$ that satisfy $|D_R| > 0$, since we only have a non-trivial statement to prove for sets R with $\mathcal{P}_{\mathcal{X}}(R) > \alpha$. Using Lemma 30 and the definition of an agnostic learning oracle, we know that for each $V = R \cap S^+$, with probability $1 - p(n)$,

$$\begin{aligned}
&\mathcal{P}_{\mathcal{X}}(R \cap S^+) \left(\bar{\ell}(R \cap S^+) - \ell(R \cap S^+)\right) + \rho \\
&= \mathop{\mathbb{E}}_{(x,y)} [\chi_{S^+}(x) \cdot r_R^+(x, y)] + \rho \\
&\geq \sup_{\chi_{S'} \in \mathcal{H}} \mathop{\mathbb{E}}_{(x,y)} [\chi_{S'}(x) \cdot r_R^+(x, y)] \\
&= \sup_{\chi_{S'} \in \mathcal{H}} \mathcal{P}_{\mathcal{X}}(R \cap S') \left(\bar{\ell}(R \cap S') - \ell(R \cap S')\right)
\end{aligned} \tag{5.5}$$

The same argument applies for $V = R \cap S^-$, and we obtain

$$\mathcal{P}_{\mathcal{X}}(R \cap S^+) \left(\ell(R \cap S^-) - \bar{\ell}(R \cap S^-)\right) + \rho \geq \sup_{\chi_{S'} \in \mathcal{H}} \mathcal{P}_{\mathcal{X}}(R \cap S') \left(\ell(R \cap S') - \bar{\ell}(R \cap S')\right). \tag{5.6}$$

Combining (5.4), (5.5), and (5.6), we get that with probability $1 - 2|\mathbb{R}|p(n)$,

$$\sup_{\chi_{S'} \in \mathcal{H}} |\bar{\ell}(R \cap S') - \ell(R \cap S')| \le \frac{\alpha' + \rho}{\mathcal{P}_{\mathcal{X}}(R \cap S')}$$

□

Theorem 26. *With probability at least* $1 - 3\delta Q_\alpha \left((m^2 + m) + m^2 Q_\beta\right)$*, the running time of Algorithm 15 is bounded by* $O(Qm^2\tau(n))$*, where* Q *is the total number of gradient descent operations as defined in Theorem 22.*

Proof. The running time of Algorithm 13 is $O(m^2\tau(n))$, as we always call it with $|\mathbb{R}| = O(m^2)$ and we assumed $\tau(n) = \Omega(n)$, meaning the running time of the learning oracle dominates the calculations in the empirical check. And Theorem 25 gives that with probability $1 - 3\delta Q_\alpha \left((m^2 + m) + m^2 Q_\beta\right)$, there will be at most Q gradient descent operations. Because the number of gradient descent operations is equal to the number of subroutine calls to Algorithm 13, the overall running time is $O(Qm^2\tau(n))$. □

5.D. A Submodular Set-Cover Formulation

We can define the following problem. Theorem 27 shows us that for every even a, and every $G \in \mathcal{G}$, $i, j \in [m]$, $I_{\gamma,a}(x)$ forms a valid marginal prediction interval for every set $G(\overline{\mu}, \overline{m}_a, i, j)$ with probability at least γ under $\mathcal{P}_{\mathcal{X}}$. Can we construct tighter prediction intervals using all $\lfloor \frac{k}{2} \rfloor$ moments?

We make the following simplifying assumptions in this section:

1. $\mathcal{X}$ is a set with finite cardinality.
2. $\mathcal{P}_{\mathcal{X}}$ is known exactly (note that we do not assume we know the distribution on *labels* y, which preserves the core motivation of the problem).
3. For every $x \in \mathcal{X}$, there exists $G \in \mathcal{G}$, a even s.t. $1 < a \le k$, and $i, j \in [m]$ such that

$x \in G(\overline{\mu}, \overline{m}_a, i, j)$ and $\mathcal{P}_{\mathcal{X}}(G(\overline{\mu}, \overline{m}_a, i, j)) \geq \gamma$ (otherwise there is no way to give a valid marginal prediction interval for such an x).

Let us define the set of all relevant sets as

$$\mathcal{S} \equiv \{G(\overline{\mu}, \overline{m}_a, i, j) : \forall G \in \mathcal{G}, i, j \in [m], 1 < a \leq k, a \text{ even s.t. } \mathcal{P}_{\mathcal{X}}(G(\overline{\mu}, \overline{m}_a, i, j)) \geq \delta\}.$$

With each set $S \in \mathcal{S}$, we associate the width $\Delta_S(\cdot)$ in the obvious way.

Given any $\mathcal{S}' \subseteq \mathcal{S}$ we say that $\mathcal{S}'$ covers $\mathcal{X}$ if $\forall x \in \mathcal{X}$, $\exists S \in \mathcal{S}'$ s.t. $x \in S$. Given any $\mathcal{S}' \subseteq \mathcal{S}$ that covers $\mathcal{X}$, we can construct valid marginal prediction intervals for all $x \in X$:

$$\Delta_{\mathcal{S}'}(x) \equiv \max_{S \in \mathcal{S}' | x \in S} \Delta_S(x),$$
$$I_{\mathcal{S}'}(x) = [\overline{\mu}(x) - \Delta_{\mathcal{S}'}(x), \overline{\mu}(x) + \Delta_{\mathcal{S}'}(x)].$$

To see that this will result in a valid prediction interval, observe that for any $x \in X$, it is covered by some $S \in \mathcal{S}'$. By definition of $\mathcal{S}'$, $S = G(\overline{\mu}, \overline{m}_a, i, j)$ for some a even, $i, j \in [m]$, $G \in \mathcal{G}$. Note that $I_{\gamma,a}(x) \subseteq I_{\mathcal{S}'}(x)$ by construction of $I_{\mathcal{S}'}(\cdot)$. Therefore Theorem 27 ensures that these prediction intervals are valid for any $S \in \mathcal{S}'$, and indeed, therefore valid for any group $G \in \mathcal{G}$.

A natural optimization problem is to find a subset $\mathcal{S}'$ that covers $\mathcal{X}$ so as to minimize the expected width of the marginal prediction intervals that can be produced in this way, i.e. solves (exactly or approximately)

$$\min_{\mathcal{S}' \subseteq \mathcal{S}} \mathop{\mathbb{E}}_{x \sim \mathcal{P}_{\mathcal{X}}} [\Delta_{\mathcal{S}'}(x)]$$
$$\text{s.t. } \mathcal{S}' \text{ covers } \mathcal{X}.$$

We can rewrite the problem in the following way. Let A be a $0-1$ matrix of dimension

$|\mathcal{X}| \times |\mathcal{S}|$. The columns correspond to sets $S \in \mathcal{S}$ and the rows to elements $x \in \mathcal{X}$. If $A_{xS} = 1$ this means that element $x \in \mathcal{X}$ is contained in set $S \in \mathcal{S}$. Associated with each column S there is a function Δ_S. Recall that $\mathcal{P}_{\mathcal{X}}(x)$ denotes the probability of x.

We can denote any subset of $\mathcal{S}' \subseteq \mathcal{S}$ by a 0/1 vector $w \in \{0,1\}^{|\mathcal{S}|}$ such that $w_S = 1$ if $S \in \mathcal{S}'$. We can therefore recast the optimization problem:

$$\begin{aligned} &\min_{z,w} \sum_{x \in \mathcal{X}} \mathcal{P}_{\mathcal{X}}(x) z_x \\ &\text{s.t. } z_x - \Delta_S(x) A_{xS} w_S \geq 0 && \forall x \in \mathcal{X} \\ &\qquad \sum_{S \in \mathcal{S}} A_{xS} w_S \geq 1 && \forall x \in \mathcal{X}. \end{aligned}$$

For any subset $\mathcal{S}' \subseteq \mathcal{S}$ let $f_x(\mathcal{S}') = \max_{S \in \mathcal{S}'} \Delta_S(x) A_{xS}$. Notice $f_x(\mathcal{S}')$ is a non-decreasing and submodular function of $\mathcal{S}'$. Let $f(\mathcal{S}') = \sum_x \mathcal{P}_{\mathcal{X}}(x) f_x(\mathcal{S}')$, clearly $f(\cdot)$ is non-decreasing and submodular. Similarly, for any $\mathcal{S}' \subseteq \mathcal{S}$ let w be the associated 0/1 vector and define $g(\mathcal{S}') = |\{x : \sum_{S \in \mathcal{S}} A_{xS} w_S \geq 1\}|$. Again $g(\cdot)$ is a non-decreasing and submodular function of $\mathcal{S}'$. Observe that can write our problem as:

$$\begin{aligned} &\min_{\mathcal{S}' \subseteq \mathcal{S}} f(\mathcal{S}'), \\ &\text{s.t. } g(\mathcal{S}') \geq |\mathcal{X}|. \end{aligned}$$

Therefore, our problem is the submodular cost submodular cover problem.

We can now hope to apply known results to solve it. For example, [94] show that the greedy solution (to iteratively add the set with the smallest average width) is approximately optimal. In particular, their Theorem 2.1 guarantees that the greedy solution provides a $\frac{k}{2}H$-approximate to our optimization problem where H is the ℓ^{th}-harmonic number, $\ell = \max\{|S| : S \in \mathcal{S}\}$, and k is the number of moments we have access to.

Unfortunately in this context, these guarantees are unsatisfactory: the approximation grows with the number of moments k we have access to, and with $\log |\mathcal{X}|$, which will typically be linear in the data dimension. Note that [94] study general submodular objective functions and does not exploit the specific structure of the objective function here. We leave the question of whether guarantees can be offered for this problem to future research. Another natural question is how to approximate this optimization when $\mathcal{P}_{\mathcal{X}}$ is not known, i.e. we only have a finite sample from $\mathcal{P}$.

Chapter 6

Uncertainty Estimation for Subgroups: Online

6.1. Introduction

Consider the problem of making predictions about the prognoses of patients with an infectious disease at the early stages of a pandemic. To be able to guide the allocation of medical interventions, we may want to predict, from each patient's observable features x, things such as the expected severity of the disease y in two days' time. And since we will be using these predictions to allocate scarce resources, we will want to be able to quantify the uncertainty of our predictions: perhaps by providing estimates of the variance of outcomes, or perhaps by providing prediction intervals at a desired level of confidence.

This is slightly different than the setting considered in the previous chapter, as it's an *online* problem. In the previous chapter, we showed how to calibrate moment estimates for a large number of groups and how to leverage Chebyshev's inequality on these multicalibrated moments in order to construct uncertainty estimates. However, there was an underlying assumption about a distribution and our access to this distribution via a set of samples drawn i.i.d. from the distribution. By contrast, in the above described problem, we must start making predictions before we have much data, and the predictions are needed immediately upon the arrival of a patient. It is also a problem in which the environment is rapidly changing: the distribution of patients changes as the disease spreads through different populations, and the conditional distribution on outcomes given features changes as we learn how to better treat the disease.

How can we approach this problem? The *conformal prediction* literature [86] aims to equip arbitrary regression and classification procedures for making point predictions with prediction intervals that contain the true label with (say) 95% probability. But for the application in our example, conformal prediction has two well-known shortcomings:

Marginal Guarantees: Conformal prediction only gives *marginal* prediction intervals: in other words, it provides guarantees that (e.g.) 95% of the prediction intervals produced over a sequence of predictions cover their labels. But these guarantees are averages over what are typically large, heterogeneous populations and therefore provide little guidance for making decisions about individuals. For example, it would be entirely consistent with the guarantee of a 95% marginal prediction interval $[\ell_t, u_t]$ if for individuals from some demographic group G making up less than 5% of the population, their labels y_t fall outside of $[\ell_t, u_t]$ 100% of the time.[20] One could run many parallel algorithms for different demographic groups G_i, but then there would be no clear way to interpret the many different predictions one would receive for an individual belonging to several demographic groups at once ($x \in G_i$ for multiple groups G_i); for example, prediction intervals corresponding to different demographic groups could be disjoint. To see that marginal guarantees on their own are extremely weak, consider a batch (distributional) setting in which labelled points are drawn from a fixed distribution $\mathcal{D}$: $(x, y) \sim \mathcal{D}$. Then we could provide valid 95% marginal prediction intervals by entirely ignoring the features and giving a fixed prediction interval of $[\ell, u]$ for every point, where $[\ell, u]$ is such that $\Pr_{(x,y)\sim\mathcal{D}}[y \notin [\ell, u]] = 0.05$.

Distributional Assumptions: The conformal prediction literature almost exclusively assumes that the data is drawn from an *exchangeable* distribution (for example, i.i.d. data satisfies this property) and does not offer any guarantees when the data can quickly change in unanticipated or adversarial ways.

In this chapter, we give techniques for dealing with both of these problems (and similar issues that arise for the problem of predicting label means and higher moments) by drawing on ideas from the literature on *calibration* [17, 29]. Calibration is similar to conformal prediction in that it aims to give point estimates in nonparametric settings that satisfy marginal rather than conditional guarantees (i.e. that agree with the true distribution as averaged over the data rather than conditioned on the features of a particular data point). But calibration

[20] Even more insidious reversals, albeit not in the context of conformal prediction, have been observed on real world data—see the Wikipedia entry for Simpson's paradox (https://en.wikipedia.org/wiki/Simpson%27s_paradox) for several examples.

is concerned with predicting label expectations, rather than giving prediction intervals. Informally speaking, calibrated predictions satisfy that when averaging over all rounds over which the prediction was (approximately) p, the realized labels average to (approximately) p, for all p. Note that in a distributional setting, if a learner truly was predicting the conditional label expectations conditional on features $p_x = \mathbb{E}_{(x,y)\sim\mathcal{D}}[y|x]$, then the forecasts would be calibrated — but just as with marginal prediction intervals, calibration on its own is a very weak condition in a distributional setting. For example, a learner could achieve calibration simply by making a single, constant prediction of $p = \mathbb{E}_{(x,y)\sim\mathcal{D}}[y]$ for every point, and so calibrated predictions need not convey much information. Thus, just like the conformal prediction literature, the calibration literature is primarily focused on the online prediction setting. But from early on, the calibration literature has focused on the *adversarial* setting in which no distributional assumptions need to be made at all [29, 33, 82].

As described in Chapter 5, we emphasize again that calibration also suffers from the weaknesses that come with marginal guarantees: namely that calibrated predictions may have little to do with the conditional label expectations for members of structured sub-populations. Hébert-Johnson et al. [43] proposed an elegant solution to this problem in the batch setting, when predicting expectations, which they termed "multicalibration". Informally speaking, a guarantee of multicalibration is parameterized by a large collection of potentially intersecting subsets of the feature space $\mathcal{G}$ (corresponding e.g. to demographic groups or other categories relevant for the prediction task at hand). Multicalibration asks for predictions that are not just calibrated over the full distribution $\mathcal{P}$ but are also simultaneously calibrated over all of the induced distributions that are obtained by conditioning on membership in a set $G \in \mathcal{G}$. Moreover, Hébert-Johnson et al. [43] showed how to obtain multicalibrated estimators in the batch, distributional setting with sample complexity that depends only logarithmically on $|\mathcal{G}|$. In Chapter 5, we showed how to extend the notion of (multi)calibration from expectations to variances and other higher moments — and derived algorithms for obtaining such estimates in the batch setting.

6.1.1. Our Results and Techniques

In this chapter, we give a general method for obtaining different kinds of "multivalid" predictions in an online, adversarial setting. This includes mean estimates that satisfy the notion of mean multicalibration from [43], moment estimates that satisfy the notion of mean-conditioned moment multicalibration from Chapter 5, and prediction intervals which satisfy a new notion of multivalidity, defined in this chapter. The latter asks for tight marginal prediction intervals, which are simultaneously valid over each demographic group $G \in \mathcal{G}$. We give a formal definition in Section 6.3 (and review the definitions of mean and moment multicalibration), but informally, multivalidity for prediction intervals asks, given a target coverage probability $1-\delta$, that for each group $G \in \mathcal{G}$ there be roughly a $1-\delta$-fraction of points (x_t, y_t) with $x_t \in G$ whose label is contained within the predicted interval ($y_t \in [\overline{\ell}_t, \overline{u}_t)$). In fact, we ask for the stronger calibration-like guarantee, that these marginal coverage guarantees hold even conditional on the prediction interval, which (among other things) rules out the trivial solution to marginal coverage that predicts the full interval with probability $1-\delta$ and an empty interval with probability δ. Because our algorithms handle adversarially selected examples, they can equally well be used to augment arbitrary point prediction procedures which give predictions $f_t(x_t) = \hat{y}_t$, independently of how they are trained: we can simply feed our algorithms for multivalid predictions with the *residuals* $\hat{y}_t - y_t$. For example, we can get variance estimates or prediction intervals for the residuals to endow the *predictions* of f_t with uncertainty estimates. Endowing point predictors with prediction intervals in this way provides an alternative to conformal prediction that gives stronger-than-marginal (multivalid) guarantees, under much weaker assumptions (adversarially chosen examples). In general, for each of our techniques, if we instantiate them with the trivial group structure (i.e. one group, containing all points), then we recover standard (or slightly stronger) marginal guarantees: i.e. simple calibrated predictions and simple marginal prediction intervals.[21] But as we enrich our collection of sets $\mathcal{G}$, our guarantees

[21] In fact, even with the trivial group structure, our guarantees (with appropriately set parameters) remain stronger than marginal coverage. This is because our prediction intervals remain valid even conditioning on the prediction that we made. For example, a prediction interval $[\ell, u)$ is valid not just as averaged over all rounds t, but also as averaged over all rounds t for which we made that specific prediction: $t : [\overline{\ell}_t, \overline{u}_t) = [\ell, u)$.

become correspondingly stronger.

The General Strategy We derive our online algorithms using a general strategy that dates back to Fudenberg and Levine [33], who used it to give online algorithms for the problem of simple calibration in a setting without features (see also the argument by Sergiu Hart, communicated in Foster and Vohra [29] and more recently elaborated on in Hart [42]). In our context, the general strategy proceeds as follows:

1. Define a surrogate loss function, such that if the surrogate loss is small at the end of T rounds, then the learner's predictions satisfy our chosen notion of multivalidity over the empirical distribution of the history of the interaction.

2. Argue that if at each round t, the adversary's chosen distribution over labelled examples were known to the learner, then there would be some prediction that the learner could make that would guarantee that the expected increase in the surrogate loss function at that round would be small. This step is often straightforward, because once we fix a known data distribution $\mathcal{D}$, "true distributional quantities" like conditional label expectations, conditional label variances, conditional label quantiles, etc, generally satisfy our corresponding multivalidity desideratum by design.

3. Appeal to the minimax theorem to conclude that there must therefore exist a randomized prediction strategy for the learner that guarantees that the expected increase in the surrogate loss function is small for *any* choice of the adversary.

On its own, carrying out this strategy for a particular notion of multivalidity proves the *existence* of an algorithm that can obtain the appropriate notion of multivalidity against an adversary; but turning it into an actual (and efficient) algorithm requires the ability to *compute* at each round the equilibrium strategy whose existence is shown in Step 3 above.

We instantiate this general strategy in Section 6.4 for the case of mean multicalibration, which also serves as a template for our derivation and analysis of algorithms for moment multicalibration in Section 6.5 and prediction interval multivalidity in Section 6.6. The framework of our analysis is the same in each case, but the details differ: to carry out Step 2, we must bound the value of a different game, and to carry out Step 3, we must solve for the equilibrium of a different game. In each case, we obtain efficient online algorithms for obtaining high probability α-approximate multivalidity bounds (of different flavors), with α scaling roughly as $\alpha \approx \sqrt{\log|\mathcal{G}|/T}$, over interactions of length T — but see Sections 6.4.2, 6.5.2, and 6.6.2 for exact theorem statements. In all cases, our algorithms have per-round runtime that is linear in $|\mathcal{G}|$, and polynomial in the other parameters of the problem. In fact, both our run-time and our convergence bounds can be improved if each individual appears in only a bounded number of groups. Our algorithms can at each step t be implemented in time linear in the number of groups $G \in \mathcal{G}$ that *contain the current example* x_t. This is linear in $|\mathcal{G}|$ in the worst case, but can be substantially smaller. Similarly, we show in Appendix 6.A that if each individual appears in at most d groups, then the $\log|\mathcal{G}|$ term in our convergence bounds can be replaced with $\log(d)$, which gives informative bounds even if $\mathcal{G}$ is infinitely large. Without assumptions of this sort, running time that is polynomial in $|\mathcal{G}|$ (rather than logarithmic in $|\mathcal{G}|$, as our convergence bounds are) is necessary in the worst case, even for mean multicalibration in the offline setting, as shown by Hébert-Johnson et al. [43].

Adapting the original approach of Fudenberg and Levine [33] runs into several obstacles, stemming from the fact that the *action space* of both the learner and the adversary and the *number of constraints* defining our calibration desideratum are both much larger in our setting. Consider the case of mean prediction — in which the goal is to obtain calibrated predictions. In the featureless setting studied by Fudenberg and Levine [33], the action space for the learner corresponds to a discretization of the real unit interval $[0, 1]$, and the action space of the adversary is binary. In our setting, in which data points are endowed with features from a large feature space $\mathcal{X}$, the learner's action space corresponds to the set of all *functions* mapping $\mathcal{X}$ to $[0, 1]$, and the adversary's action space corresponds to the set of all labelled examples $\mathcal{X} \times [0, 1]$. Similarly, for simple calibration, the number of constraints is equal to the chosen discretization granularity of the unit interval $[0, 1]$, whereas in our case, the number of constraints also grows linearly with $|\mathcal{G}|$, the number of groups over which we want to be able to promise guarantees.

Convergence Rates and Sample Complexity The surrogate loss function used by Fudenberg and Levine [33] bounds the ℓ_2 calibration error — i.e. the average squared violation of all of the constraints used to define calibration. Because all of the notions of multivalidity that we consider consist of a set of constraints of size scaling linearly with $|\mathcal{G}|$, if we were to attempt to bound the ℓ_2 violation of our multivalidity constraints, we would necessarily obtain convergence bounds that scale polynomially with $|\mathcal{G}|$. Instead, we use a different surrogate loss function — a sign-symmetrized version of an exponential soft-max — that can be used to bound the ℓ_∞ violation of our multivalidity constraints and allows us to obtain bounds that scale only logarithmically with $|\mathcal{G}|$. For moment multicalibration, we face the further complication of needing to define a potential function bounding a linear surrogate for what is ultimately a nonlinear measure of distributional fidelity. An outline of the specific ideas needed here adapting the techniques from Chapter 5 can be found in Section 6.5.1. For interval multivalidity, we face the further complication that tight prediction intervals need not exist even in the distributional setting, for worst-case distributions. An outline of the new ideas we need to overcome this can be found in Section 6.6.1. Finally, we note that ℓ_∞ violation is consistent with how the existing literature on batch multicalibration [43] has quantified approximation guarantees. In fact, by using standard online-to-offline reductions, we can derive new sample complexity bounds for mean and moment multicalibration for the *batch distributional* setting that improve on the sample complexity bounds given in Hébert-Johnson et al. [43] and Chapter 6 — see Appendix A of Jung et al. [53] which this chapter is based off of. This is because when applied to the batch setting, our online algorithms take only a single pass through the data and avoid issues related to adaptive data re-use that complicated previous algorithms in the batch setting.

Computation of Equilibrium Strategies To compute equilibria of the large action space games we define, we do not attempt to directly compute or represent the function that we use at each round t to map features to labels. Instead, we represent this function implicitly by "lazily" solving a smaller equilibrium computation problem only after we have observed the adversary's choice of feature vector x (but before we have observed the label y) to compute a distribution over predictions. We show in each of our three settings that this computation is tractable. In the case of mean multicalibration, we are able to analytically derive a simple algorithm for sampling from this equilibrium strategy, presented in Section 6.4.3. For mean-conditioned k^{th} moment multicalibration we show that the equilibrium can be found using a linear program with polynomially many variables and $2^k + 1$ constraints. For the most interesting cases, k is a small constant (e.g. for variance, $k = 2$, and so the linear program has only 5 constraints). Even when k is large, we show that this linear program has a separation oracle that runs in time $O(k)$, and so it can be solved efficiently via the Ellipsoid algorithm. We show in Appendix 6.B that there always exists an equilibrium for the learner with support over at most $k + 1$ many predictions, limiting the extent to which it needs to deploy randomization. Finally, for prediction interval multivalidity, we show in Section 6.6.3 that we can express the equilibrium computation problem as a linear program. Although the linear program is naively defined by infinitely many constraints, we show that it can ultimately be represented with only finitely many constraints, and that it has an efficient separation oracle, so can be solved in polynomial time using the Ellipsoid algorithm.

Advantages of Conformal Prediction We have thus far emphasized the advantages that our techniques have over conformal prediction — but we also want to highlight the strengths of conformal prediction relative to our work, and directions for future improvement. Conformal prediction aims to obtain marginal coverage with respect to some (unknown) underlying distribution. As a result of the distributional assumption, it is able to obtain coverage (over the randomness of the distribution) at a rate of coverage $1-\delta+O(1/T)$ [69]. In contrast, in our setting, there is no underlying distribution. We therefore give guarantees on *empirical coverage* — i.e the fraction of labels that our predicted intervals have covered in the realized sequence of examples. As a result, our coverage bounds necessarily have error terms that tend to 0 at a rate of $O(1/\sqrt{T})$, over sequences of length T. We note that conformal prediction methods also obtain *empirical* coverage on the order of $1-\delta \pm O(1/\sqrt{T})$, as our methods do [69]. Conformal prediction methods naturally give one sided coverage error on the distribution (i.e. the coverage probability is always $\geq 1-\delta$), whereas as we present our bounds, our empirical coverage has two sided error. Techniques from the conformal prediction literature also can be applied to very general label domains $\mathcal{Y}$ and can be used to produce very general kinds of prediction *sets*. In this chapter, we restrict attention to real-valued labels $\mathcal{Y} = [0, 1]$ and prediction *intervals*. We do not believe that there are any fundamental obstacles to generalizing our techniques to other label domains and prediction sets, and this is an interesting direction for future work. Finally, the conformal prediction literature has developed a number of very simple, practical techniques. In this chapter, we give polynomial time algorithms, of varying complexity. Our algorithm for mean multicalibration in Section 6.4 is very simple to implement, but our algorithm for multivalid interval prediction in Section 6.6 requires solving a linear program with a separation oracle. Another important direction for future work is reducing the complexity of our techniques, and doing empirical evaluations.

6.2. Related Work

Work on calibrated mean prediction dates back to Dawid [17]. Foster and Vohra [29] were the first to show that in the online setting without features, it is possible to obtain asymptotic calibration even against an adversary. Once this initial result was proven, a number of proofs of it were given using different techniques, including Blackwell's approachability theorem [25] and a non-constructive minimax argument (originally communicated verbally by Sergiu Hart, appearing first in [29], and more recently formalized in [42]). This argument was "non-constructive" because it was a minimax argument over the entire algorithm design space. [33] gave a more tractable per-round minimax argument, which we adapt to our work — although they were satisfied with an existential argument, and do not derive a concrete algorithm. The algorithm we give for online multicalibration is similar to the algorithm given by Foster and Hart [27] for the simple calibration problem in the special case of a featureless setting and the trivial group structure. Lehrer [68] and Sandroni et al. [82] (and in a slightly different context, Fudenberg and Levine [34]) generalized this literature and showed that it was possible to extend these ideas in order to satisfy dramatically more demanding notions of calibration (e.g. calibration on all computable subsequences of rounds). This line of work primarily gives limit results via non-constructive arguments without establishing rates. There are two notable exceptions. Foster et al. [30] give a non-constructive argument establishing that it is possible to obtain mean calibration loss $\tilde{O}(\sqrt{\frac{\log K}{T}})$ with respect to a set of K "checking rules" which define subsequences over which the algorithm must be calibrated. These results are derived in a setting without features x, but we believe their techniques could be used to establish the same convergence bounds that we do, for mean multicalibration: $\alpha = \tilde{O}(\sqrt{\frac{\log |\mathcal{G}|}{T}})$. Foster and Kakade [28] give an efficient algorithm based on ridge-regression which can be used to achieve what we call *mean consistency*[22] on a collection of sets $\mathcal{G}$ with error rates converging as $\alpha = \tilde{O}(\sqrt{\frac{|\mathcal{G}|}{T}})$. Their algorithm is deterministic, which in particular means it cannot be used to achieve the standard notion of calibration, which can only be achieved by randomized algorithms in adversarial environments [76]. It can be used to achieve what is called "weak calibration" by Kakade and Foster [55] and "smooth calibration" by Foster and Hart [26] — a relaxation that can be obtained by deterministic algorithms. In comparison, our algorithm for mean multicalibration achieves the standard notion of calibration with the optimal sample complexity dependence on $\log |\mathcal{G}|$, while simultaneously being explicitly defined and computationally efficient.

There has also been a recent resurgence of interest in calibration in the computer science community, in part motivated by fairness concerns [13, 63, 79]. It is from this literature that the original proposal for multicalibration arose [43], as well as the related notion of multiaccuracy [43, 61]. Shabat et al. [85] prove uniform convergence bounds for multicalibrated predictors, Dwork et al. [21] draw connections between multicalibrated predictors and notions of fair rankings, and Dwork et al. [22] define a notion of outcome indistinguishability related to distribution testing, and show close connections to multicalibration. Jung et al.

[22] This is also what is known as *multi-accuracy* in Hébert-Johnson et al. [43] and Kim et al. [61].

[53], which Chapter 5 is based off of, extend the notion of mean calibration to variances and higher moments, and give efficient algorithms for learning moment multicalibrated predictors. Recall that in Chapter 5, we showed show that moment predictors can be used to derive *conservative* multivalid prediction intervals, using Chebyshev's inequality and generalizations to higher moments. In general, however, these moment-based inequalities give intervals that may cover their label much more frequently than the target $1 - \delta$ coverage probability and cannot achieve the kinds of tight multicoverage guarantees that we obtain in this chapter.

All of this work operates in the batch, distributional setting. Recently, Qiao and Valiant [80] proved lower bounds for simple mean calibration in the online setting, showing that no algorithm can obtain rates better than $O(T^{-0.472})$ against an adversary. At first blush, our upper bounds appear to contradict these lower bounds — but they do not, because we study convergence in the ℓ_∞ sense, whereas they study it in the ℓ_1 sense.

Conformal prediction is motivated similarly to calibration, but aims to produce marginal prediction intervals rather than mean estimates — see Shafer and Vovk [86] for an overview. The problems that we highlight — namely, that marginal guarantees are weak, and that this literature relies on strong distributional assumptions — have been noted before. For example, Barber et al. [6] prove that even in the distributional setting, *conditional* prediction intervals are impossible to provide, and aim instead for a goal that is similar to ours: providing marginal prediction intervals that are valid as averaged over a large number of subgroups $\mathcal{G}$. They take a conservative approach, by using a holdout set to estimate empirical prediction intervals separately for each group, and then taking the union of all of these prediction intervals over the demographic groups of a new individual. The result is that their prediction intervals — unlike ours — do not become tight, even in the limit. Chernozhukov et al. [12] consider the problem of conformal prediction for time series data, for which the exchangeability assumption may not hold. They show that if the data comes from a rapidly mixing process (so that, in particular, points that are well separated in the sequence are approximately independent) then it is still possible to obtain approximate marginal coverage guarantees. Tibshirani et al. [91] consider the problem of conformal prediction under *covariate shift*, in which the marginal distribution on features $\mathcal{X}$ differs between the training and test distributions, but the conditional distribution on labels $\mathcal{Y}|\mathcal{X}$ remains the same. They show how to adapt techniques from conformal prediction when the likelihood ratio between the training and test distribution is known. In the distributional setting, Gupta et al. [36] have proven close relationships between calibration, confidence intervals, and prediction intervals.

Finally, the notion of multicalibration is related to subgroup fairness notions [57, 59, 60] that ask for statistical "fairness" constraints of various sorts (beyond calibration) to hold across all subgroups defined by some rich class $\mathcal{G}$. See Chouldechova and Roth [14] for a survey.

6.3. Preliminaries

6.3.1. Notation

We write $\mathcal{X}$ to denote a feature domain and $\mathcal{Y} = [0, 1]$ to denote a label domain. We write $\mathcal{G} \subseteq 2^{\mathcal{X}}$ to denote a collection of subsets of $\mathcal{X}$. Given any $x \in \mathcal{X}$, we write $\mathcal{G}(x)$ for the set of groups that contain x, i.e. $\mathcal{G}(x) = \{G \in \mathcal{G} : x \in G\}$. Given an integer T we write $[T]$ to denote the set of integers $[T] = \{1, \ldots, T\}$. In general, we denote our random variables with tildes (e.g. $\tilde{X}$, $\tilde{Y}$) to distinguish them from their realizations (denoted e.g. X, Y). Given a finite set A, we write ΔA for the probability simplex over the elements in A.

6.3.2. Online Prediction

Online (contextual) prediction proceeds in rounds that we index by $t \in [T]$, for a given finite horizon T. In each round, an interaction between a *learner* and an *adversary* proceeds as follows. In each round t:

1. The *adversary* chooses a joint distribution over feature vectors $x_t \in \mathcal{X}$ and labels $y_t \in \mathcal{Y}$. The learner receives x_t (a realized feature vector), but no information about y_t is revealed.

2. The *learner* chooses a distribution over predictions $p_t \in \mathcal{P}$. (We will consider several different kinds of predictions in this chapter, and so are agnostic to the domain of the prediction for now — we use $\mathcal{P}$ as a generic domain).

3. The learner observes y_t (a realized label).

For an index $s \in [T]$, we denote by π_s the *transcript* of the interaction in rounds $t = 1$ through s: $\pi_s = ((x_t, p_t, y_t))_{t=1}^s$. We write Π^* as the domain of all transcripts.

Formally, the adversary is modelled as a probabilistic mapping $\text{Adv} : \Pi^* \to \Delta(\mathcal{X} \times \mathcal{Y})$ from transcripts to distributions over labelled data points, and the learner is modeled as a mapping $\text{Learn} : \Pi^* \to (\mathcal{X} \to \Delta\mathcal{P})$ from transcripts to a probabilistic mapping from feature vectors to distributions over predictions. An adversary may be either unconstrained (free to play any point in $\Delta(\mathcal{X} \times \mathcal{Y})$) or constrained to choose from some specified subset of $\Delta(\mathcal{X} \times \mathcal{Y})$. Fixing both a learner and an adversary induces a probability distribution over transcripts. Our goal is to derive particular learning algorithms, and to prove that various kinds of bounds hold either in expectation, or with high probability over the randomness of the transcript, in the worst case over transcript distributions, where we quantify over all possible adversaries.

Given a transcript π_T, a group $G \in \mathcal{G}$ and a set of rounds $S \subseteq [T]$, we write

$$G_S = \{t \in S : x_t \in G\}.$$

In words, this is the set of rounds in S in which the realized feature vectors in the transcript belonged to G. When it is clear from context, we sometimes overload notation, and for a group $G \in \mathcal{G}$, and a period $s \le T$, write G_s to denote the set of data points (indexed by their rounds) in a transcript π_s that are members of the group G:

$$G_s = \{t \in [s] : x_t \in G\}.$$

Types of Predictions, and Notions of Validity

We consider three types of predictions in this paper: Mean predictions, pairs of mean and higher moment predictions (e.g. variance), and prediction intervals.

Mean Predictions For mean predictions, the prediction domain will be the unit interval: $\mathcal{P}_{\text{mean}} = [0,1]$. The learner will select $p_t \equiv \overline{\mu}_t \in \mathcal{P}_{\text{mean}}$ in each round t, with the goal of predicting the conditional label expectation $\mathbb{E}[y_t|x_t]$. For any subset of days $S \subseteq [T]$, we write

$$\mu(S) = \frac{1}{|S|}\sum_{t\in S} y_t, \quad \overline{\mu}(S) = \frac{1}{|S|}\sum_{t\in S}\overline{\mu}_t$$

to denote the true label population mean conditional on $t \in S$ and the average of our mean estimates over days $t \in S$, respectively. We will ask for our predictions to satisfy large numbers of *mean consistency* constraints: that the conditional label averages be (approximately) equal to conditional prediction averages over different sets S.

Definition 21 (Mean Consistency)**.** *Given a transcript π_T, we say that the mean predictions $\{\overline{\mu}_t\}_{t=1}^T$ are α-mean consistent on $S \subseteq [T]$, if*

$$|\mu(S) - \overline{\mu}(S)| \leq \alpha\frac{T}{|S|}.$$

Remark 8. *Note the scaling with both T and $|S|$. If $S = [T]$, then this condition simply asks for the true label mean and the average prediction to be within α of one another, as averaged over the entire transcript. For smaller sets, the allowable error grows with the inverse of $\frac{|S|}{T}$ — i.e. the measure of S within the uniform distribution over the transcript. Even in a distributional setting, estimates inevitably degrade with the size of the set we are conditioning on, and our formulation here corresponds exactly to how mean consistency is defined in Chapter 5. Our definitions are also consistent with how the literature on online calibration quantifies calibration error with respect to subsequences. Hébert-Johnson et al. [43] handle this issue slightly differently, by asking for uniform bounds, but in the end proving* *bounds only for sets S that have sufficient mass γ in the underlying probability distribution. In the batch setting, our formulation can recover bounds that are strictly stronger than those of Hébert-Johnson et al. [43] after a reparametrization $\alpha \leftarrow \gamma\alpha$.*

Next, we define multicalibration in our setting. Informally, a sequence of mean predictions is *calibrated* if the average realized label y_t on all days for which $\overline{\mu}_t$ is (roughly) p is (roughly) p. The need to consider days in which the prediction was *roughly* p arises from the fact that a learning algorithm will not necessarily ever make the same prediction twice. More generally, by bucketing predictions at a fixed granularity, we can guarantee that the average number of predictions within each bucket grows linearly with T.

To collect mean predictions $\overline{\mu}_t$ that are approximately equal to p for each p, we group real-valued predictions into n buckets of width $\frac{1}{n}$. Here n is a parameter controlling the coarseness of our calibration guarantee. For any coarseness parameter n and bucket index $i \in [n-1]$, we write $B_n(i) = \left[\frac{i-1}{n}, \frac{i}{n}\right)$ and $B_n(n) = \left[\frac{n-1}{n}, 1\right]$ so that these buckets partition the unit interval. Conversely, given a $\overline{\mu} \in [0,1]$, define $B_n^{-1}(\overline{\mu}) \in [n]$ in the obvious way i.e. $B_n^{-1}(\overline{\mu}) = i$ where i is such that $\overline{\mu} \in B_n(i)$. When clear from the context, we elide the subscript n and write $B(i)$ and $B^{-1}(\overline{\mu})$.

For any $S \subseteq [T]$ and $i \in [n]$, we write

$$S(i) = \left\{t \in S : \overline{\mu}_t \in B_n(i)\right\}.$$

In words, $S(i)$ corresponds to the subset of rounds in S where the mean prediction falls in the i^{th} bucket.

(Simple) calibration asks for the sequence of predictions to be α-mean-consistent on all sets $[T](i)$ for $i \in [n]$ — i.e. for the subset of rounds in which the prediction fell into the i^{th} bucket, for all i. Multicalibration asks for the predictions to be calibrated not just on the overall sequence, but also simultaneously on all the subsequences corresponding to each group $G \in \mathcal{G}$. In our notation, it asks for mean consistency on each set $G(i)$, for every group

$G \in \mathcal{G}$ and $i \in [n]$.

Definition 22 (Mean-Multicalibration)**.** *Given a transcript π_T, we say that the mean predictions $\{\overline{\mu}_t\}_{t=1}^T$ are* (α, n)-mean multicalibrated *with respect to $\mathcal{G}$ if we have that for every $G \in \mathcal{G}$ and $i \in [n]$, the mean-predictions are α-mean consistent on $G_T(i)$:*

$$|\mu(G_T(i)) - \overline{\mu}(G_T(i))| \leq \alpha \frac{T}{|G_T(i)|}.$$

Remark 9. *Note that we define mean multicalibration (and our other notions of multivalidity, shortly) to have two parameters: n, which controls the coarseness of the guarantee, and α, which controls the error of the guarantee. These parameters can be set independently — in the sense that we will be able to achieve (α, n) mean multicalibration for any pair (α, n) — but they should be interpreted together. For example, to avoid the trivial solution in which the learner simply selects uniformly at random at each iteration (thereby guaranteeing that $|G_T(i)| \leq T/n$ for all G, i), we should set $\alpha \ll \frac{1}{n}$.*

(Mean, Moment) Predictions In this case, the prediction domain is the product of the unit interval with itself: $\mathcal{P}_{(\text{mean,moment})} = [0,1] \times [0,1]$. In each round t, the learner selects $p_t = (\overline{\mu}_t, \overline{m}_t^k)$ with the goal of matching $\mathbb{E}[y_t|x_t]$ and $\mathbb{E}[(y_t - \mathbb{E}[y_t|x_t])^k|x_t]$ respectively — the conditional label expectation, and its conditional k^{th} central moment. For simplicity, we assume throughout that k is even, so the k^{th} moment has nonnegative range, but there is no obstacle other than notation to handling odd moments as well.

We group continuous predictions $(\overline{\mu}, \overline{m}^k)$ into a finite set of discrete buckets—again, defined with respect to a pair of discretization parameters n and n'. Recall our bucketing notation for mean prediction: for any $i \in [n-1]$, we wrote $B_n(i) = \left[\frac{i-1}{n}, \frac{i}{n}\right)$ and $B_n(n) = \left[\frac{n-1}{n}, 1\right]$. Here we generalize this notation to pairs, and write for any $i \in [n]$ and $j \in [n']$:

$$B_{n,n'}(i,j) = \left\{(a,b) \in [0,1] \times [0,1] : a \in B_n(i), b \in B_{n'}(j)\right\}.$$

If $n = n'$, we will write $B_n(i, j)$. Once again, when n and n' are clear from the context, we may elide the subscript (n, n') entirely.

Analogously to our notation for mean prediction, for any $S \subseteq [T]$ we write

$$m_k(S) = \frac{1}{|S|} \sum_{t \in S} (y_t - \mu(S))^k, \quad \overline{m}^k(S) = \frac{1}{|S|} \sum_{t \in S} \overline{m}_t^k$$

for the empirical k^{th} central moment of the label distribution on the subsequence S, and for the average of the moment prediction on S, respectively. Just as with mean consistency, moment consistency asks that these two quantities be approximately equal on a set S.

Definition 23 (Moment Consistency)**.** *Given a transcript π_T, we say that moment predictions $\{\overline{m}_t^k\}_{t=1}^T$ are α-moment consistent on set $S \subseteq [T]$ if*

$$|m_k(S) - \overline{m}^k(S)| \leq \alpha \frac{T}{|S|}.$$

It is not sensible to ask for moment consistency on arbitrary sets S, because higher central moments are not linear, and so even true conditional label moments would not satisfy moment consistency conditions on arbitrary sets S. True conditional label moments *do* satisfy moment consistency on sets of points x that share the same label mean, however, and so this is what we will ask of our predictions as well just as in Chapter 5. To that end, for any $S \subseteq [T]$ and $i \in [n], j \in [n']$, we write

$$S(i, j) = \left\{ t \in S : (\overline{\mu}_t, \overline{m}_t^k) \in B_{n,n'}(i, j) \right\}.$$

In words, $S(i, j)$ corresponds to the subset of rounds in S in which our predicted mean and moment fall into the bucket $B_{n,n'}(i, j)$.

Definition 24 (Mean-Conditioned Moment Multicalibration)**.** *Given a transcript π_T, we say that the (mean, moment) predictions $\{(\overline{\mu}_t, \overline{m}_t^k)\}_{t=1}^T$ are* (α, β, n, n')-mean-conditioned moment multicalibrated with respect to $\mathcal{G}$, *if for every* $i \in [n], j \in [n']$ *and* $G \in \mathcal{G}$, *we have that the mean predictions are* α*-mean consistent on* $G_T(i,j)$ *and the moment predictions are* β*-moment consistent on* $G_T(i,j)$*:*

$$|\mu(G_T(i,j)) - \overline{\mu}(G_T(i,j))| \leq \alpha \frac{T}{|G_T(i,j)|},$$
$$|m_k(G_T(i,j)) - \overline{m}^k(G_T(i,j))| \leq \beta \frac{T}{|G_T(i,j)|}.$$

Interval Predictions In this case, the prediction domain is the set of ordered pairs of endpoints in the unit interval: $\mathcal{P}_{\text{interval}} = \{(\ell, u) : \ell \leq u, \text{ and } u, \ell \in [0,1]\}$. Given a pair $(\ell, u) \in \mathcal{P}_{\text{interval}}$, we say that it *covers* a label $y \in [0,1]$ if y falls between ℓ and u, which we write as $\text{Cover}((\ell,u),y) = 1$. To avoid issues of "double counting", we define coverage in the same manner as we defined our bucketing, using intervals that are closed on the left but open on the right, with the exception of $u = 1$:

$$\text{Cover}((\ell,u),y) = \begin{cases} \mathbb{1}(y \in [\ell, u)) & \text{if } u < 1, \\ \mathbb{1}(y \in [\ell, u]) & \text{if } u = 1. \end{cases}$$

In each round t, we will predict an interval $p_t = (\overline{\ell}_t, \overline{u}_t)$ with the goal of achieving

$$\mathbb{E}[\text{Cover}((\overline{\ell}_t, \overline{u}_t), y) | x_t] = 1 - \delta$$

for some target coverage probability $1 - \delta \in [0,1]$. We again bucket our coverage intervals using a discretization parameter n, using the same notation as for moment predictions.

For any $S \subseteq [T]$ and $i \leq j \in [n]$, we write

$$S(i,j) = \left\{ t \in S : (\overline{\ell}_t, \overline{u}_t) \in B_n(i,j) \right\}.$$

In words, $S(i,j)$ corresponds to the subset of rounds in S in which our predicted interval's endpoints are in buckets i and j, respectively. We can now define multivalidity analogously to how we defined multicalibration.

For any $S \subseteq [T]$, we write

$$\overline{H}(S) = \frac{1}{|S|}\sum_{t \in S} \mathrm{Cover}((\bar{\ell}_t, \overline{u}_t), y_t).$$

Definition 25. *We say that interval predictions $\{(\bar{\ell}_t, \overline{u}_t)\}_{t=1}^{T}$ are α-*consistent *on set S with respect to failure probability $\delta \in (0,1)$, if the following holds:*

$$|\overline{H}(S) - (1-\delta)| \leq \alpha \frac{T}{|S|}.$$

Definition 26. *Given a transcript π_T, we say that the interval predictions are (α, n)-*multivalid with respect to δ and $\mathcal{G}$*, if for every $i \leq j \in [n]$ and $G \in \mathcal{G}$, we have that the interval predictions are α-consistent on $G_T(i,j)$ with respect to coverage probability $1-\delta$:*

$$|\overline{H}(G_T(i,j)) - (1-\delta)| \leq \alpha \frac{T}{|G_T(i,j)|}.$$

6.3.3. Zero-sum Games

Our analysis will hinge on properties of zero-sum games, and in particular on the minimax theorem.

Definition 27. *A zero-sum game is defined by:*

1. *A* minimization *player with a convex and compact strategy space $\mathcal{Q}_1 \subseteq \mathbb{R}^{d_1}$ for some $d_1 \in (0, \infty)$.*

2. *A* maximization *player with a convex and compact strategy space $\mathcal{Q}_2 \subseteq \mathbb{R}^{d_2}$ for some $d_2 \in (0, \infty)$.*

3. *An* objective function $u : \mathcal{Q}_1 \times \mathcal{Q}_2 \to \mathbb{R}$*, concave in its first argument and convex in its second argument.*

Zero-sum games are often defined by endowing each player with a finite set of *pure* strategies X_1, X_2. The convex compact strategy sets $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are then formed by allowing players to randomize over their pure strategies and taking $\mathcal{Q}_1 = \Delta X_1$, $\mathcal{Q}_2 = \Delta X_2$ to be the probability simplices over the pure strategies of each player. An objective function $u : X_1 \times X_2 \to \mathbb{R}$ can be linearly extended to $\mathcal{Q}_1$ and $\mathcal{Q}_2$ in the natural way (i.e. by taking expectations over the randomized strategies of each player) – i.e. for any $Q_1 \in \mathcal{Q}_1$ and $Q_2 \in \mathcal{Q}_2$, we write $u(Q_1, Q_2) = \mathbb{E}_{x_1 \sim Q_1, x_2 \sim Q_2}[u(x_1, x_2)]$.

In a zero-sum game, the minimization player chooses some action $Q_1 \in \mathcal{Q}_1$ and the maximization player chooses some action $Q_2 \in \mathcal{Q}_2$, resulting in objective value $u(Q_1, Q_2)$. The goal of the minimization player is to minimize the objective value, and the goal of the maximization player is to maximize it. The key property of zero-sum games, first proved by von Neumann for the case of games with finite sets of pure strategies and generalized to general zero-sum games of the form considered in Definition 27 by Sion, is that the order of play does not affect the objective value that each player can guarantee. This is captured in the minimax theorem, which says that whether the minimization player *first* gets to observe the strategy of the maximization player, and *then* best respond, or whether she must first announce her strategy and allow the maximization player to best respond, she is able to guarantee herself the same value.

Theorem 29 (Sion's Minimax Theorem)**.** *For any zero-sum game* $(\mathcal{Q}_1, \mathcal{Q}_2, u)$*:*

$$\min_{Q_1 \in \mathcal{Q}_1} \max_{Q_2 \in \mathcal{Q}_2} u(Q_1, Q_2) = \max_{Q_2 \in \mathcal{Q}_2} \min_{Q_1 \in \mathcal{Q}_1} u(Q_1, Q_2).$$

The minimax theorem justifies the following definitions:

Definition 28 (Value, Equilibrium, and Best Response)**.** *The* value *of a zero-sum game*

$(\mathcal{Q}_1, \mathcal{Q}_2, u)$ *is the unique* $v \in \mathbb{R}$ *such that*

$$\min_{Q_1 \in \mathcal{Q}_1} \max_{Q_2 \in \mathcal{Q}_2} u(Q_1, Q_2) = \max_{Q_2 \in \mathcal{Q}_2} \min_{Q_1 \in \mathcal{Q}_1} u(Q_1, Q_2) = v.$$

We say that a strategy for the minimization player $Q_1^* \in \mathcal{Q}_1$ *is a (minimax) equilibrium strategy if it guarantees that the objective value is at most the value of the game, for any strategy* $Q_2 \in \mathcal{Q}_2$ *of the maximization player:*

$$\max_{Q_2 \in \mathcal{Q}_2} u(Q_1^*, Q_2) = v.$$

We say that Q_2 *is a best response for the maximization player in response to* Q_1^* *if it realizes the above maximum.*

In our analysis, we will identify the Learner with the minimization player and the Adversary with the maximization player, and so will denote their strategy spaces as $\mathcal{Q}^L$ and $\mathcal{Q}^A$ respectively.

6.4. Online Mean Multicalibration

In this section, we show how to obtain mean multicalibrated estimators in an online adversarial setting. Our derivation also serves as a warm up example of our general technique, which we also instantiate (in somewhat more involved settings) in Sections 6.5 and 6.6 to derive online algorithms for mean-conditioned moment multicalibrated estimators and for multivalid prediction intervals respectively.

6.4.1. An Outline of Our Approach

At a high level, the derivation of our algorithm and its proof of correctness proceeds as follows:

1. For each group $G \in \mathcal{G}$, $i \in [n]$, and transcript π_s up to period s, we define an empirical quantity $V_s^{G,i}$ (Definition 29) which represents the calibration error that our algorithm has incurred with respect to group G over those of the rounds 1 through s when the i^{th} bucket was predicted. These quantities are defined so that if for each G and i,

$|V_T^{G,i}|$ is small, then our algorithm is approximately multicalibrated with respect to $\mathcal{G}$ across T rounds.

The premise of our algorithm will be to greedily make decisions at each round s so as to minimize the *maximum possible increase* of these quantities ($\max_{G,i} |V_{s+1}^{G,i}| - \max_{G,i} |V_s^{G,i}|$), in the worst case over the choices of the adversary. If we could bound this quantity at every round, then by telescoping, we would have a bound on $\max_{G,i} |V_T^{G,i}|$ at the end of the interaction, and therefore a guarantee of mean multicalibration.

2. The increase in the maximum value of $|V_{s+1}^{G,i}|$ is inconvenient to work with, and so we instead define a smooth potential function L_s (Definition 30) corresponding to a soft-max function which upper bounds $\max_{G,i} |V_s^{G,i}|$. Our design goal instead becomes to upper bound the increase in our potential function from round to round: $\Delta_{s+1} = L_{s+1} - L_s$. We view this as defining a zero-sum game, in which the learner's goal is to minimize this increase, and the adversary's goal is to maximize it.

3. We show that for each fixed distribution that the adversary could employ at each round $s+1$, there is a prediction the learner could employ (if only she knew the adversary's distribution) that would guarantee that the increase in potential Δ_{s+1} is small. Intuitively, this is because if we knew the true joint distribution over feature label pairs, then we could predict the true conditional expectations, $\overline{\mu}_{s+1} = \mathbb{E}[y_{s+1}|x_{s+1}]$, which would be perfectly calibrated on all groups. Of course, the learner does not have the luxury of knowing the adversary's distribution before choosing her own. But this thought experiment establishes the value of the game, and so we can conclude via the minimax theorem that there must be some fixed *distribution* over prediction rules that the learner can play that will guarantee Δ_{s+1} being small against *all* actions of the adversary.

4. Step 3 suffices to argue for the *existence* of an algorithm obtaining multicalibration guarantees (Algorithm 16). However, to actually derive an implementable algorithm

we need to find a way to compute the equilibrium strategy at each round, whose existence was argued in Step 3. A priori, this seems daunting because the learner's strategy space consists of all randomized mappings between $\mathcal{X}$ and $\mathcal{Y}$, and the adversary's strategy space consists of all joint distributions on $\mathcal{X} \times \mathcal{Y}$. However, we derive a simple algorithm in Section 6.4.3 that implements the optimal equilibrium strategy needed to realize Step 3. Informally, we are able to do so by representing the mapping between $\mathcal{X}$ and $\mathcal{Y}$ only implicitly, and delaying all computation until x_t has been chosen. We then show that the equilibrium strategy for the learner has a simple structure and randomizes over only at most 2 predictions. Our final algorithm (Algorithm 17) simply computes the relevant portion of the equilibrium strategy at each round and then samples from it.

5. To apply the minimax theorem, and to derive a concrete algorithm, we need to restrict our algorithm to making predictions in $[0, 1]$ that are discretized at units of $1/rn$ for some $r > 1$. This parameter r appears in our final bounds, but neither the runtime of our algorithm nor our convergence rate has any dependence on r, and so it can be imagined to be arbitrarily small. Taking it to be $r = 1/\sqrt{T}$ causes it to become a low order term in our final bounds.

In the appendix of [38] which this chapter is based off of, we give a standard online-to-offline conversion to show how to use our Algorithm 17 to solve offline (batch) multicalibration problems. This gives optimal sample complexity bounds for the offline problem, yielding an improvement over those proven in Hébert-Johnson et al. [43] and Chapter 5. The crux of the improvement is that unlike the algorithms given in [43] and Chapter 5, our algorithm takes only a single pass over the data, and so avoids complications that arise from data re-use. However, unlike previous batch algorithms which make deterministic predictions, the batch algorithm that we obtain through this reduction makes randomized predictions.

6.4.2. An Existential Derivation of the Algorithm and Multicalibration Bounds

We begin by defining notation $V_s^{G,i}$ for the (unnormalized) *portion* of the mean calibration error corresponding to each group $G \in \mathcal{G}$ and bucket $i \in [n]$:

Definition 29. *Given a transcript $\pi_s = ((x_t, \overline{\mu}_t, y_t))_{t=1}^s$, we define the mean calibration error for a group $G \in \mathcal{G}$ and bucket $i \in [n]$ at time s to be:*

$$V_s^{G,i}(\pi_s) = |G_s(i)| \left(\mu\left(G_s(i)\right) - \overline{\mu}\left(G_s(i)\right)\right) = \sum_{t=1}^{s} \mathbb{1}[\overline{\mu}_t \in B(i), x_t \in G] \left(y_t - \overline{\mu}_t\right) \tag{6.1}$$

When the transcript is clear from context we will sometimes simply write $V_s^{G,i}$.

Observe that our definition of mean multicalibration (Definition 22) corresponds to asking that $|V_s^{G,i}|$ be small for all i, G.

Observation 2. *Fix a transcript π_T. If for all $G \in \mathcal{G}$, $i \in [n]$, we have that:*

$$\left|V_T^{G,i}\right| \leq \alpha T,$$

then the corresponding sequence of predictions is (α, n)-mean multicalibrated with respect to $\mathcal{G}$.

We next define a surrogate loss function that we can use to bound our calibration error.

Definition 30 (Surrogate loss function)**.** *Fixing a transcript $\pi_s \in \Pi^*$ and a parameter $\eta \in [0, \frac{1}{2}]$, define a surrogate calibration loss function at day s as:*

$$L_s(\pi_s) = \sum_{\substack{G \in \mathcal{G}, \\ i \in [n]}} \left(\exp(\eta V_s^{G,i}) + \exp(-\eta V_s^{G,i})\right).$$

When the transcript π_s is clear from context, we will sometimes simply write L_s.

We will leave η unspecified for now, and choose it later to optimize our bounds. Observe that this “soft-max style” function allows us to tightly upper bound our calibration loss:

Observation 3. *For any transcript π_T, and any $\eta \in [0, \frac{1}{2}]$, we have that:*

$$\max_{G \in \mathcal{G}, i \in [n]} \left| V_T^{G,i} \right| \leq \frac{1}{\eta} \ln(L_T) \leq \max_{G \in \mathcal{G}, i \in [n]} \left| V_T^{G,i} \right| + \frac{\ln\left(2|\mathcal{G}|n\right)}{\eta}.$$

Part of our analysis will depend on viewing the transcript as a random variable: in this case, in keeping with our convention for random variables, we refer to it as $\tilde{\pi}$. The associated random variables tracking calibration and surrogate loss are denoted $\tilde{V}$ and $\tilde{L}$ respectively.

Our goal is to find a strategy for the learner that guarantees that our surrogate loss L_T remains small. Towards this end, we define $\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu}_{s+1})$ to be the expected increase in the surrogate loss function in the event that the adversary plays feature vector x_{s+1} *and* the learner plays prediction $\overline{\mu}_{s+1}$. Here the expectation is over the only remaining source of randomness after the conditioning — the distribution over labels y_{s+1} (which we observe is determined, once we fix π_s and x_{s+1}).

Definition 31 (Conditional Change in Surrogate Loss)**.**

$$\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu}_{s+1}) = \mathop{\mathbb{E}}_{\tilde{y}_{s+1}} \left[\tilde{L}_{s+1} - L_s \middle| x_{s+1}, \overline{\mu}_{s+1}, \pi_s \right].$$

We begin with a simple bound on $\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu}_{s+1})$:

Lemma 35. *For any transcript $\pi_s \in \Pi^*$, any $x_{s+1} \in \mathcal{X}$, and any $\overline{\mu}_{s+1} \in \mathcal{P}_{mean}$ such that $\overline{\mu}_{s+1} \in B(i)$ for some $i \in [n]$:*

$$\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu}_{s+1}) \leq \eta \left(\mathop{\mathbb{E}}_{\tilde{y}_{s+1}} [\tilde{y}_{s+1}] - \overline{\mu}_{s+1} \right) C_s^i(x_{s+1}) + 2\eta^2 L_s,$$

where for each $i \in [n]$:

$$C_s^i(x_{s+1}) \equiv \sum_{\mathcal{G}(x_{s+1})} \exp(\eta V_s^{G,i}) - \exp(-\eta V_s^{G,i}). \tag{6.2}$$

Proof. Fix any transcript $\pi_s \in \Pi^*$ (which defines L_s), feature vector $x_{s+1} \in \mathcal{X}$, and $\overline{\mu}_{s+1}$ such that $\overline{\mu}_{s+1} \in B(i)$ for some $i \in [n]$. By direct calculation, we obtain:

$$\begin{aligned}
&\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu}_{s+1})\\
&= \mathop{\mathbb{E}}_{\tilde{y}_{s+1}}\Bigg[\sum_{G \in \mathcal{G}(x_{s+1})} \exp(\eta V_s^{G,i})\left(\exp(\eta(\tilde{y}_{s+1} - \overline{\mu}_{s+1})) - 1\right)\\
&+ \exp(-\eta V_s^{G,i})\left(\exp(-\eta(\tilde{y}_{s+1} - \overline{\mu}_{s+1})) - 1\right)\Bigg]\\
&\le \mathop{\mathbb{E}}_{\tilde{y}_{s+1}}\Bigg[\sum_{G \in \mathcal{G}(x_{s+1})} \exp(\eta V_s^{G,i})\left(\eta(\tilde{y}_{s+1} - \overline{\mu}_{s+1}) + 2\eta^2\right) + \exp(-\eta V_s^{G,i})\left(-\eta(\tilde{y}_{s+1} - \overline{\mu}_{s+1}) + 2\eta^2\right)\Bigg]\\
&= \eta\left(\mathop{\mathbb{E}}_{\tilde{y}_{s+1}}[\tilde{y}_{s+1}] - \overline{\mu}_{s+1}\right)\sum_{G \in \mathcal{G}(x_{s+1})}\left(\exp(\eta V_s^{G,i}) - \exp(-\eta V_s^{G,i})\right)\\
&+ 2\eta^2 \sum_{G \in \mathcal{G}(x_{s+1})}\left(\exp(\eta V_s^{G,i}) + \exp(-\eta V_s^{G,i})\right)\\
&\le \eta\left(\mathop{\mathbb{E}}_{\tilde{y}_{s+1}}[\tilde{y}_{s+1}] - \overline{\mu}_{s+1}\right)\left(\sum_{G \in \mathcal{G}(x_{s+1})} \exp(\eta V_s^{G,i}) - \exp(-\eta V_s^{G,i})\right) + 2\eta^2 L_s\\
&= \eta\left(\mathop{\mathbb{E}}_{\tilde{y}_{s+1}}[\tilde{y}_{s+1}] - \overline{\mu}_{s+1}\right) C_s^i(x_{s+1}) + 2\eta^2 L_s.
\end{aligned}$$

Here, the first inequality follows from the fact that for $0 < |x| < \frac{1}{2}$, $\exp(x) \le 1 + x + 2x^2$. □

Using this bound, we define a zero-sum game between the learner and the adversary and use the minimax theorem to conclude that the learner always has a strategy that guarantees that the per-round increase in surrogate loss can be bounded. To satisfy the convexity and compactness requirements of the minimax theorem, it will be convenient for us to imagine that the learner's pure strategy space is a finite, discrete subset of $\mathcal{P}_{\text{mean}} = [0, 1]$. To this end, we define the following discretization for any $r \in \mathbb{N}$ (here n is the discretization parameter we use to define the coarseness of our bucketing):

$$\mathcal{P}^{rn} = \left\{0, \frac{1}{rn}, \frac{2}{rn}, \dots, 1\right\}.$$

We use this discretization also in our algorithm in Section 6.4.3 — but we remark at the outset that the need to discretize is only for technical reasons, and our algorithm will have no dependence — neither in runtime nor in its convergence rate — on the value of r that we choose, so we can imagine the discretization to be arbitrarily fine.

To simplify notation, for each $\overline{\mu} \in \mathcal{P}^{rn}$, define $C_s^{\overline{\mu}} \equiv C_s^i$ where $i \in [n]$ s.t. $\overline{\mu} \in B_n(i)$.

Lemma 36. *For any transcript $\pi_s \in \Pi^*$, any $x_{s+1} \in \mathcal{X}$, and any $r \in \mathbb{N}$ there exists a distribution over predictions for the learner $Q_{s+1}^L \in \Delta\mathcal{P}^{rn}$, such that regardless of the adversary's choice of distribution of y_{s+1} over $\Delta\mathcal{Y}$, we have that:*

$$\mathop{\mathbb{E}}_{\overline{\mu} \sim Q_{s+1}^L} [\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu})] \leq L_s \left(\frac{\eta}{rn} + 2\eta^2\right).$$

Proof. We define a zero-sum game played between the learner (the minimization player) and the adversary (the maximization player). The learner's pure strategy space is the set of discrete predictions $X_1 = \mathcal{P}^{rn}$. The adversary's pure strategy space is (a priori) the set of all distributions over labels in $[0, 1]$. However, we will observe in a moment that the objective function of our game depends only on the *expected value* of the label, and so without loss of generality, we will be able to take the adversary's full strategy space to be the set of all pure strategies, i.e., $\mathcal{Q}^A = [0, 1]$ (which is closed and convex), because it already spans the set of realizable expectations. As usual, we take the learner's full strategy space to be the set of distributions over pure strategies: $\mathcal{Q}^L = \Delta\mathcal{P}^{rn}$.

Fix the transcript π_s and the feature vector x_{s+1}. We define the objective of this game to be the upper bound we proved on $\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu})$ in Lemma 35. For each $\overline{\mu} \in \mathcal{P}^{rn}$ and each $y \in [0,1]$, we let:

$$u(\overline{\mu}, y) = \eta \left(y - \overline{\mu}\right) C_s^{\overline{\mu}}(x_{s+1}) + 2\eta^2 L_s.$$

Note that for any distribution over labels y of the adversary, the expected objective value depends on his strategy only through $\mathbb{E}[\tilde{y}]$ because the above objective function is linear in y: that is, $\mathbb{E}_{\tilde{y}}[u(\overline{\mu}, \tilde{y})] = u(\overline{\mu}, \mathbb{E}[\tilde{y}])$. Thus we are justified in our reduced-form representation of the adversary's full strategy as choosing $\mathbb{E}[\tilde{y}]$ in the interval $[0,1]$.

We now establish the value of this game. Observe that for any strategy of the adversary (which fixes $\mathbb{E}[\tilde{y}]$), the learner can respond by playing $\overline{\mu}^* = \operatorname{argmin}_{\overline{\mu} \in \mathcal{P}^{rn}} |\,\mathbb{E}[\tilde{y}] - \overline{\mu}|$, and that because of our discretization, $\min |\,\mathbb{E}[\tilde{y}] - \overline{\mu}^*| \leq \frac{1}{rn}$. Therefore, the value of the game is at most:

$$\begin{aligned}
\max_{y \in [0,1]} \min_{\overline{\mu}^* \in \mathcal{P}^{rn}} u(\overline{\mu}^*, y) \quad &\leq \quad \max_{\overline{\mu} \in \mathcal{P}^{rn}} \frac{\eta}{rn} \left| C_s^{\overline{\mu}}(x_{s+1}) \right| + 2\eta^2 L_s, \\
&\leq \quad L_s \left(\frac{\eta}{rn} + 2\eta^2 \right).
\end{aligned}$$

Here the latter inequality follows since $C_s^{\overline{\mu}}(x_{s+1}) \leq L_s$ for all $\overline{\mu} \in \mathcal{P}^{rn}$, by observation. We can now apply the minimax theorem (Theorem 29) to conclude that there exists a fixed distribution $Q_{s+1}^L \in \mathcal{Q}^L$ for the learner that guarantees that simultaneously for *every* label $y \in [0,1]$ that might be chosen by the adversary:

$$\mathop{\mathbb{E}}_{\overline{\mu} \sim Q_{s+1}^L} \left[u(\overline{\mu}, y) \right] \leq L_s \left(\frac{\eta}{rn} + 2\eta^2 \right),$$

as desired. □

Corollary 5. *For every $r \in \mathbb{N}$, $s \in [T]$, $\pi_s \in \Pi^*$, and $x_{s+1} \in \mathcal{X}$ (which fixes L_s and Q_{s+1}^L), and any distribution over $\mathcal{Y}$:*

$$\mathop{\mathbb{E}}_{\overline{\mu}_{s+1} \sim Q_{s+1}^L} [\tilde{L}_{s+1} | \pi_s] = L_s + \mathop{\mathbb{E}}_{\overline{\mu}_{s+1} \sim Q_{s+1}^L} [\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu}_{s+1})] \leq L_s \left(1 + \frac{\eta}{rn} + 2\eta^2 \right).$$

Lemma 36 defines (existentially) an algorithm that the learner can use to make predictions—Algorithm 16. We will now show that Algorithm 16 (if we could compute the distributions Q_t^L) results in multicalibrated predictions. In Section 6.4.3 we show a simple and efficient method for sampling from Q_t^L.

Algorithm 16: A Generic Multicalibrator

```
for $t = 1, \ldots, T$ do
    Observe $x_t$. Given $\pi_{t-1}$ and $x_t$, let $Q_t^L \in \mathcal{Q}_t^L$ be the distribution over predictions
      whose existence
    is established in Lemma 36. Sample $\overline{\mu} \sim Q_t^L$ and predict $\overline{\mu}_t = \overline{\mu}$
end
```

We now prove two convergence bounds for Algorithm 16. The first will bound its multicalibration error *in expectation*, and the other will provide a high probability bound. To show these bounds, we first state a helper theorem that will be useful not just in this section, but also in deriving the final convergence bounds for the algorithms presented in Sections 6.5 and 6.6. The proof is in Appendix 6.C.

Theorem 30. *Consider a nonnegative random process $\tilde{X}_t$ adapted to the filtration $\mathcal{F}_t = \sigma(\pi_t)$, where $\tilde{X}_0$ is constant a.s. Suppose we have that for any period t, and any π_{t-1}, $\mathbb{E}[\tilde{X}_t|\pi_{t-1}] \le X_{t-1}(1 + \eta c + 2\eta^2)$ for some $\eta \in [0, \frac{1}{2}], c \in [0, 1]$. Then we have that:*

$$\mathop{\mathbb{E}}_{\tilde{\pi}_T}[\tilde{X}_T] \le X_0 \exp\left(T\eta c + 2T\eta^2\right). \tag{6.3}$$

Further, define a process $\tilde{Z}_t$ adapted to the same filtration by $\tilde{Z}_t = Z_{t-1} + \ln \tilde{X}_t - \mathbb{E}[\ln(\tilde{X}_t)|\pi_{t-1}]$. Suppose that $|Z_t - Z_{t-1}| \le 2\eta$, where $Z_0 = 0$ a.s. Then, with probability $1 - \lambda$,

$$\ln(X_T(\pi_T)) \le \ln(X_0) + T\left(\eta c + 2\eta^2\right) + \eta\sqrt{8T \ln\left(\frac{1}{\lambda}\right)}. \tag{6.4}$$

We are now ready to bound our multicalibration error. As a straightforward consequence of Corollary 5 and the first part of Theorem 30, we have the following Corollary.

Corollary 6. *Against any adversary, Algorithm 16 instantiated with discretization parameter r results in surrogate loss satisfying:*

$$\mathop{\mathbb{E}}_{\tilde{\pi}_T}[\tilde{L}_T] \leq 2|\mathcal{G}|n \exp\left(\frac{T\eta}{rn} + 2T\eta^2\right).$$

Proof. Note that the first part of Theorem 30 applies to the process L with $L_0 = 2|\mathcal{G}|n$ and $c = \frac{1}{rn}$. The bound follows by plugging these values into (6.3). □

Next, we can convert this into a bound on Algorithm 16's expected calibration error:

Theorem 31. *When Algorithm 16 is run using n buckets for calibration, discretization $r \in \mathbb{N}$, and $\eta = \sqrt{\frac{\ln(2|\mathcal{G}|n)}{2T}} \in (0, 1/2)$, then against any adversary, its sequence of mean predictions is (α, n)-multicalibrated with respect to $\mathcal{G}$, where:*

$$\mathbb{E}[\alpha] \leq \frac{1}{rn} + 2\sqrt{\frac{2\ln(2|\mathcal{G}|n)}{T}}.$$

For $r = \frac{\sqrt{T}}{\epsilon n\sqrt{2\ln(2|\mathcal{G}|n)}}$ this gives:

$$\mathbb{E}[\alpha] \leq (2+\epsilon)\sqrt{\frac{2}{T}\ln\left(2|\mathcal{G}|n\right)}.$$

Here the expectation is taken over the randomness of the transcript π_T.

Proof. From Observation 2, it suffices to show that

$$\frac{1}{T}\mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\max_{G\in\mathcal{G}, i\in[n]} |\tilde{V}_T^{G,i}|\right] \leq \frac{1}{rn} + 2\sqrt{\frac{2\ln(2|\mathcal{G}|n)}{T}}.$$

We begin by computing a bound on the (exponential of) the expectation of this quantity:

$$\begin{aligned}
\exp\left(\eta \mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\max_{G,i}|\tilde{V}_T^{G,i}|\right]\right) &\leq \mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\exp\left(\eta \max_{G,i}|\tilde{V}_T^{G,i}|\right)\right], \\
&= \mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\max_{G,i}\exp\left(\eta|\tilde{V}_T^{G,i}|\right)\right], \\
&\leq \mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\max_{G,i}\left(\exp\left(\eta\tilde{V}_T^{G,i}\right)+\exp\left(-\eta\tilde{V}_T^{G,i}\right)\right)\right], \\
&\leq \mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\sum_{G,i}\left(\exp\left(\eta\tilde{V}_T^{G,i}\right)+\exp\left(-\eta\tilde{V}_T^{G,i}\right)\right)\right], \\
&= \mathop{\mathbb{E}}_{\tilde{\pi}_T}[\tilde{L}_T], \\
&\leq 2|\mathcal{G}|n\exp\left(\frac{T\eta}{rn}+2T\eta^2\right).
\end{aligned}$$

Here the first step is by Jensen's inequality and the last one follows from Corollary 6. Taking the logarithm of both sides and dividing by ηT, we have

$$\frac{1}{T}\mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\max_{G,i}|\tilde{V}_T^{G,i}|\right] \leq \frac{\ln(2|\mathcal{G}|n)}{\eta T}+\frac{1}{rn}+2\eta.$$

Choosing $\eta = \sqrt{\frac{\ln(2|\mathcal{G}|n)}{2T}}$, we thus obtain the desired inequality

$$\frac{1}{T}\mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\max_{G,i}|\tilde{V}_T^{G,i}|\right] \leq \frac{1}{rn}+2\sqrt{\frac{2\ln(2|\mathcal{G}|n)}{T}}.$$

□

Now, given $\tilde{L}$, let us define its associated martingale process $\tilde{Z}$ as in the second part of Theorem 30. The next lemma shows that the increments of $\tilde{Z}$ are uniformly bounded over all rounds t. The proof is in Appendix 6.C.

Lemma 37. *At any round $t \in [T]$ and for any realized transcript π_t, $|Z_t - Z_{t-1}| \leq 2\eta$.*

We can now use the second part of Theorem 30 to prove a high probability bound on the multicalibration error of Algorithm 16.

Theorem 32. *When Algorithm 16 is run using n calibration buckets, discretization $r \in \mathbb{N}$ and $\eta = \sqrt{\frac{\ln(2|\mathcal{G}|n)}{2T}} \in (0, 1/2)$, then against any adversary, its sequence of mean predictions is α-multicalibrated, with respect to $\mathcal{G}$ with probability $1-\lambda$ over the randomness of the transcript π_T, for*

$$\alpha \leq \frac{1}{rn} + 4\sqrt{\frac{2}{T}\ln\left(\frac{2|\mathcal{G}|n}{\lambda}\right)}.$$

Choosing $r = \frac{\sqrt{T}}{\epsilon n\sqrt{2\ln(2|\mathcal{G}|n/\lambda)}}$, this gives:

$$\alpha \leq (4+\epsilon)\sqrt{\frac{2}{T}\ln\left(\frac{2|\mathcal{G}|n}{\lambda}\right)}.$$

Proof. By Lemma 37, the second part of Theorem 30 applies; plugging in $L_0 = 2|\mathcal{G}|n$ and $c = \frac{1}{rn}$, we have:

$$\ln(L_T(\pi_T)) \leq \ln(2|\mathcal{G}|n) + T\left(\frac{\eta}{rn} + 2\eta^2\right) + \eta\sqrt{8T\ln\left(\frac{1}{\lambda}\right)}.$$

Now, note that

$$\begin{aligned}
\exp\left(\eta \max_{G,i}|V_T^{G,i}|\right) &= \max_{G,i}\exp\left(\eta|V_T^{G,i}|\right), \\
&\leq \max_{G,i}\left(\exp\left(\eta V_T^{G,i}\right) + \exp\left(-\eta V_T^{G,i}\right)\right), \\
&\leq \sum_{G,i}\left(\exp\left(\eta V_T^{G,i}\right) + \exp\left(-\eta V_T^{G,i}\right)\right), \\
&= L_T(\pi_T).
\end{aligned}$$

Taking log on both sides and dividing both sides by ηT, we get

$$\frac{1}{T}\max_{G,i}|V_T^{G,i}| \leq \frac{1}{\eta T}\ln(L_T(\pi_T)) \leq \frac{\ln(2|\mathcal{G}|n)}{\eta T} + \frac{1}{rn} + 2\eta + \sqrt{\frac{8\ln\left(\frac{1}{\lambda}\right)}{T}}.$$

Choosing $\eta = \sqrt{\frac{\ln(2|\mathcal{G}|n)}{2T}}$, we thus obtain the desired inequality

$$\frac{1}{T}\max_{G,i}|V_T^{G,i}| \leq \frac{1}{rn} + 2\sqrt{\frac{2\ln(2|\mathcal{G}|n)}{T}} + \sqrt{\frac{8\ln\left(\frac{1}{\lambda}\right)}{T}} \leq \frac{1}{rn} + 4\sqrt{\frac{2}{T}\ln\left(\frac{2|\mathcal{G}|n}{\lambda}\right)}. \qquad \square$$

Remark 10. *In both Theorems 31 and 32, the dependence on* $\log(|\mathcal{G}|)$ *can be replaced with a dependence on* $\log(d)$ *under the assumption that* $|\mathcal{G}(x_t)| \leq d$ *for all* t *— i.e. that each observed data point is contained in only boundedly many groups. This gives us non-trivial guarantees even when* $\mathcal{G}$ *is infinitely large. See Appendix 6.A for details.*

6.4.3. Deriving an Efficient Algorithm via Equilibrium Computation

Algorithm 17: Von Neumann's Mean Multicalibrator(η, n, r)

```
for $t = 1, \ldots, T$ do
    Observe $x_t$ and compute for each $i \in [n]$ $C^i_{t-1}(x_t)$ as defined in (6.2). if
    $C^i_{t-1}(x_t) > 0$ for all $i \in [n]$ then
        Predict $\overline{\mu}_t = 1$.
    else if $C^i_{t-1}(x_t) < 0$ for all $i \in [n]$ then
        Predict $\overline{\mu}_t = 0$.
    else
        Find $i^* \in [n-1]$ such that $C^{i^*}_{t-1}(x_t) \cdot C^{i^*+1}_{t-1}(x_t) \leq 0$
        Define $0 \leq q_t \leq 1$ such that $q_t C^{i^*}_{t-1}(x_t) + (1-q_t)C^{i^*+1}_{t-1}(x_t) = 0$. In other
         words, define it as follows (using the convention that $0/0 = 1$):

            $$q_t = \frac{|C^{i^*+1}_{t-1}(x_t)|}{|C^{i^*+1}_{t-1}(x_t)| + |C^{i^*}_{t-1}(x_t)|}.$$

        Predict $\overline{\mu}_t = \frac{i^*}{n} - \frac{1}{rn}$ with probability $q_t$ and $\overline{\mu}_t = \frac{i^*}{n}$ with probability $1 - q_t$.
    end
end
```

In Section 6.4.2, we derived Algorithm 16 and proved that it results in mean multicalibrated predictions. However, Algorithm 16 was not defined explicitly: it relies on the distributions Q^L_t, whose existence we showed in Lemma 36 but which we did not explicitly construct. In this section, we derive a scheme for sampling from these distributions Q_t^L, which leads to Algorithm 17 — an explicit, efficient implementation of Algorithm 16.

Theorem 33. *Algorithm 17 implements Algorithm 16. In particular it obtains the multicalibration guarantees proven in Theorems 31 and 32.*

Proof. Recall that Algorithm 16 samples at every round $s+1$ from a distribution Q_{s+1}^L that is a minimax equilibrium strategy of a game between the learner and the adversary, with objective function

$$u(\overline{\mu}, y) = \eta\,(y - \overline{\mu})\,C_s^{\overline{\mu}}(x_{s+1}) + 2\eta^2 L_s.$$

The equilibrium structure of the game is preserved under positive affine transformations, so instead we consider

$$u(\overline{\mu}, y) = (y - \overline{\mu})\,C_s^{\overline{\mu}}(x_{s+1}).$$

We wish to find a distribution $Q_{s+1}^L \in \mathcal{Q}^L$ that guarantees — against any strategy of the adversary — an objective value that is at most the bound on the value of the game we proved in Lemma 36. For the transformed game, this bound is:

$$\max_{y\in[0,1]} \mathop{\mathbb{E}}_{\overline{\mu}\sim Q_{s+1}} [u(\overline{\mu}, y)] \leq \frac{1}{rn} L_s.$$

We can start by characterizing the best response of the adversary.

Observation 4. *For any $Q^L \in \mathcal{Q}^L$:*

$$\max_{y\in[0,1]} \mathop{\mathbb{E}}_{\overline{\mu}\sim Q^L} [u(\overline{\mu}, y)] = \left(\mathop{\mathbb{E}}_{\overline{\mu}\sim Q^L} [C_s^{\overline{\mu}}(x_{s+1})]\right)^+ - \mathop{\mathbb{E}}_{\overline{\mu}\sim Q^L} \left[\overline{\mu} C_s^{\overline{\mu}}(x_{s+1})\right],$$

where $(x)^+ = \max(x, 0)$.

Proof. Note that:

$$\begin{aligned} u(\mu, y) &= (y - \overline{\mu})\, C_s^{\overline{\mu}}(x_{s+1}) \\ &= y C_s^{\overline{\mu}}(x_{s+1}) - \overline{\mu} C_s^{\overline{\mu}}(x_{s+1}). \end{aligned}$$

Observe that only the first term depends on y. Therefore, if the learner plays according to Q^L, then the adversary will choose y so as to maximize the linear expression $y\,\mathbb{E}_{\overline{\mu}\sim Q^L}[C_s^{\overline{\mu}}(x_{s+1})]$. This is always maximized either at $y = 0$ or $y = 1$. It is maximized at $y = 1$ when $\mathbb{E}_{\overline{\mu}\sim Q^L}[C_s^{\overline{\mu}}(x_{s+1})] > 0$, and at $y = 0$ otherwise. □

Finally, we can reduce the analysis to three disjoint cases:

1. $C_s^i(x_{s+1}) > 0$ for all $i \in [n]$: Then for any distribution Q^L, by Observation 4 we have:

$$\max_{y\in[0,1]} \mathop{\mathbb{E}}_{\overline{\mu}\sim Q^L}[u(\overline{\mu}, y)] = \mathop{\mathbb{E}}_{\overline{\mu}\sim Q^L}[C_s^{\overline{\mu}}(x_{s+1})] - \mathop{\mathbb{E}}_{\overline{\mu}\sim Q^L}\left[\overline{\mu} C_s^{\overline{\mu}}(x_{s+1})\right].$$

   In this case, letting Q^L be a point mass on $\overline{\mu} = 1$ achieves a value of $0 < \frac{1}{rn} L_s$.

2. $C_s^i(x_{s+1}) < 0$ for all $i \in [n]$: Then for any distribution Q^L, by Observation 4 we have:

$$\max_{y\in[0,1]} \mathop{\mathbb{E}}_{\overline{\mu}\sim Q^L}[u(\overline{\mu}, y)] = -\mathop{\mathbb{E}}_{\overline{\mu}\sim Q^L}\left[\overline{\mu} C_s^{\overline{\mu}}(x_{s+1})\right]$$

   In this case, letting Q^L be a point mass on $\overline{\mu} = 0$ achieves a value of $0 < \frac{1}{rn} L_s$.

3. In the remaining case, there must exist some index $i^* \in [n-1]$ such that either $C_s^{i^*}(x_{s+1})$ and $C_s^{i^*+1}(x_{s+1})$ have opposite signs, or such that at least one of them

takes value exactly zero. Randomizing as in the algorithm results in:

$$\begin{aligned}
&\max_{y\in[0,1]} \mathop{\mathbb{E}}_{\overline{\mu}\sim Q^L_{s+1}} [u(\overline{\mu}, y)] \\
=&\left(\mathop{\mathbb{E}}_{\overline{\mu}\sim Q^L_{s+1}} \left[C_s^{\overline{\mu}}(x_{s+1})\right]\right)^+ - E_{\overline{\mu}\sim Q^L_{s+1}} \left[\overline{\mu} C_s^{\overline{\mu}}(x_{s+1})\right] \\
=&\left(q_{s+1}C_s^{i^*}(x_{s+1}) + (1-q_{s+1})C_s^{i^*+1}(x_{s+1})\right)^+ \\
&-\left(q_{s+1}\left(\tfrac{i^*}{n} - \tfrac{1}{rn}\right) C_s^{i^*}(x_{s+1}) + (1-q_{s+1})\tfrac{i^*}{n}C_s^{i^*+1}(x_{s+1})\right) \\
=&\frac{1}{rn}C_s^{i^*}(x_{s+1}) \\
\leq&\frac{1}{rn}L_s.
\end{aligned}$$

Algorithm 17 plays according to this distribution Q^L_{s+1} at every round, which completes the proof. □

Running Time Our algorithm is elementary, and given values for $C^i_{t-1}(x_t)$, it runs in time per iteration which is linear in the number of buckets n. For large collections of groups $\mathcal{G}$, the bulk of the computational cost is due to the first step of Algorithm 17, in which we compute the quantities $C^i_{t-1}(x_t)$ as in Equation 6.2:

$$C^i_{t-1}(x_t) \equiv \sum_{\mathcal{G}(x_t)} \exp(\eta V^{G,i}_{t-1}) - \exp(-\eta V^{G,i}_{t-1})$$

These quantities are a sum over every group $G \in \mathcal{G}$ such that $x_t \in G$. In the worst case, we can compute this by enumerating over all such groups, and we obtain runtime that is linear in $|\mathcal{G}|$. However, for any class $\mathcal{G}$ such that we can efficiently enumerate the set of groups containing x_t (i.e. $\mathcal{G}(x_t)$), our per-round runtime is only linear in $|\mathcal{G}(x_t)|$, which may be substantially smaller than $|\mathcal{G}|$. For example, this property holds for collections $\mathcal{G}$ of groups induced by conjunctions or disjunctions of binary features. Finally, we observe that our runtime is entirely independent of the choice of the discretization parameter r.

6.5. Online Moment Multicalibration

6.5.1. An Outline of Our Approach

In this section, we derive an online algorithm for supplying mean and k^{th}-moment predictions that are mean-conditioned moment multicalibrated with respect to some collection of groups $\mathcal{G}$, as defined in Definition 24. We follow the same basic strategy that we developed in Section 6.4 for making multicalibrated mean predictions. In particular, the first few steps of our approach exactly mirror the approach in Section 6.4: Analogously to Steps 1 and 2 of Section 6.4.1 we define calibration losses and a convenient soft-max style surrogate loss function and bound the increase to that surrogate loss function at each round. However, we make a couple of important deviations.

1. The first complication that arises is that moment consistency is not a linearly separable constraint across rounds (because moments are nonlinear). However, we are able to define linearly separable "pseudo-moment" consistency losses M and prove in Lemma 38 that if both our pseudo-moment consistency losses M and our mean consistency losses V are small then our predictions are mean-conditioned moment multicalibrated.

2. The next complication arises when we attempt to define a zero-sum game using our bound on the per-round increase of the surrogate loss. The bound on the loss that we obtain for mean-conditioned moment multicalibration is nonlinear in both the learner's (mean) prediction and the adversary's choice of label y. We cannot directly apply a minimax theorem because the necessary concavity and convexity conditions are not satisfied. Our argument instead requires a change of variables: we show that in the game we define, the adversary's payoff, fixing the strategy of the learner, is linear in the first k (uncentered) moments of the distribution over the labels chosen by the adversary. We also expand the strategy space of the adversary to allow him to pick k arbitrary real numbers, representing the first k centered moments of his label distribution, unencumbered by the requirement that these chosen values actually

correspond to the moments of any real label distribution. Enlarging the adversary's strategy space in this way can only *increase* the value of the game, and so the upper bounds we prove on the value of this simplified game continue to hold for the original game. Moreover, a minimax theorem applies to this transformed game, and therefore guarantees the *existence* of a prediction strategy for the learner that is approximately mean-conditioned moment multicalibrated.

3. In order to implement this strategy with an explicit efficient prediction algorithm, we need to solve a game in which the learner has $r^2 nn'$ pure strategies. Doing this naively would inherit a running time dependence on r, a discretization parameter that we want to take to be very small. However, we prove a "structure theorem" about the enlarged game described above: that without loss of generality, the learner need only randomize over a support of at most $4nn'$ pure strategies. With this structure theorem in hand, we show that the equilibrium computation problem can be cast as a linear program with $4nn'$ variables and $2^k + 1$ constraints. If k is a small constant (e.g. $k = 2$ for variance multicalibration), then this linear program can be explicitly described and solved. But even when k is too large to enumerate all 2^k constraints, we show that there is a separation oracle that runs in time $O(k)$, allowing us to efficiently solve this linear program using the Ellipsoid algorithm. In Appendix 6.B, we show that there exist solutions to the learner's problem that have small support—in which the learner mixes over at most $k + 1$ strategies.

6.5.2. *An Existential Derivation of the Algorithm and Moment Multicalibration Bounds*

We will calibrate our mean predictions $\{\overline{\mu}_t\}_{t=1}^T$ over n buckets, and k^{th} moment predictions $\{\overline{m}^k\}_{t=1}^T$ over $n' < n$ buckets. As before, we introduce notation to denote the *portion* of the mean calibration error corresponding to each pair of buckets (i, j) and group G, and consider a similar quantity that serves as a proxy for the portion of the moment calibration error corresponding to each group $G \in \mathcal{G}$ and buckets $i \in [n]$, $j \in [n']$. We will need an extra piece of notation: for any $i \in [n]$, define $\hat{\mu}_i \equiv \frac{2i-1}{2n}$. For any $i \in [n]$ and $\overline{\mu} \in B_n(i)$, we abuse notation and write $\hat{\mu}_{\overline{\mu}} = \hat{\mu}_i$.

Definition 32. *Given a transcript $\pi_s = ((x_t, (\overline{\mu}_t, \overline{m}_t^k), y_t))_{t=1}^s$, for each group $G \in \mathcal{G}$ and buckets $i \in [n], j \in [n']$ at time s, we write*

$$V_s^{G,i,j}(\pi_s) = \sum_{t=1}^{s} \mathbb{1}[\overline{\mu}_t \in B_n(i), \overline{m}_t^k \in B_n(j), x_t \in G]\,(y_t - \overline{\mu}_t)\,,$$
$$M_s^{G,i,j}(\pi_s) = \sum_{t=1}^{s} \mathbb{1}[\overline{\mu}_t \in B_n(i), \overline{m}_t^k \in B_n(j), x_t \in G]\left((y_t - \hat{\mu}_i)^k - \overline{m}_t^k\right).$$

When the transcript π_s is clear from context we will simply write $V_s^{G,i,j}, M_s^{G,i,j}$.

In words, $V_s^{G,i,j}$ calculates the difference between the true mean and the mean of our predictions over the subset of periods up to s in which the realized feature vector was in group G and the learner predicted a mean $\overline{\mu} \in B_n(i)$ and a moment $\overline{m}^k \in B_{n'}(j)$. $M_s^{G,i,j}$ defines a similar quantity for moments — but not exactly. Instead of calculating the empirical moment around the empirical mean (i.e. $(y_t - \mu(G_s(i,j)))^k$), we center around $\hat{\mu}_i$, i.e. the middle of the bucket $B_n(i)$. We do this to make $M_s^{G,i,j}$ linearly separable across rounds.

We show, using an argument similar[23] to Chapter 5, that if our mean predictions are sufficiently calibrated — which ensures $\hat{\mu}_i \approx \mu(G_T(i,j))$ — then we can still bound the mean-conditioned moment multicalibration error through our proxy quantity $M_s^{G,i,j}$.

Lemma 38. *For a given $i \in [n], j \in [n']$ and $G \in \mathcal{G}$, if $\frac{1}{T}|V_T^{G,i,j}| \le \alpha, \frac{1}{T}|M_T^{G,i,j}| \le \beta$, then we have*

$$|\mu(G_T(i,j)) - \overline{\mu}(G_T(i,j))| \le \frac{\alpha T}{|G_T(i,j)|}, \qquad \text{(Mean Consistency)}$$
$$\left|m^k(G_T(i,j)) - \overline{m}^k(G_T(i,j))\right| \le \frac{(\beta + k\alpha + \frac{k}{2n})T}{|G_T(i,j)|}. \qquad \text{(Moment Consistency)}$$

[23] $(y_t - \hat{\mu}_i)^k$ roughly corresponds to what is referred to as a pseudo-moment in Chapter 5

Proof. It is easy to see mean-consistency:

$$\frac{|G_T(i,j)|}{T}\left|\overline{\mu}(G_T(i,j)) - \mu(G_T(i,j))\right| = \frac{1}{T}\left|\sum_{t\in G_T(i,j)} (\overline{\mu}_t - y_t)\right| = \frac{1}{T}|V_T^{G,i,j}| \leq \alpha.$$

Now, we show that we achieve mean-conditioned moment consistency. First note that

$$\frac{1}{T}|M_T^{G,i,j}| = \frac{1}{T}\left|\sum_{t\in G_T(i,j)} \overline{m}_t^k - (\hat{\mu}_i - y_t)^k\right| \leq \beta.$$

Now,

$$\begin{aligned}
&\left|m^k(G_T(i,j)) - \frac{1}{|G_T(i,j)|}\sum_{t\in G_T(i,j)} (y_t - \hat{\mu}_i)^k\right| \\
=&\left|\frac{1}{|G_T(i,j)|}\sum_{t\in G_T(i,j)} ((y_t - \hat{\mu}_i) + (\hat{\mu}_i - \mu(G_T(i,j))))^k - (y_t - \hat{\mu}_i)^k\right|, \\
\leq&\frac{k}{|G_T(i,j)|}\sum_{t\in G_T(i,j)} \left|\hat{\mu}_i - \mu(G_T(i,j))\right|, \\
=&\frac{k}{|G_T(i,j)|}\sum_{t\in G_T(i,j)} \left|\hat{\mu}_i - \overline{\mu}(G_T(i,j)) + \overline{\mu}(G_T(i,j)) - \mu(G_T(i,j))\right|, \\
\leq&\frac{k}{|G_T(i,j)|}\sum_{t\in G_T(i,j)} \left|\hat{\mu}_i - \overline{\mu}(G_T(i,j))\right| + \left|\overline{\mu}(G_T(i,j)) - \mu(G_T(i,j))\right|, \\
\leq&\frac{Tk(\alpha + \frac{1}{2n})}{|G_T(i,j)|},
\end{aligned}$$

where the first inequality follows from the fact that $|a^k - b^k| \leq k|a-b|$ for any $a, b \in [0,1]$ with $a = (y_t - \hat{\mu}_i) + (\hat{\mu}_i - \mu(G_T(i,j)))$ and $b = y_t - \hat{\mu}_i$. The last inequality follows from the guarantee of mean consistency as shown above in the proof and the fact that $\overline{\mu}(G_T(i,j)) \in B_n(i)$ and $|\hat{\mu}_i - x| \leq \frac{1}{2n}$ for any $x \in B_n(i)$.

Therefore, we can invoke the triangle inequality to conclude

$$\begin{aligned}
&\left|m^k(G_T(i,j)) - \overline{m}^k(G_T(i,j))\right| \\
&\leq \left|m^k(G_T(i,j)) - \frac{1}{|G_T(i,j)|}\sum_{t\in G_T(i,j)}(y_t - \hat{\mu}_i)^k\right| \\
&+ \left|\frac{1}{|G_T(i,j)|}\sum_{t\in G_T(i,j)}(y_t - \hat{\mu}_i)^k - \overline{m}^k(G_T(i,j))\right| \\
&\leq \frac{(\beta + k\alpha + \frac{k}{2n})T}{|G_T(i,j)|}.
\end{aligned}$$

□

This lemma implies that if we can force each term $V_s^{G,i,j}, M_s^{G,i,j}$ to be small, then we will have achieved our desired goal of mean-conditioned moment multicalibration (Definition 24).

Observation 5. *Suppose a transcript π_T is such that for all $i \in [n], j \in [n']$ and $G \in \mathcal{G}$, we have that $|V_T^{G,i,j}|, |M_T^{G,i,j}| \leq \alpha T$. Then the predictions are (α, β, n, n')-mean-conditioned moment multicalibrated in the sense of Definition 24 for $\beta = (k+1)\alpha + \frac{k}{2n}$.*

Remark 11. *Note that with this parametrization, we can take α as small as we like relative to n, and by choosing an appropriately large value of n, we can take $\beta = (k+1)\alpha + \frac{k}{2n}$ as small as we like relative to n'.*

As before, we define a surrogate loss function at each round s.

Definition 33 (Surrogate Loss)**.** *Fixing a transcript $\pi_s \in \Pi^*$ and a parameter $\eta \in [0, \frac{1}{2}]$, define:*

$$L_s(\pi_s) = \sum_{\substack{G\in\mathcal{G},\\ i\in[n], j\in[n']}} \left(\exp(\eta V_s^{G,i,j}) + \exp(-\eta V_s^{G,i,j}) + \exp(\eta M_s^{G,i,j}) + \exp(-\eta M_s^{G,i,j})\right),$$

where V and M are functions of π_s as defined in Definition 32. When the transcript π_s is clear from context we will sometimes simply write L_s.

As before, our goal is to find a strategy for the learner that guarantees that our surrogate loss L_T remains small. Towards this end, we define $\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu}, \overline{m}^k)$ to be the expected increase in the surrogate loss function in the event that the adversary plays feature vector x_{s+1} *and* the learner predicts $(\overline{\mu}, \overline{m}^k)$. Here the expectation is over the only remaining source of randomness after the conditioning — the distribution over labels y_{s+1}, which for any adversary is defined once we fix π_s and x_{s+1}.

Definition 34 (Conditional Change in Surrogate Loss)**.**

$$\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu}, \overline{m}^k) = \mathop{\mathbb{E}}_{\tilde{y}_{s+1}}\left[\tilde{L}_{s+1} - L_s \middle| \pi_s, x_{s+1}, \overline{\mu}, \overline{m}^k\right].$$

We again show a simple bound on $\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu}, \overline{m}^k)$:

Lemma 39. *For any transcript $\pi_s \in \Pi^*$, any $x_{s+1} \in \mathcal{X}$, and any predictions $\overline{\mu}, \overline{m}^k \in [0,1]$ such that $\overline{\mu} \in B_n(i)$ and $\overline{m}^k \in B_{n'}(j)$ for some $i \in [n]$ and $j \in [n']$:*

$$\begin{aligned}\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu}, \overline{m}^k) &\leq \eta \left(\mathop{\mathbb{E}}_{\tilde{y}_{s+1}}[\tilde{y}_{s+1}] - \overline{\mu}\right) C_s^{\overline{\mu}, \overline{m}^k}(x_{s+1}) \\ &+ \eta \left(\mathop{\mathbb{E}}_{\tilde{y}} (\tilde{y}_{s+1} - \hat{\mu}_{\overline{\mu}})^k - \overline{m}^k\right) D_s^{\overline{\mu}, \overline{m}^k}(x_{s+1}) + 2\eta^2 L_s,\end{aligned}$$

where

$$C_s^{\overline{\mu}, \overline{m}^k}(x_{s+1}) = C_s^{i,j}(x_{s+1}) = \sum_{G \in \mathcal{G}(x_{s+1})} \exp(\eta V_s^{G,i,j}) - \exp(-\eta V_s^{G,i,j}), \tag{6.5}$$

$$D_s^{\overline{\mu}, \overline{m}^k}(x_{s+1}) = D_s^{i,j}(x_{s+1}) = \sum_{G \in \mathcal{G}(x_{s+1})} \exp(\eta M_s^{G,i,j}) - \exp(-\eta M_s^{G,i,j}). \tag{6.6}$$

For economy of notation, we will generally elide the dependence on x_{s+1} for the C and D quantities and simply write $C_s^{i,j}, D_s^{i,j}$ when the feature vector is clear from context.

Proof. To see this, observe that by definition:

$$\begin{aligned}
&\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu}, \overline{m}^k) \\
&= \mathop{\mathbb{E}}_{\tilde{y}_{s+1}}\Bigg[\sum_{\mathcal{G}(x_{s+1})} \underbrace{\exp(\eta V_s^{G,i,j})\left(\exp\left(\eta\left(\tilde{y}_{s+1} - \overline{\mu}\right)\right) - 1\right) + \exp(-\eta V_s^{G,i,j})\left(\exp\left(-\eta\left(\tilde{y}_{s+1} - \overline{\mu}\right)\right) - 1\right)}_{*} \\
&\quad + \underbrace{\exp(\eta M_s^{G,i,j})\exp\left(\eta\left(\left(\tilde{y}_{s+1} - \hat{\mu}_{\overline{\mu}}\right)^k - \overline{m}^k\right) - 1\right)}_{**} \\
&\quad + \underbrace{\exp(-\eta M_s^{G,i,j})\exp\left(-\eta\left(\left(\tilde{y}_{s+1} - \hat{\mu}_{\overline{\mu}}\right)^k - \overline{m}^k\right) - 1\right)}_{***}\Bigg].
\end{aligned}$$

Using the fact that for $0 < |x| < \frac{1}{2}$, $\exp(x) \leq 1 + x + 2x^2$, we have that

$$* \leq \exp(\eta V_s^{G,i,j})\left(\eta\left(y_{s+1} - \overline{\mu}\right) + 2\eta^2\right) + \exp(-\eta V_s^{G,i,j})\left(-\eta\left(y_{s+1} - \overline{\mu}\right) + 2\eta^2\right).$$

Similarly, we have

$$\begin{aligned}
** + *** \leq{}& \exp(\eta M_s^{G,i,j})\left(\eta\left(\left(y_{s+1} - \hat{\mu}_{\overline{\mu}}\right)^k - \overline{m}^k\right) + 2\eta^2\right) \\
&+ \exp(-\eta M_s^{G,i,j})\left(-\eta\left(\left(\tilde{y}_{s+1} - \hat{\mu}_{\overline{\mu}}\right)^k - \overline{m}^k\right) + 2\eta^2\right).
\end{aligned}$$

Now, using the linearity of expectation and distributing the outer expectation to each relevant term where $\tilde{y}_{s+1}$ appears, we get

$$\begin{aligned}
&\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu}, \overline{m}^k) \\
&\leq \sum_{\mathcal{G}(x_{s+1})} \exp(\eta V_s^{G,i,j})\left(\eta\left(\mathbb{E}[\tilde{y}_{s+1}] - \overline{\mu}\right) + 2\eta^2\right) + \exp(-\eta V_s^{G,i,j})\left(-\eta\left(\mathbb{E}[\tilde{y}_{s+1}] - \overline{\mu}\right) + 2\eta^2\right) \\
&\quad + \exp(\eta M_s^{G,i,j})\left(\eta\left(\mathbb{E}\left[\left(\tilde{y}_{s+1} - \hat{\mu}_{\overline{\mu}}\right)^k\right] - \overline{m}^k\right) + 2\eta^2\right) \\
&\quad + \exp(-\eta M_s^{G,i,j})\left(-\eta\left(\mathbb{E}\left[\left(\tilde{y}_{s+1} - \hat{\mu}_{\overline{\mu}}\right)^k\right] - \overline{m}^k\right) + 2\eta^2\right).
\end{aligned}$$

Collecting terms appropriately and observing that

$$\sum_{\mathcal{G}(x_{s+1})} \left(\exp(\eta V_s^{G,i,j}) + \exp(-\eta V_s^{G,i,j}) + \exp(\eta M_s^{G,i,j}) + \exp(-\eta M_s^{G,i,j})\right) \leq L_s,$$

we have the desired bound. □

As before, we proceed by defining a zero-sum game between the learner and the adversary and using the minimax theorem to conclude that the learner always has a strategy that guarantees a bounded per-round increase in surrogate loss. To satisfy the convexity and compactness requirements of the minimax theorem, we will again consider a game where the learner's pure strategy space is a finite subset of $\mathcal{P}_{(\text{mean,moment})}$. To this end, we define the following grids for any $r \in \mathbb{N}$ (n and n' are the coarseness parameters of our bucketings from above):

$$\begin{aligned} \mathcal{P}^{rn} &= \left\{0, \frac{1}{rn}, \frac{2}{rn}, \ldots, 1\right\}, \\ \mathcal{P}^{rn'} &= \left\{0, \frac{1}{rn'}, \frac{2}{rn'}, \ldots, 1\right\}. \end{aligned}$$

As in the previous section, the need to discretize is only for technical reasons, and our algorithm has no dependence — neither in runtime nor in its convergence rate — on the value of r that we choose, so we can imagine the discretization to be arbitrarily fine.

Lemma 40. *For any transcript $\pi_s \in \Pi^*$ and any $x_{s+1} \in \mathcal{X}$, there exists a distribution over predictions for the learner $Q_{s+1}^L \in \Delta(\mathcal{P}^{rn} \times \mathcal{P}^{rn'})$, such that regardless of the adversary's choice of distribution of y_{s+1} over $\Delta\mathcal{Y}$, we have that:*

$$\mathop{\mathbb{E}}_{(\overline{\mu}, \overline{m}^k) \sim Q_{s+1}^L} \left[\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu}, \overline{m}^k)\right] \leq L_s \left(\frac{\eta}{rn} + \frac{\eta}{rn'} + 2\eta^2\right).$$

Proof. Fix the transcript π_s and the feature vector x_{s+1}. As before, we define a zero-sum game played between the learner (the minimization player) and the adversary (the maximization player), where the objective function of the game equals the upper bound on $\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu}, \overline{m}^k)$ from Lemma 39. Then, we again show that for every strategy of the adversary (i.e. distribution over y), there exists a best response for the learner that guarantees the objective function of the game is small. Finally, we appeal to the minimax theorem to conclude that there always exists a strategy for the learner that guarantees small objective value against any strategy of the adversary.

More precisely, consider the following objective function for the game:

$$\begin{aligned} u((\overline{\mu}, \overline{m}^k), y) &= \eta\,(y - \overline{\mu})\, C_s^{\overline{\mu}, \overline{m}^k} + \eta \left((y - \hat{\mu}_{\overline{\mu}})^k - \overline{m}^k \right) D_s^{\overline{\mu}, \overline{m}^k} + 2\eta^2 L_s \\ &= \eta\,(y - \overline{\mu})\, C_s^{\overline{\mu}, \overline{m}^k} + \eta \left(\left(\sum_{\ell=0}^{k} \binom{k}{\ell} (-\hat{\mu}_{\overline{\mu}})^{k-\ell} y^\ell \right) - \overline{m}^k \right) D_s^{\overline{\mu}, \overline{m}^k} + 2\eta^2 L_s \end{aligned}$$

where the pure strategy space for the learner is $X_1 = \mathcal{P}^{rn} \times \mathcal{P}^{rn'}$ and that of the adversary is (a priori) the set of all distributions over $[0,1]$. However, we observe that the expected value of the objective for any label distribution over $[0,1]$ is linear in $\mathbb{E}[y], \ldots, \mathbb{E}[y^k]$. So the payoff for any mixed strategy of the adversary is determined only by the associated k terms: $\mathbb{E}[y], \ldots, \mathbb{E}[y^k]$.

With this observation in mind, we perform a change of variables and define a new game with an enlarged strategy space for the adversary. In the new game, the strategy space for the learner remains $\mathcal{Q}^L = \Delta(\mathcal{P}^{rn} \times \mathcal{P}^{rn'})$. The strategy space for the adversary becomes $\mathcal{Q}^A = [0,1]^k$, representing a choice for each of the values $\mathbb{E}[y], \ldots \mathbb{E}[y^k]$. Note that this strategy space for the adversary is unencumbered by the requirement that these chosen values actually correspond to any feasible label distribution over $[0,1]$. The objective function of the game is obtained by replacing each term $\mathbb{E}[y^\ell]$ from our previous objective function with ψ_ℓ:

$$u((\overline{\mu}, \overline{m}^k), \psi) = \quad \eta\,(\psi_1 - \overline{\mu})\, C_s^{\overline{\mu}, \overline{m}^k} + \eta \left(\left(\hat{\mu}_{\overline{\mu}}^k + \sum_{\ell=1}^{k} \binom{k}{\ell} (-\hat{\mu}_{\overline{\mu}})^{k-\ell} \psi_\ell \right) - \overline{m}^k \right) D_s^{\overline{\mu}, \overline{m}^k} + 2\eta^2 L_s.$$

As we have noted, in the original game, the set of achievable moments $\mathbb{E}[y], \ldots, \mathbb{E}[y^k]$ is a strict subset of $[0,1]^k$. However, enlarging the strategy space of the maximization player can only increase the ($\max\min$) value of the game, so the upper bound we are about to prove on the game value against this more powerful adversary also applies to the adversary who is implicitly choosing moments $\mathbb{E}[y], \ldots, \mathbb{E}[y^k]$ via some distribution over $[0,1]$.

Note that u thus defined is linear in both players' strategies, and the strategy spaces for both players $\mathcal{Q}^L$ and $\mathcal{Q}^A$ are compact and convex. Hence, Sion's minimax theorem (Theorem 29) applies to this game. We now establish (a bound on) the value of this game. Observe that for any strategy of the adversary, the learner can pick $\overline{\mu} \in \mathcal{P}^{rn}$ as close as possible to ψ_1, and then pick $\overline{m}^k \in \mathcal{P}^{rn'}$ as close as possible to $\hat{\mu}_{\overline{\mu}}^k + \sum_{\ell=1}^k \binom{k}{\ell}(-\hat{\mu}_{\overline{\mu}})^{k-\ell}\psi_\ell$. Therefore, since $C_s^{\overline{\mu},\overline{m}^k}, D_s^{\overline{\mu},\overline{m}^k} \leq L_s$ by definition, we have that:

$$\forall \psi \in [0,1]^k, \exists (\overline{\mu}, \overline{m}^k) \in (\mathcal{P}^{rn} \times \mathcal{P}^{rn'}) \text{ s.t. } u((\overline{\mu}, \overline{m}^k), \psi) \leq L_s \left(\frac{\eta}{rn} + \frac{\eta}{rn'} + 2\eta^2 \right).$$

We can now apply the minimax theorem (Theorem 29) to conclude that there exists a fixed distribution $Q_{s+1}^L \in \mathcal{Q}^L$ for the learner that guarantees objective value that is at most the above bound for every choice of the adversary, i.e.

$$\exists Q_{s+1}^L \in \mathcal{Q}^L \text{ s.t. } \forall \psi \in [0,1]^k : u(Q_{s+1}^L, \psi) \leq L_s \left(\frac{\eta}{rn} + \frac{\eta}{rn'} + 2\eta^2 \right),$$

as desired. □

Corollary 7. *For every $s \in [T]$, $\pi_s \in \Pi^*$, $x_{s+1} \in \mathcal{X}$ (which fixes L_s and Q_{s+1}^L), and every adversary (which fixes a distribution over $\mathcal{Y}$):*

$$\mathop{\mathbb{E}}_{Q_{s+1}^L}[\tilde{L}_{s+1}|\pi_s] = L_s + \mathop{\mathbb{E}}_{Q_{s+1}^L}[\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu}, \overline{m}^k)|\pi_s] \leq L_s \left(1 + \frac{\eta}{rn} + \frac{\eta}{rn'} + 2\eta^2 \right).$$

Lemma 40 defines (existentially) an algorithm that the learner can use to make predictions—Algorithm 18. We will now show that Algorithm 18 (if we could compute the distributions

Q_t^L) results in mean-conditioned moment multicalibrated predictions. In Section 6.5.3 we show how to compute Q_t^L.

Algorithm 18: A Generic Mean Moment Multicalibrator

```
for $t = 1, \ldots, T$ do
    Observe $x_t$. Given $\pi_{t-1}$ and $x_t$, let $Q_t^L \in \Delta(\mathcal{P}^{rn} \times \mathcal{P}^{rn'})$ be the distribution over
    predictions whose existence is established in Lemma 40. Sample $\overline{\mu}, \overline{m}^k \sim Q_t^L$
    and predict $(\overline{\mu}_t, \overline{m}_t^k) = (\overline{\mu}, \overline{m}^k)$.
end
```

We are now ready to bound our multicalibration error. The results that follow mirror the structure of Section 6.4.2: essentially, we apply Theorem 30 to the surrogate loss function of this section. As a straightforward consequence of Corollary 7 and the first part of Theorem 30, we have the following result.

Corollary 8. *Against any adversary, Algorithm 18 instantiated with discretization parameter r results in surrogate loss satisfying:*

$$\mathop{\mathbb{E}}_{\tilde{\pi}_T}[\tilde{L}_T] \leq 4|\mathcal{G}|n \cdot n' \cdot \exp\left(\frac{T\eta}{rn} + \frac{T\eta}{rn'} + 2T\eta^2\right).$$

Proof. Note that the first part of Theorem 30 applies in this case to the process L, with $L_0 = 4|G|n \cdot n'$ and $c = \frac{1}{rn} + \frac{1}{rn'}$. The bound follows by plugging these values into (6.3). □

Next, we can convert this into a bound on Algorithm 16's expected calibration error, using Theorem 30. The proof mirrors the argument in Section 6.4 and can be found in the Appendix.

Theorem 34. *When Algorithm 18 is run using bucketing coarseness parameters n and n', discretization parameter $r \in \mathbb{N}$, and $\eta = \sqrt{\frac{\ln(4|\mathcal{G}|n \cdot n')}{2T}} \in (0, 1/2)$, then against any adversary, its sequence of mean-moment predictions is (α, β, n, n')-mean-conditioned moment* *multicalibrated with respect to $\mathcal{G}$, where $\beta = (k+1)\alpha + \frac{k}{2n}$ and:*

$$\mathbb{E}[\alpha] \leq \frac{1}{rn} + \frac{1}{rn'} + 2\sqrt{\frac{2\ln(4|\mathcal{G}|n \cdot n')}{T}}.$$

For $r = \frac{\sqrt{T}(n+n')}{\varepsilon n \cdot n' \cdot \sqrt{2\ln(4|\mathcal{G}|n \cdot n')}}$, this gives:

$$\mathbb{E}[\alpha] \leq (2+\varepsilon)\sqrt{\frac{2}{T}\ln\left(4|\mathcal{G}|n \cdot n'\right)}.$$

Here the expectation is taken over the randomness of the transcript π_T.

We can similarly use the second part of Theorem 30 to prove a high probability bound on the multicalibration error of Algorithm 18. The proof is in the Appendix.

Theorem 35. *When Algorithm 18 is run using bucketing coarseness parameters n and n', discretization $r \in \mathbb{N}$ and $\eta = \sqrt{\frac{\ln(4|\mathcal{G}|n \cdot n')}{2T}} \in (0, 1/2)$, then against any adversary, with probability $1-\lambda$ over the randomness of the transcript, its sequence of predictions is (α, β, n, n')-mean-conditioned moment multicalibrated with respect to $\mathcal{G}$ for $\beta = (k+1)\alpha + \frac{k}{2n}$ and:*

$$\alpha \leq \frac{1}{rn} + \frac{1}{rn'} + 4\sqrt{\frac{2}{T}\ln\left(\frac{4|\mathcal{G}|n \cdot n'}{\lambda}\right)}.$$

For $r = \frac{\sqrt{T}(n+n')}{\epsilon n \cdot n' \sqrt{2\ln(4|\mathcal{G}|n \cdot n'/\lambda)}}$, this gives:

$$\alpha \leq (4+\epsilon)\sqrt{\frac{2}{T}\ln\left(\frac{4|\mathcal{G}|n \cdot n'}{\lambda}\right)}.$$

6.5.3. Deriving an Efficient Algorithm via Equilibrium Computation

Previously, we derived Algorithm 18 and proved that it results in mean-conditioned moment multicalibrated predictions. But Algorithm 18 is not explicitly defined, as it relies on the distributions Q_t^L whose existence we showed in Lemma 40 but which we did not explicitly construct. In this section, we show how to efficiently solve for this distribution Q_t^L using a linear program with $4n \cdot n'$ variables and $2^k + 1$ constraints. If k is a small constant (e.g. $k = 2$ for variance multicalibration), then this linear program can be explicitly described and solved. But even when k is too large to enumerate all 2^k constraints, we show that there is a separation oracle that runs in time $O(k)$, allowing us to efficiently solve this linear program (i.e. in time polynomial in $n, n', T, |\mathcal{G}|$, and k) using the Ellipsoid algorithm.

Recall that in our simplified game, the learner has pure strategies $(\overline{\mu}, \overline{m}^k) \in \mathcal{P}^{rn} \times \mathcal{P}^{rn'}$, and the adversary has strategy space $\mathcal{Q}^A = [0,1]^k$. Since the objective function is linear in the adversary's action ψ, we can view this as the set of mixed strategies over the 2^k pure strategies $\psi \in \{0,1\}^k$. We recall the objective function:

$$u((\overline{\mu}, \overline{m}^k), \psi) = \quad \eta\left(\psi_1 - \overline{\mu}\right) C_s^{\overline{\mu}, \overline{m}^k} + \eta \left(\left(\hat{\mu}_{\overline{\mu}}^k + \sum_{\ell=1}^{k} \binom{k}{\ell} (-\hat{\mu}_{\overline{\mu}})^{k-\ell} \psi_\ell \right) - \overline{m}^k \right) D_s^{\overline{\mu}, \overline{m}^k} + 2\eta^2 L_s.$$

Since the equilibrium structure stays the same under positive affine transformations of the objective function, for the purposes of computing equilibria, we may redefine the objective function to be:

$$u((\overline{\mu}, \overline{m}^k), \psi) = \quad \left(\psi_1 - \overline{\mu}\right) C_s^{\overline{\mu}, \overline{m}^k} + \left(\left(\hat{\mu}_{\overline{\mu}}^k + \sum_{\ell=1}^{k} \binom{k}{\ell} (-\hat{\mu}_{\overline{\mu}})^{k-\ell} \psi_\ell \right) - \overline{m}^k \right) D_s^{\overline{\mu}, \overline{m}^k}. \qquad (6.7)$$

The specific values of $C_s^{\overline{\mu}, \overline{m}^k}$, $\hat{\mu}_{\overline{\mu}}$ and $D_s^{\overline{\mu}, \overline{m}^k}$ do not matter for the analysis that follows—but what is relevant is that by definition, they are constant for any two $(\overline{\mu}, \overline{m}^k)$ and $(\overline{\mu}', \overline{m}^{k\prime})$ both in the same bucket — in other words, if $\exists i \in [n], j \in [n']$ such that $(\overline{\mu}, \overline{m}^k), (\overline{\mu}', \overline{m}^{k\prime}) \in B_{n,n'}(i,j)$. We wish to find a minimax strategy for the learner in this game, i.e. to find a solution to

$$\underset{Q^L \in \mathcal{Q}^L}{\operatorname{argmin}} \max_{Q^A \in \mathcal{Q}^A} u(Q^L, Q^A).$$

A priori, the learner has $r^2 n' n$ pure strategies (i.e. $|\mathcal{P}^{rn} \times \mathcal{P}^{rn'}| = r^2 n' n$), and a minimax strategy could potentially be supported over all of them (causing our algorithm to have running time depending on r). However, we prove that we can without loss of generality reduce the size of the learner's pure strategy space to $4n'n$ (Lemma 41), which will eliminate any running time dependence on r and allow us to choose as fine a discretization as we like. We also show in Appendix 6.B that the learner always has a minimax strategy that randomizes over a support of at most $k + 1$ actions. Thus, as with mean multicalibration, we need only make limited use of randomness (at least for k small).

We first reduce the space of "relevant" pure strategies for the learner — intuitively, points that are at—or just barely below—the boundary of a bucket:

$$\hat{\mathcal{P}}^{r,n} = \bigcup_{i \in [n-1]} \left\{ \frac{i-1}{n}, \frac{i}{n} - \frac{1}{rn} \right\} \bigcup \left\{ \frac{n-1}{n}, 1 \right\} \subset \mathcal{P}^{rn},$$
$$\hat{\mathcal{P}}^{r,n'} = \bigcup_{i \in [n'-1]} \left\{ \frac{i-1}{n'}, \frac{i}{n'} - \frac{1}{rn'} \right\} \bigcup \left\{ \frac{n'-1}{n'}, 1 \right\} \subset \mathcal{P}^{rn'}.$$

Given these sets, define $\hat{\mathcal{Q}}^L_{r,n,n'} \equiv \Delta\left(\hat{\mathcal{P}}^{r,n} \times \hat{\mathcal{P}}^{r,n'}\right) \subset \mathcal{Q}^L$.

Lemma 41. *In the game with objective function u as defined in (6.7), the value of the game is unaffected if the learner is restricted to mixed strategies in $\hat{\mathcal{Q}}^L_{r,n,n'}$, a set of distributions which in particular have support over at most $4nn'$ actions. In other words:*

$$\min_{Q^L \in \mathcal{Q}^L} \max_{Q^A \in \mathcal{Q}^A} u(Q^L, Q^A) = \min_{\hat{Q}^L \in \hat{\mathcal{Q}}^L_{r,n,n'}} \max_{Q^A \in \mathcal{Q}^A} u(\hat{Q}^L, Q^A).$$

Proof. Fix any strategy $Q^L \in \mathcal{Q}^L$. Since $\hat{\mathcal{Q}}^L_{r,n,n'} \subseteq \mathcal{Q}^L$, it is sufficient to show that there exists a strategy $\hat{Q}^L \in \hat{\mathcal{Q}}^L_{r,n,n'}$ such that:

$$\max_{Q^A \in \mathcal{Q}^A} u(Q^L, Q^A) \geq \max_{Q^A \in \mathcal{Q}^A} u(\hat{Q}^L, Q^A).$$

To see this, first observe that we can regroup terms in the objective function (6.7) and write it as:

$$u((\overline{\mu}, \overline{m}^k), \psi) = -\overline{\mu} C_s^{\overline{\mu}, \overline{m}^k} + \hat{\mu}_{\overline{\mu}}^k D_s^{\overline{\mu}, \overline{m}^k} - \overline{m}^k D_s^{\overline{\mu}, \overline{m}^k} + \sum_{\ell=1}^{k} \psi_\ell F_\ell^{\overline{\mu}, \overline{m}^k} \tag{6.8}$$

$$\text{where } F_1^{\overline{\mu}, \overline{m}^k} = C_s^{\overline{\mu}, \overline{m}^k} - k \hat{\mu}_{\overline{\mu}}^{k-1} C_s^{\overline{\mu}, \overline{m}^k}, \tag{6.9}$$

$$\forall \ell > 1, \ell \in [n] : F_\ell^{\overline{\mu}, \overline{m}^k} = \binom{k}{\ell} (-\hat{\mu}_{\overline{\mu}})^{k-\ell} D_s^{\overline{\mu}, \overline{m}^k}. \tag{6.10}$$

Further, by definition for any $\overline{\mu}, \overline{\mu}' \in B_n(i)$ for some $i \in [n]$ and $\overline{m}^k, \overline{m}^{k\prime} \in B_{n'}(j)$, we have, for $X = C, D$,

$$X_s^{\overline{\mu}, \overline{m}^k} = X_s^{\overline{\mu}', \overline{m}^{k\prime}} = X_s^{i,j},$$
$$\hat{\mu}_{\overline{\mu}} = \hat{\mu}_{\overline{\mu}'},$$

and therefore this equality holds for $X = F$ as well. Against a given strategy Q^L for the learner, the adversary' payoff from pure strategy ψ is:

$$u(Q^L, \psi) = \sum_{(\overline{\mu}, \overline{m}^k)} Q^L(\overline{\mu}, \overline{m}^k) \left(-\overline{\mu} C_s^{\overline{\mu}, \overline{m}^k} + \hat{\mu}_{\overline{\mu}}^k D_s^{\overline{\mu}, \overline{m}^k} - \overline{m}^k D_s^{\overline{\mu}, \overline{m}^k} + \sum_{\ell=1}^{k} \psi_\ell F_\ell^{\overline{\mu}, \overline{m}^k} \right),$$

which, given the previous fact about F, can be rewritten as

$$u(Q^L, \psi) = \underbrace{\sum_{(\overline{\mu}, \overline{m}^k)} Q^L(\overline{\mu}, \overline{m}^k) \left(-\overline{\mu} C_s^{\overline{\mu}, \overline{m}^k} + \hat{\mu}_{\overline{\mu}}^k D_s^{\overline{\mu}, \overline{m}^k} - \overline{m}^k D_s^{\overline{\mu}, \overline{m}^k} \right)}_{(*)}$$
$$+ \underbrace{\sum_{\ell=1}^{k} \psi_\ell \sum_{\substack{i \in [n], \\ j \in [n']}} F_\ell^{i,j} \left(\sum_{(\overline{\mu}, \overline{m}^k) \in B(i,j)} Q^L(\overline{\mu}, \overline{m}^k) \right)}_{(**)}.$$

Observe that term $(*)$ is independent of ψ. Therefore, fixing a Q^L, it is equivalent for the adversary to maximize $(**)$. By observation, for any mixed strategy of the learner Q^L, the adversary's incentives are only affected through the induced distribution over buckets.

So, given Q^L, the best response of the adversary is preserved for any other strategy $\hat{Q}^L$ that maintains the same mass on each bucket, i.e. for all $i \in [n]$ and $j \in [n']$,

$$\sum_{(\overline{\mu},\overline{m}^k)\in B(i,j)} \left(Q^L(\overline{\mu},\overline{m}^k) - \hat{Q}^L(\overline{\mu},\overline{m}^k)\right) = 0.$$

Consider the learner's problem of minimizing the objective value among strategies of this form, i.e. preserving the mass on each bucket. This reduces to solving, for each $i \in [n], j \in [n']$, the optimization problem

$$\begin{aligned} \min_{\hat{Q}^L \geq 0} \quad & \sum_{(\overline{\mu},\overline{m}^k)\in B(i,j)} \hat{Q}^L(\overline{\mu},\overline{m}^k)\left(-\overline{\mu}C_s^{i,j} + \hat{\mu}_i^k D_s^{i,j} - \overline{m}^k D_s^{i,j}\right) \\ \text{s.t.} \quad & \sum_{(\overline{\mu},\overline{m}^k)\in B(i,j)} \left(Q^L(\overline{\mu},\overline{m}^k) - \hat{Q}^L(\overline{\mu},\overline{m}^k)\right) = 0. \end{aligned}$$

Within a bucket, the coefficients $\left(-\overline{\mu}C_s^{i,j} + \hat{\mu}_i^k D_s^{i,j} - \overline{m}^k D_s^{i,j}\right)$ are linear in $\overline{\mu}, \overline{m}^k$ and therefore there must exist a solution that puts all mass $\sum_{(\overline{\mu},\overline{m}^k)\in B(i,j)} Q^L(\overline{\mu},\overline{m}^k)$ on an extreme point of the bucket. For example, if $i \in [n-1]$, $j \in [n'-1]$; all mass can be placed without loss of generality on one of the four points in $\left\{\frac{i-1}{n}, \frac{i}{n} - \frac{1}{rn}\right\} \times \left\{\frac{j-1}{n'}, \frac{j}{n} - \frac{1}{rn'}\right\}$. If $i = n$, the corresponding set is $\{\frac{n-1}{n}, 1\}$, and if $j = n'$, the corresponding set is $\{\frac{n'-1}{n'}, 1\}$. Moving all the mass in each bucket to the optimal corner point, we have that for any strategy Q^L of the learner, there exists $\hat{Q}^L \in \hat{\mathcal{Q}}^L_{r,n,n'}$ such that $\max_{Q^A \in \mathcal{Q}^A} u(Q^L, Q^A) \geq \max_{Q^A \in \mathcal{Q}^A} u(\hat{Q}^L, Q^A)$, as desired. This concludes the proof. □

The result is that to compute the equilibrium strategy for the learner, it suffices to solve:

$$\operatorname*{argmin}_{Q^L \in \hat{\mathcal{Q}}^L_{r,n,n'}} \max_{\psi \in \{0,1\}^k} u(Q^L, \psi).$$

We can directly express this as a linear program with $4nn'$ variables and $2^k + 1$ constraints — see Linear Program 6.5.1.

$$\begin{aligned} \min_{Q^L \in \hat{\mathcal{Q}}^L_{r,n,n'}} \quad & \gamma \text{ s.t.} \\ \forall \psi \in \{0,1\}^k : & u(Q^L, \psi) \le \gamma, \\ & \sum\nolimits_{(\overline{\mu}, \overline{m}^k) \in \hat{\mathcal{P}}^{r,n} \times \hat{\mathcal{P}}^{r,n'}} Q^L((\overline{\mu}, \overline{m}^k)) = 1, \\ \forall (\overline{\mu}, \overline{m}^k) \in \hat{\mathcal{P}}^{r,n} \times \hat{\mathcal{P}}^{r,n'} : & Q^L((\overline{\mu}, \overline{m}^k)) \ge 0. \end{aligned}$$

Figure 6.5.1: A Linear Program for Computing a Minimax Equilibrium Strategy for the Learner at Round t.

This is a linear program in $4nn' + 1$ variables, with $2^k + 1$ constraints. If k is a constant, this is a polynomially sized linear program that can be solved explicitly. If k is superconstant, we will see that we can still solve the linear program with the Ellipsoid algorithm, because we can efficiently find violated constraints.

```
Algorithm 19: Von Neumann's Mean Moment Multicalibrator
  Input: $\epsilon > 0$
  for $t = 1, \dots, T$ do
      Observe $x_t$ and compute $C^{\overline{\mu}, \overline{m}^k}_{t-1}(x_t), D^{\overline{\mu}, \overline{m}^k}_{t-1}(x_t), (F^{\overline{\mu}, \overline{m}^k}_{\ell, t-1}(x_t))_{\ell=1}^n$ for each
        $(\overline{\mu}, \overline{m}^k) \in \hat{\mathcal{P}}^{r,n} \times \hat{\mathcal{P}}^{r,n'}$ as in Equations (6.5, 6.6, 6.9, 6.10).
      Find an $\epsilon$-approximate solution to the linear program from Figure 6.5.1, to
        obtain solution $Q^L_t \in \hat{\mathcal{Q}}^L_{r,n,n'}$.
      Predict $(\overline{\mu}_t, \overline{m}^k_t) = (\overline{\mu}, \overline{m}^k)$ with probability $Q^L_t((\overline{\mu}, \overline{m}^k))$.
  end
```

We thus obtain the following theorem:

Theorem 36. *Algorithm 19 implements Algorithm 18. In particular, it obtains multivalidity guarantees arbitrarily close to those of Theorems 34 and 35. Namely, for any desired $\epsilon > 0$, we have the following.*

Choosing $\eta = \sqrt{\frac{\ln(4|\mathcal{G}|n \cdot n' + \epsilon)}{2T}} \in (0, 1/2)$*, against any adversary, over the randomness of the transcript, the sequence of mean-moment predictions produced by Algorithm 19 is* (α, β, n, n')*-mean-conditioned moment multicalibrated with respect to* $\mathcal{G}$ *where* $\beta = (k+1)\alpha + \frac{k}{2n}$ *and:*

$$\mathbb{E}[\alpha] \leq \frac{1}{rn} + \frac{1}{rn'} + 2\sqrt{\frac{2\ln(4|\mathcal{G}|n \cdot n' + \epsilon)}{T}}.$$

For $r = \frac{\sqrt{T}(n+n')}{\epsilon' n \cdot n' \cdot \sqrt{2\ln(4|\mathcal{G}|n \cdot n' + \epsilon)}}$*, this gives:*

$$\mathbb{E}[\alpha] \leq \left(2 + \epsilon'\right)\sqrt{\frac{2}{T}\ln\left(4|\mathcal{G}|n \cdot n' + \epsilon\right)}.$$

Moreover, choosing $\eta = \sqrt{\frac{\ln(4|\mathcal{G}|n \cdot n') + \epsilon T}{2T}} \in (0, 1/2)$*, with probability* $1 - \lambda$ *over the randomness of the transcript* π_T *we have*

$$\alpha \leq \frac{1}{rn} + \frac{1}{rn'} + 4\sqrt{\frac{2}{T}\ln\left(\frac{4|\mathcal{G}|n \cdot n'}{\lambda}\right) + 2\epsilon}.$$

For $r = \frac{(n+n')}{\epsilon' n \cdot n' \sqrt{\frac{2}{T}\ln(4|\mathcal{G}|n \cdot n'/\lambda) + 2\epsilon}}$*, this gives:*

$$\alpha \leq \left(4 + \epsilon'\right)\sqrt{\frac{2}{T}\ln\left(\frac{4|\mathcal{G}|n \cdot n'}{\lambda}\right) + 2\epsilon}.$$

The runtime of Algorithm 21 scales as $O(|\mathcal{G}|)$ *with the total number of groups* $|\mathcal{G}|$*, and is polynomial in* n, n', T, k*, and* $\log(\frac{1}{\epsilon})$ *(and is independent of* r*).*

Remark 12. *As before, if* $|\mathcal{G}(x_t)|$ *is efficiently enumerable, then the running time dependence on* $|\mathcal{G}|$ *can be replaced with a dependence on* $|\mathcal{G}(x_t)|$*.*

Proof. First consider the running time of the algorithm. The quantities $C_{t-1}^{\overline{\mu},\overline{m}^k}(x_t), D_{t-1}^{\overline{\mu},\overline{m}^k}(x_t)$, and $F_{\ell,t-1}^{\overline{\mu},\overline{m}^k}(x_t)$ are simple sums, which can be computed in time linear in $|\mathcal{G}|$ (or $|\mathcal{G}(x_t)|$ if it is efficiently enumerable) and T. The linear program has $4nn' + 1$ variables, and $2^k + 1$

constraints. If k is a constant, this is polynomially sized. Now consider the case in which k is large. In this case we will solve the linear program by applying the Ellipsoid algorithm to its “rational” modification (see below). The runtime of this approach is polynomial under several well-known conditions, which are given in the following theorem:

Theorem 37 ([84], Corollary 14.1a)**.** *For an optimization program of a linear objective with rational coefficients over a rational polyhedron P in $\mathbb{R}^q$ for which we are given a separation oracle, the Ellipsoid algorithm solves it exactly in time polynomial in the following parameters: the number of variables q, the largest bit complexity ϕ of any linear inequality defining P, the bit complexity c of the objective function, and the runtime of a separation oracle.*

Linear Program 6.5.1 has finitely many constraints so its feasible region is a polyhedron. However, exponential terms in the coefficients of the constraints associated with the adversarial best-responses (which are due to our definition of the soft-max surrogate loss) prevent it from being *rational*. To fix this, we only keep $O(\log \frac{1}{\epsilon})$ bits of precision after the integer part of every coefficient of LP 6.5.1, resulting in a new LP whose coefficients are all rational and within $\pm\frac{\epsilon}{2}$ from their original values in LP 6.5.1. The new LP indeed has a rational polyhedron as its feasible region. We now pause to see that solving the rational LP achieves value within ϵ of the desired optimum of LP 6.5.1. This is shown more generally in the following technical lemma, which we will reuse in Section 6.6.3; its proof is deferred to the Appendix.

Lemma 42. *Consider a linear program of the following form, with variables $x \in \mathbb{R}^m$, $\gamma \in \mathbb{R}$ for some m:*

$$\textit{Minimize } \gamma, \quad \textit{subject to:} \quad Ax \leq \gamma \boldsymbol{1}^m, x \cdot \boldsymbol{1}^m = 1, x \geq 0.$$

Here, $\boldsymbol{1}^m \in \mathbb{R}^m$ is the all-ones vector, and $A = (a_{ji})$ is a finite matrix with real entries.

Take any $\epsilon > 0$. Modify the above linear program by replacing matrix A with matrix $\tilde{A} = (\tilde{a}_{ji})$, where each $\tilde{a}_{ji}$ is a rational number within $\pm\frac{\epsilon}{2}$ from a_{ji}, obtained by truncating a_{ji} to $O(\log\frac{1}{\epsilon})$ bits of precision. Then, any optimal solution $(x^{*,r}, \gamma^{*,r})$ of the resulting rational linear program is an ϵ-approximately optimal feasible solution of the original linear program.

Linear Program 6.5.1 is of the type given in Lemma 42, so we have that solving the rational LP gives the desired ϵ-approximation to the optimum of Linear Program 6.5.1. Now we verify that all linear constraints of the rational version of LP 6.5.1 have polynomial bit complexity. Recall that the left side of any constraint bounding the objective function can be written as:

$$u(Q^L, \psi) = \underbrace{\sum_{(\overline{\mu}, \overline{m}^k)} Q^L(\overline{\mu}, \overline{m}^k) \left(-\overline{\mu} C_{t-1}^{\overline{\mu}, \overline{m}^k} + \hat{\mu}_{\overline{\mu}}^k D_{t-1}^{\overline{\mu}, \overline{m}^k} - \overline{m}^k D_{t-1}^{\overline{\mu}, \overline{m}^k} \right)}_{(*)}$$
$$+ \underbrace{\sum_{\ell=1}^{k} \psi_\ell \sum_{\substack{i \in [n], \\ j \in [n']}} F_\ell^{i,j} \left(\sum_{(\overline{\mu}, \overline{m}^k) \in B(i,j)} Q^L(\overline{\mu}, \overline{m}^k) \right)}_{(**)}.$$

There are $4nn' + 1$ variables. We can bound the coefficient in which any $Q^L(\overline{\mu}, \overline{m}^k)$ appears in (*) by:

$$\begin{aligned}
&\max_{\overline{\mu}, \overline{m}^k} \sum_G \exp(\eta V_{t-1}^{G,i,j}) - \exp(-\eta V_{t-1}^{G,i,j}) + 2\left(\exp(\eta M_{t-1}^{G,i,j}) - \exp(-\eta M_{t-1}^{G,i,j})\right) \\
&\leq |\mathcal{G}|(6\exp(\eta 2T)) \\
&\leq 6|\mathcal{G}|\exp(2T).
\end{aligned}$$

The coefficient of any variable $Q^L(\overline{\mu}, \overline{m}^k)$ in (**) is at most:

$$\sum_{\ell=1}^{k} \psi_\ell \sum_{\substack{i \in [n], \\ j \in [n']}} F_\ell^{i,j} \leq k \cdot (nn') \cdot \max_{i,j} \left\{ 2^k \left(\sum_G 2\exp(\eta M_T^{G,i,j}) \right) \right\} \leq 2^{k+1} k |\mathcal{G}| nn' \cdot \exp(2T).$$

Recalling that we are also keeping $O(\log \frac{1}{\epsilon})$ bits of precision for each coefficient, it follows that the maximum bit complexity of any constraint is bounded by

$$O\left(2 \cdot 4nn' \cdot \left(\log\left(2^{k+1}k|\mathcal{G}|nn' \cdot \exp(2T)\right) + \log\frac{1}{\epsilon}\right)\right) = \text{poly}\left(n, n', |\mathcal{G}|, T, k, \log\frac{1}{\epsilon}\right).$$

Of course, the objective value, which is simply γ, also has polynomial bit complexity.

Next, we describe an efficient separation oracle for the LP. Consider a candidate solution (Q^L, γ). The constraint requiring that Q^L be a probability distribution can be checked explicitly. Thus, it remains to either find a violated constraint corresponding to some pure strategy $\psi \in \{0,1\}^k$ of the adversary, or to assert that none exists. But this reduces to the problem of finding the most violated such constraint, which corresponds to the adversary's pure best response problem. Note that only the (**) term of the objective function (see the formula above) depends on the adversary's action. Thus, the best response problem of the adversary corresponds to finding

$$\psi^* = \arg\max_{\psi \in \{0,1\}^k} \sum_{\ell=1}^{k} \psi_\ell \sum_{i \in [n], j \in [n']} F_\ell^{i,j} \sum_{(\overline{\mu}, \overline{m}^k) \in B(i,j)} Q^L(\overline{\mu}, \overline{m}^k).$$

The best response for the adversary given a fixed distribution Q^L can be computed by setting each coordinate $\ell \in [k]$ independently to be either 0 or 1: namely, $\psi_\ell = 1$ if $\sum_{\substack{i \in [n],\\ j \in [n']}} F_\ell^{i,j} \left(\sum_{(\overline{\mu}, \overline{m}^k) \in B(i,j)} Q^L(\overline{\mu}, \overline{m}^k)\right) \geq 0$ and $\psi_\ell = 0$ otherwise. This takes $O(k)$ iterations, at each of which the expression whose sign determines ψ_ℓ is computed in polynomial time. Once the adversary's best response has been computed, the oracle simply outputs the corresponding constraint if it is violated, and otherwise it asserts that the proposed solutions is feasible. Thus, we have a polynomial-time separation oracle for Linear Program 6.5.1.

This completes the proof that Linear Program 6.5.1 can be solved, at each round, to precision $\epsilon > 0$ in time polynomial in $n, n', \log|\mathcal{G}|, T, k, \log\frac{1}{\epsilon}$. The runtime of Algorithm 19 is therefore also $\text{poly}(n, n', |\mathcal{G}|, T, k, \log\frac{1}{\epsilon})$, where the dependence on $|\mathcal{G}|$ is $O(|\mathcal{G}|)$ — since at the beginning of each round t, we precompute the coefficients of the linear program in time linear in $|\mathcal{G}|$, and the Ellipsoid runs in time polynomial in $\log |\mathcal{G}|$.

Finally, we need to demonstrate that the claimed multivalidity guarantees (which are a function of the chosen $\epsilon > 0$) indeed hold. If we were exactly solving the linear program, this would be immediate from Lemma 41 and the fact that Linear Program 6.5.1 is directly solving for:

$$\underset{Q^L \in \hat{\mathcal{Q}}^L_{r,n,n'}}{\operatorname{argmin}} \max_{\psi \in \{0,1\}^k} u(Q^L, \psi).$$

We only need to verify that our approximate guarantees follow from approximately solving the linear program.

Lemma 43. *Algorithm 19 achieves the multivalidity guarantees specified in Theorem 36.*

The proof of this lemma involves repeating several calculations from Section 6.5.2 with an ϵ error term, and so is deferred to the Appendix. □

6.6. Online Multivalid Marginal Coverage

6.6.1. An Outline of Our Approach

In this section, we derive an online algorithm for supplying prediction intervals with a coverage target $1 - \delta$ that are multivalid with respect to some collection of groups $\mathcal{G}$. When $\mathcal{G} = \{\mathcal{X}\}$, this corresponds to giving simple marginal prediction intervals — a similar problem as solved by conformal prediction[24], but without requiring distributional assumptions. For richer classes $\mathcal{G}$, we obtain correspondingly stronger guarantees. We follow the same basic strategy that we developed in Section 6.4 for making multicalibrated mean predictions, with a couple of important deviations.

1. First, we observe that even in the distributional setting, it is *not* always possible to

[24] In fact, even with $\mathcal{G} = \{\mathcal{X}\}$ the guarantees are stronger than the marginal guarantees promised by conformal prediction techniques, because they remain valid even conditioning on the prediction. This is important and rules out trivial solutions, like predicting the full interval with probability $1 - \delta$ and an empty interval with probability δ.

provide prediction intervals that have coverage probability exactly $1 - \delta$. Consider, for example, the case in which the label distribution is a point mass. Then, any prediction interval will have coverage probability either 0 or 1 — in both cases, bounded away from the target $1 - \delta$. More generally, if we are giving prediction intervals with endpoints in some discrete set $\{0, 1/rn, \ldots, 1\}$, in order for there to exist prediction intervals with approximately the desired coverage probability in the distributional setting, the distribution must not be overly concentrated on any sub-interval of width $1/rn$. We define a sufficient smoothness condition (Definition 36) for appropriately tight prediction intervals to be guaranteed to exist in the distributional setting — a condition that becomes increasingly mild as we take our discretization parameter r to be larger. We then derive — existentially, using the minimax theorem — the existence of an online algorithm that gives prediction intervals that are multivalid at the desired coverage probability when played against an adversary who is constrained at every round to play smooth label distributions. We observe (Remark 14) that our smoothness condition is very mild, in the sense that we can *enforce it ourselves* by adding noise $U[-\epsilon, \epsilon]$ to the adversary's labels, rather than making assumptions about the adversary. When we do this, the intervals we obtain continue to have valid coverage if we widen both endpoints by ϵ.

2. To instantiate our algorithm, we again need to compute equilibrium strategies for an appropriately defined game for our learner to sample from. Unlike in the cases of mean and moment multicalibration, however, the equilibrium strategies in this case do not appear to have any nice structure. We can still derive an efficient algorithm, however, by solving a linear program at each round to compute an equilibrium of the corresponding game. Because we assume that our adversary plays label distributions that are appropriately smooth, the adversary has exponentially many pure strategies in this game, and so we cannot efficiently enumerate all of the constraints in our equilibrium computation program. Instead, we show that a simple greedy algorithm is able to implement a separation oracle, which allows us to solve the linear program

efficiently using the Ellipsoid algorithm.

6.6.2. An Existential Derivation of the Algorithm and Multicoverage Bounds

Our goal in this section is to derive an algorithm which at each round, makes predictions $(\bar{\ell}_t, \overline{u}_t) \in \mathcal{P}_{\text{interval}}$ that are multivalid with respect to some target coverage probability $1-\delta$.

Towards this end, we define the coverage error of a group G and interval (ℓ, u):

Definition 35. *Given a transcript $\pi_s = (x_t, (\bar{\ell}_t, \overline{u}_t), y_t)_{t=1}^{s}$, we define the coverage error for a group $G \in \mathcal{G}$ and bucket $(i,j) \in [n] \times [n]$ at time s to be:*

$$V_s^{G,(i,j)} = \sum_{t=1}^{s} \mathbb{1}[x_t \in G, (\bar{\ell}_t, \overline{u}_t) \in B_n(i,j)] \cdot v_\delta((\bar{\ell}_t, \overline{u}_t), y_t),$$

$$\textit{where } v_\delta((\ell, u), y) = \text{Cover}((\ell, u), y) - (1-\delta).$$

Just as before, our coverage error serves as a bound on our multicoverage error.

Observation 6. *Fix a transcript π_T. If for all $G \in \mathcal{G}$, and buckets $(i,j) \in [n] \times [n]$, we have that:*

$$\left| V_T^{G,(i,j)} \right| \leq \alpha T$$

then the corresponding sequence of prediction intervals are (α, n)-multivalid with respect to $\mathcal{G}$.

We now pause to observe that even in the easier distributional setting where data are drawn from a fixed distribution: $(x,y) \sim \mathcal{D}$ — there may not be any interval $(\ell, u) \in \mathcal{P}_{\text{interval}}$ that satisfies the desired target coverage value, i.e. that guarantees that $|\mathbb{E}_{(x,y)\sim\mathcal{D}}[v_\delta((\ell, u), y]|$ is small. Consider for example a label distribution that places all its mass on a single value $y = i \in [0,1]$. Then any interval (ℓ, u) covers the label with probability 1 or probability 0, which for $\delta \notin \{0,1\}$ is bounded away from our target coverage probability. Of course, if achieving the target coverage is impossible in the easier distributional setting, then it is also impossible in the more challenging online adversarial setting. With this in mind, we define a class of smooth distributions for which achieving (approximately) the target coverage is always possible for some interval (ℓ, u) defined over an appropriately finely discretized range:

$$\mathcal{P}^{rn}_{\text{interval}} = \{(i, j) \in \mathcal{P}_{\text{interval}} : i, j \in \mathcal{P}^{rn}\},$$

where as before, $\mathcal{P}^{rn}$ is the uniform grid on $[0, 1]$, $\{0, \frac{1}{rn}, \ldots, 1\}$. We show that we can similarly achieve (approximately) our target coverage goals in the online adversarial setting when the adversary is constrained to playing smooth distributions.

Definition 36. *A label distribution $Q \in \Delta\mathcal{Y}$ is* (ρ, rn)-smooth *if for any $0 \leq a \leq b \leq 1$ such that $|a - b| \leq \frac{1}{rn}$,*

$$\Pr_{y \sim Q}[y \in [a, b]] \leq \rho.$$

We say that a joint distribution $\mathcal{D} \in \Delta(\mathcal{X} \times \mathcal{Y})$ is (ρ, rn)-smooth *if for every $x \in \mathcal{X}$, the marginal label distribution conditional on x, $\mathcal{D}|_x$, is (ρ, rn)-smooth.*

Observation 7. *For any $\delta \in [0, 1]$ and any fixed (ρ, rn)-smooth label distribution Q, there always exists some interval $(\bar{\ell}, \bar{u}) \in \mathcal{P}^{rn}_{interval}$ such that $|\Pr_{y \sim Q}[\text{Cover}((\ell, u), y)] - (1 - \delta)| \leq \rho$.*

Remark 13. *The assumption of (ρ, rn)-smoothness becomes more mild for any ρ as $r \to \infty$. Just as for mean and moment multicalibration, in which our error bounds inevitably depend on the level of discretization r that we choose, here our error bounds will depend on the smoothness level ρ of the adversary's distributions at the discretization level r that we choose. Finally, observe that smoothness is an extremely mild condition in that we can enforce it ourselves if we so choose, rather than assuming that the adversary is constrained. We elaborate on this in Remark 14.*

Definition 37. *We write $\mathcal{Q}_{\rho,rn}$ for the set of all (ρ, rn) smooth distributions over $[0, 1]$. We write $\hat{\mathcal{Q}}_{\rho,rn}$ for the set of all (ρ, rn)-smooth distributions whose support belongs to the grid $\mathcal{P}^{rn} = \{0, \frac{1}{rn}, \ldots, 1\}$:*

$$\hat{\mathcal{Q}}_{\rho,rn} \equiv \Delta\mathcal{P}^{rn} \cap \mathcal{Q}_{\rho,rn}.$$

We will show (in Lemma 46) that when the learner is restricted to selecting intervals from $\mathcal{P}^{rn}_{\text{interval}}$, without loss of generality, rather than considering adversaries that play arbitrary distributions over $\mathcal{Q}_{\rho,rn}$, it suffices to consider adversaries that play discrete distributions from $\hat{\mathcal{Q}}_{\rho,rn}$, which will be more convenient for us.

To bound the maximum absolute value of our coverage errors across all groups and interval predictions, we again introduce the same style of surrogate loss function:

Definition 38 (Surrogate loss). *Fixing a transcript $\pi_s \in \Pi^*$ and a parameter $\eta \in (0, 1/2)$, define a surrogate coverage loss function at day s as:*

$$L_s(\pi_s) = \sum_{\substack{G \in \mathcal{G}, \\ (i,j) \in [n] \times [n]}} \left(\exp(\eta V_s^{G,(i,j)}) + \exp(-\eta V_s^{G,(i,j)}) \right),$$

where $V_s^{G,(i,j)}$ are implicitly functions of π_s. When the transcript is clear from context we will sometimes simply write L_s.

Once again, $0 < \eta < \frac{1}{2}$ is a parameter that we will set later.

As before, we proceed by bounding the conditional change in the surrogate loss function:

Definition 39 (Conditional Change in Surrogate Loss). *Fixing $\pi_s \in \Pi^*$, $x_{s+1} \in \mathcal{X}$ and an interval $(\ell, u) \in \mathcal{P}^{rn}_{interval}$, define the conditional change in surrogate loss to be:*

$$\Delta_{s+1}(\pi_s, x_{s+1}, (\overline{\ell}_{t+1}, \overline{u}_{t+1})) = \mathop{\mathbb{E}}_{\tilde{y}_{s+1}} [\tilde{L}_{s+1} - L_s | x_{s+1}, (\overline{\ell}_{s+1}, \overline{u}_{s+1}), \pi_s].$$

Lemma 44. *For every transcript $\pi_s \in \Pi^*$, every $x_{s+1} \in \mathcal{X}$, and every $(\overline{\ell}_{s+1}, \overline{u}_{s+1}) \in B_n(i,j)$ we have that:*

$$\Delta_{s+1}(\pi_s, x_{s+1}, (\overline{\ell}_{s+1}, \overline{u}_{s+1})) \leq \left(\eta (\mathop{\mathbb{E}}_{\tilde{y}_{s+1}} [v_\delta((\overline{\ell}_{s+1}, \overline{u}_{s+1}), \tilde{y}_{s+1})]) \right) C_s^{i,j}(x_{s+1}) + 2\eta^2 L_s,$$

where for each $i \leq j \in [n]$, we have defined

$$C_s^{i,j}(x_{s+1}) \equiv \sum_{\mathcal{G}(x_{s+1})} \exp(\eta V_s^{G,(i,j)}) - \exp(-\eta V_s^{G,(i,j)}).$$

When x_{s+1} is clear from context, for notational economy, we will elide it and simply write $C_s^{i,j}$.

As in Section 6.5, we defer proofs that mirror previous arguments to the Appendix.

Next, we abuse notation and write $V_s^{G,(\ell,u)}$ to denote $V_s^{G,(i,j)}$ for $i, j \in [n] \times [n]$ such that $(\ell, u) \in B_n(i, j)$. Given $(\ell, u) \in \mathcal{P}_{\text{interval}}$ such that $(\ell, u) \in B_n(i, j)$, we let $C_s^{\ell,u} \equiv C_s^{i,j}$, with the latter defined in the statement of Lemma 44. That is, fixing π_s and x_{s+1}, for any $(\ell, u) \in \mathcal{P}_{\text{interval}}$ such that $(\ell, u) \in B_n(i, j)$,

$$C_s^{\ell,u}(x_{s+1}) \equiv C_s^{i,j}(x_{s+1}) = \sum_{\mathcal{G}(x_{s+1})} \exp(\eta V_s^{G,(i,j)}) - \exp(-\eta V_s^{G,(i,j)}), \tag{6.11}$$

where in turn the V's are as defined in Definition 35.

Lemma 45 (Value of the Game)**.** *For any $x_{s+1} \in \mathcal{X}$, any adversary restricted to playing (ρ, rn)-smooth distributions, and any transcript $\pi_s \in \Pi^*$, there exists a distribution over predictions for the learner $Q_{s+1}^L \in \Delta\mathcal{P}_{interval}^{rn}$ which guarantees that:*

$$\mathop{\mathbb{E}}_{(\bar{\ell},\bar{u}) \sim Q_{s+1}^L} \left[\Delta_{s+1}(\pi_s, x_{s+1}, (\bar{\ell}_{s+1}, \bar{u}_{s+1}))\right] \leq L_s \left(\eta\rho + 2\eta^2\right).$$

Proof. We again proceed by defining a zero-sum game with objective function equal to the upper bound on $\Delta_{s+1}(\pi_s, x_{s+1}, (\bar{\ell}_{s+1}, \bar{u}_{s+1}))$ that we proved in Lemma 44:

$$u((\ell, u), y) = \eta \cdot v_\delta((\ell, u), y) \cdot C_s^{\ell,u} + 2\eta^2 L_s.$$

Here, the strategy space for the learner (the minimization player) is the set of all distributions over $\mathcal{P}_{\text{interval}}^{rn}$: $\mathcal{Q}^L = \Delta\mathcal{P}_{\text{interval}}^{rn}$. A priori, the strategy space for the adversary is $\mathcal{Q}_{\rho,rn}$

the set of all (ρ, rn)-smooth distributions, but we show that it suffices to take $\mathcal{Q}^A = \hat{\mathcal{Q}}_{\rho,rn}$, the set of all discrete (ρ, rn)-smooth distributions (i.e. restricting the adversary in this way does not change the value of the game).

Lemma 46. *For any strategy $Q^L \in \Delta\mathcal{P}^{rn}_{interval}$ for the learner, the adversary has a best response amongst the set of all (ρ, rn)-smooth distributions with support only over the discretization $\{0, 1/rn, \ldots, 1\}$. In other words, for any $Q^L \in \Delta\mathcal{P}^{rn}_{interval}$, there exists a $\hat{Q}^A \in \hat{\mathcal{Q}}_{\rho,rn}$ such that:*

$$\hat{Q}^A \in \underset{Q^A \in \mathcal{Q}_{\rho,rn}}{\operatorname{argmax}} \underset{\substack{(\ell,u)\sim Q^L,\\ y\sim Q^A}}{\mathbb{E}} [u((\ell, u), y)].$$

Proof. Fix any $Q^{A'} \in \operatorname{argmax}_{Q^A \in \mathcal{Q}_{\rho,rn}} \mathbb{E}_{(\ell,u)\sim Q^L, y\sim Q^A}[u((\ell, u), y)]$ — i.e. an arbitrary (ρ, rn)-smooth best response for the maximization player. We will construct a discrete (ρ, rn)-smooth $\hat{Q}^A \in \hat{\mathcal{Q}}_{\rho,rn}$ that obtains the same objective value, as follows. For each $\frac{i}{rn} \in \{0, 1/rn, \ldots, 1\}$, let:

$$\Pr_{y\sim Q^A}\left[y = \tfrac{i}{rn}\right] = \Pr_{y\sim Q^{A'}}\left[y \in \left[\frac{i}{rn}, \frac{i+1}{rn}\right)\right].$$

Observe first by construction that Q^A is a discrete probability distribution (because $Q^{A'}$ is a probability distribution over $[0, 1]$, and the set of intervals $[\frac{i}{rn}, \frac{i+1}{rn})$ partition the unit interval), and that Q^A is (ρ, rn)-smooth because $Q^{A'}$ is (ρ, rn)-smooth — we have $\Pr_{y\sim Q^A}[y = \frac{i}{rn}] \leq \rho$ for all i. Finally observe that (by definition) for any $(\ell, u) \in \mathcal{P}^{rn}_{\text{interval}}$, $\ell, u \in \{0, 1/rn, \ldots, 1\}$.

Therefore, we have that for any $(\ell, u) \in \mathcal{P}^{rn}_{\text{interval}}$, any $i \in \{0, 1, \ldots, n\}$, and any $y, y' \in \left[\frac{i}{rn}, \frac{i+1}{rn}\right)$, $u((\ell, u), y) = u((\ell, u), y')$. To see this, note that $y \geq \ell$ if and only if $y' \geq \ell$, and $y < u$ if and only if $y' < u$. Since $v_\delta((\ell, u), y)$ is a function only of the indicators of the event that $\ell \leq y < u$, this proves the claim. □

Recall (from Observation 7) that for any (ρ, rn)-smooth label distribution Q^A, there exists an interval $(\ell, u) \in \mathcal{P}^{rn}_{\text{interval}}$ such that $|\Pr_{y \sim Q^A}[y \in [\ell, u)] - (1-\delta)| \leq \rho$, meaning there exists $(\overline{\ell}, \overline{u})$ such that $\mathbb{E}_{\tilde{y}_{s+1}}[v_\delta((\overline{\ell}, \overline{u}), \tilde{y}_{s+1})] \leq \rho$. We can thus bound the value of the game we have defined as follows:

$$\begin{aligned}
&\max_{Q^A \in \hat{\mathcal{Q}}_{\rho,rn}} \min_{(\ell,u)\in\mathcal{P}^{rn}_{\text{interval}}} \mathop{\mathbb{E}}_{y \sim Q^A} [u(\ell,u), y] \\
&\leq \sum_{\mathcal{G}(x_{s+1})} \exp(\eta V_s^{G,(\ell,u)})\,(\eta\rho) + \exp(-\eta V_s^{G,(\ell,u)})\,(\eta\rho) + 2\eta^2 L_s, \\
&\leq L_s(\eta\rho + 2\eta^2).
\end{aligned}$$

It is easy to verify that $\Delta\mathcal{P}^{rn}_{\text{interval}}$ and $\hat{\mathcal{Q}}_{\rho,rn}$ are both compact sets (closed and bounded in a finite dimensional Euclidean space) and convex. The lemma then follows by applying the minimax theorem (Theorem 29). □

Corollary 9. *For every $s \in [T]$, $\pi_s \in \Pi^*$, and $x_{s+1} \in \mathcal{X}$ (which fixes L_s and Q^L_{s+1}), and any distribution over $\mathcal{Y}$:*

$$\mathop{\mathbb{E}}_{(\ell,u)\sim Q^L_{s+1}} [\tilde{L}_{s+1}|\pi_s] \leq L_s + \mathop{\mathbb{E}}_{(\overline{\ell},\overline{u})\sim Q^L_{s+1}} \left[\Delta_{s+1}(\pi_s, x_{s+1}, (\overline{\ell}_{s+1}, \overline{u}_{s+1}))\right] < L_s\left(1 + \eta\rho + 2\eta^2\right).$$

As with mean multicalibration, Lemma 45 defines (existentially) an algorithm that the learner can use to make predictions — Algorithm 20. We will now show that Algorithm 20 (if we could compute the distributions Q^L_t) results in multivalid prediction intervals.

Algorithm 20: A Generic Multivalid Predictor

```
for $t = 1, \ldots, T$ do
    Observe $x_t$. Given $\pi_{t-1}$ and $x_t$, let $Q^L_t \in \Delta\mathcal{P}^{rn}_{\text{interval}}$ be the distribution over
     prediction intervals whose existence is established in Lemma 45.
    Sample $(\overline{\ell}, \overline{u}) \sim Q^L_t$ and predict $(\overline{\ell}_t, \overline{u}_t) = (\overline{\ell}, \overline{u})$
end
```

Lemma 47. *Against any adversary who is constrained to playing (ρ, rn)-smooth distribu-*

tions, Algorithm 20 results in surrogate loss satisfying:

$$\mathop{\mathbb{E}}_{\tilde{\pi}_T}[\tilde{L}_T] \leq 2|\mathcal{G}|n^2 \exp\left(T\eta\rho + 2T\eta^2\right).$$

Proof. Using Corollary 9, the first part of Theorem 30 applies in this case to the process L with $L_0 = 2|G|n^2$ and $c = \rho$. The bound follows by plugging these values into (6.3). □

Finally, we can calculate a bound on our expected multivalidity error. The proof (which mirrors similar claims in previous sections) is in the Appendix.

Theorem 38. *When Algorithm 20 is run using n buckets, discretization parameter r and $\eta = \sqrt{\frac{\ln(2|\mathcal{G}|n^2)}{2T}} \in (0, 1/2)$, then against any adversary constrained to playing (ρ, rn)-smooth distributions, its sequence of interval predictions is α-multivalid with respect to $\mathcal{G}$ in expectation over the randomness of the transcript π_T, where:*

$$\mathbb{E}[\alpha] \leq \rho + 2\sqrt{\frac{2\ln(2|\mathcal{G}|n^2)}{T}}.$$

We can also use the second part of Theorem 30 to prove a high probability bound on the multicalibration error of Algorithm 20. The proof is in the Appendix.

Theorem 39. *When Algorithm 20 is run using n buckets, discretization parameter r and $\eta = \sqrt{\frac{\ln(2|\mathcal{G}|n^2)}{2T}} \in (0, 1/2)$, then against any adversary who is constrained to playing (ρ, rn)-smooth distributions, its sequence of interval predictions is α-multivalid with respect to $\mathcal{G}$ with probability $1 - \lambda$ over the randomness of the transcript π_T:*

$$\alpha \leq \rho + 4\sqrt{\frac{2}{T}\ln\left(\frac{2|\mathcal{G}|n^2}{\lambda}\right)}.$$

Remark 14. *The hypothesis of our theorems has an assumption: that the adversary is restricted to playing (ρ, rn)-smooth distributions. This may be reasonable if we are not in* *a truly adversarial setting, and are simply concerned with unknown distribution shift. But what if we are truly in an adversarial environment? It turns out that in order to have a useful algorithm, we need not make* any *assumptions on the adversary at all. Observe that if we randomly perturb observed labels with uniform noise: $\hat{y}_t = y_t + U(-\epsilon, \epsilon)$, then the distribution on our perturbed points will be $\left(\frac{1}{2rn\epsilon}, rn\right)$-smooth by construction. Now recall that r is a parameter that we can select. By taking $r = \frac{1}{2\rho n\epsilon}$, we obtain that the distribution on the perturbed points is (ρ, rn)-smooth, for a value of ρ that we can take as small as we like. Taking $\rho = 1/\sqrt{T}$ ($r = \frac{\sqrt{T}}{2n\epsilon}$) makes the contribution of ρ to the multivalidity error a low order term. If we feed these perturbed labels to our algorithm, we will obtain prediction intervals that are multivalid for the perturbed labels. But observe that if we simply widen each of our prediction intervals by ϵ at each end, so that we predict the interval $[\bar{\ell}_t - \epsilon, \bar{u}_t + \epsilon)$, then our intervals continue to have coverage probability at least $1 - \delta$ for the original, unperturbed labels. We can similarly take ϵ as small as we like. Our algorithm in Section 6.6.3 will have running time depending polynomially on r, so with this construction obtains a polynomial dependence on $1/\epsilon$.*

6.6.3. Deriving an Efficient Algorithm via Equilibrium Computation

In this section, we show how to implement Algorithm 20 to efficiently sample from the distributions Q_t^L whose existence we established in Lemma 45. We do this by efficiently computing an equilibrium strategy Q_t^L using the Ellipsoid algorithm by solving the linear program in Figure 6.6.1. This linear program has $(rn)^2 + 1$ variables and (a priori) an infinite number of constraints. However, as we will show:

1. The number of constraints can in fact be taken to be finite (albeit exponentially large), and

2. We have an efficient separation oracle to identify violated constraints.

Together, this allows us to apply the Ellipsoid algorithm.

$$\begin{aligned} &\min_{Q^L \in \mathcal{P}^{rn}_{\text{interval}}} \gamma \text{ s.t.} \\ \forall Q^A \in \hat{\mathcal{Q}}_{\rho,rn} : &\sum\nolimits_{y \in \mathcal{P}^{rn}} Q^A(y) \left(\sum\nolimits_{(\ell,u) \in \mathcal{P}^{rn}_{\text{interval}}} Q^L((\ell,u)) \left(v_\delta((l,u),y) C^{\ell,u}_{t-1}(x_t) \right) \right) \le \gamma, \\ &\sum\nolimits_{(\ell,u) \in \mathcal{P}^{rn}_{\text{interval}}} Q^L((\ell,u)) = 1, \\ \forall (\ell,u) \in \mathcal{P}^{rn}_{\text{interval}} : &Q^L((\ell,u)) \ge 0. \end{aligned}$$

Figure 6.6.1: A Linear Program for Computing a Minimax Equilibrium Strategy for the Learner at Round t.

Algorithm 21: Von Neumann's Multivalid Predictor

```
Input: $\epsilon > 0$.
for $t = 1, \ldots, T$ do
    Observe $x_t$ and compute $C^{\ell,u}_{t-1}(x_t)$ for each $(\ell,u) \in \mathcal{P}^{rn}_{\text{interval}}$ as in (6.11).
    Solve the Linear Program from Figure 6.6.1 using the Ellipsoid algorithm, with
     Algorithm 22 as a separation oracle, to obtain an $\epsilon$-approximate solution
     $Q^L_t \in \Delta\mathcal{P}^{rn}_{\text{interval}}$.
    Predict $(\bar{\ell}_t, \bar{u}_t) = (\ell,u)$ with probability $Q^L_t((\ell,u))$.
end
```

Theorem 40. *Algorithm 21 implements Algorithm 20. In particular, it obtains multivalidity guarantees arbitrarily close to those of Theorems 38 and 39. Namely, for any desired $\epsilon > 0$, we have the following.*

Choosing $\eta = \sqrt{\frac{\ln(2|\mathcal{G}|n^2+\epsilon)}{2T}} \in (0, 1/2)$, we have against any adversary constrained to playing (ρ, rn)-smooth distributions that the sequence of prediction intervals produced by Algorithm 21 is α-multivalid with respect to $\mathcal{G}$ in expectation over the randomness of the transcript π_T, where:

$$\mathbb{E}[\alpha] \le \rho + 2\sqrt{\frac{2\ln(2|\mathcal{G}|n^2+\epsilon)}{T}}.$$

Moreover, choosing $\eta = \sqrt{\frac{\ln(2|\mathcal{G}|n^2)+\epsilon T}{2T}} \in (0, 1/2)$*, we have, with probability* $1-\lambda$ *over the randomness of the transcript* π_T*,*

$$\alpha \leq \rho + 4\sqrt{\frac{2}{T}\ln\left(\frac{2|\mathcal{G}|n^2}{\lambda}\right)} + 2\epsilon.$$

The runtime of Algorithm 21 is linear in $|\mathcal{G}|$*, and polynomial in* r, n, T*, and* $\log(\frac{1}{\epsilon})$*.*

Remark 15. *As with all of our other algorithms, the dependence on* $|\mathcal{G}|$ *can be replaced at each round with a possibly substantially smaller dependence on the number of groups which contain* x_t*,* $|\mathcal{G}(x_t)|$*, whenever this set is efficiently enumerable.*

Proof. Recall that at each round t we need to find an equilibrium strategy for the learner in the zero-sum game defined by the objective function:

$$\begin{aligned} u((\ell, u), y) &= \eta v_\delta((\ell, u), y) C_{t-1}^{\ell,u} + 2\eta^2 L_{t-1} \\ &= \eta \left(\text{Cover}((\ell, u), y) - (1-\delta)\right) C_{t-1}^{\ell,u} + 2\eta^2 L_{t-1}. \end{aligned}$$

In this game, the strategy space for the learner is the set of all distributions over discrete intervals: $\mathcal{Q}^L = \Delta\mathcal{P}_{\text{interval}}^{rn}$, and (by Lemma 46), the action space for the adversary can be taken to be the set of all discrete smooth distributions: $\mathcal{Q}^A = \hat{\mathcal{Q}}_{\rho,rn}$.

The equilibrium structure of a game is invariant to adding and multiplying the objective function by a constant. Hence we can proceed to solve the game with the objective function:

$$u((\ell, u), y) = \left(\text{Cover}((\ell, u), y) - (1-\delta)\right) C_{t-1}^{\ell,u}.$$

To compute an equilibrium of the game, we need to solve for a distribution Q^L satisfying:

$$Q^L \in \underset{Q^L \in \Delta\mathcal{P}_{\text{interval}}^{rn}}{\text{argmin}} \max_{Q^A \in \hat{\mathcal{Q}}_{\rho,rn}} \underset{\substack{y \sim Q^A, \\ (\ell,u)\sim Q^L}}{\mathbb{E}} [u(\ell, u), y)].$$

We can write this as a linear program, over the $O((rn)^2)$ variables $Q^L((\ell, u))$: see Figure 6.6.1. A priori, this linear program has infinitely many constraints.[25] Nevertheless, we show that we can efficiently implement a *separation oracle*, which given a candidate solution (Q^L, γ), can find a violated constraint whenever one exists. This is sufficient to efficiently find, using the Ellipsoid algorithm, a feasible solution of the linear program achieving value within any desired $\epsilon > 0$ of the optimum.

Algorithm 22: A Separation Oracle for Linear Program 6.6.1

```
Input: A proposed solution $Q^L$, $\gamma$ for Linear Program 6.6.1
Output: A violated constraint of Linear Program 6.6.1 if one exists, or a
        certification of feasibility.
for $i = 0, 1 \ldots, rn$ do
    Compute
        $W_i \equiv \sum_{(\ell,u)\in\mathcal{P}^{rn}_{\text{interval}}:\text{Cover}((\ell,u),\frac{i}{rn})=1} Q^L((\ell,u))C^{\ell,u}_{t-1}$
end
Let $\sigma : \{0, \ldots, rn\} \to \{0, \ldots, rn\}$ be a permutation such that:
    $W_{\sigma(0)} \geq W_{\sigma(1)} \geq \ldots \geq W_{\sigma(rn)}$.
for $i = 0, 1 \ldots, rn$ do
    Set $Q^A(\sigma(i)) = \min(\rho, 1 - \sum_{j=0}^{i-1} Q^A(\sigma(j))$
end
if $\sum_{y\in\mathcal{P}^{rn}} Q^A(y) \left(\sum_{(\ell,u)\in\mathcal{P}^{rn}_{interval}} Q^L((\ell,u)) \left(v_\delta((l,u),y)C^{\ell,u}_{t-1}\right)\right) > \gamma$, or $Q^L$ not
  a prob. dist. then
    Return the violated constraint.
Return FEASIBLE
```

We will identify the output of Algorithm 20 with the distribution Q^A associated with the constraint it outputs. Observe that if there is a violation (i.e. the proposed solution Q^L, γ

[25] Although in fact, in the proof of Lemma 48, we will show that without loss of generality we can equivalently impose only finitely (but exponentially) many constraints.

is infeasible), and there are ties, i.e. indices i and j such that $W_i = W_j$, then there are multiple candidate Q^A's that could be the output of Algorithm 22. To that end, note that a solution Q^A can be output by Algorithm 22 if and only if it is *greed-induced*:

Definition 40. *Let W_i be defined as in Algorithm 22 for $i \in \{0, \ldots, rn\}$. We say that a distribution $Q^L \in \hat{\mathcal{Q}}_{\rho,rn}$ is* greed-induced *if for every pair of indices i and j such that $W_i > W_j$:*

$$Q^A(j) > 0 \Longrightarrow Q^A(i) = \rho.$$

Lemma 48. *Algorithm 22 is a separation oracle for the Linear Program in Figure 6.6.1. It runs in time $O((rn)^3)$.*

Proof. Recall that a separation oracle is given a candidate distribution $Q^L \in \Delta\mathcal{P}^{rn}_{\text{interval}}$ and a value $\gamma \in \mathbb{R}$, and must determine if there is any $Q^A \in \hat{\mathcal{Q}}_{\rho,rn}$ such that:

$$\sum\nolimits_{y\in\mathcal{P}^{rn}} Q^A(y) \left(\sum\nolimits_{(\ell,u)\in\mathcal{P}^{rn}_{\text{interval}}} Q^L((\ell,u)) \left(v_\delta((l,u),y) C^{\ell,u}_{t-1} \right) \right) > \gamma.$$

Suppose the learner is playing a distribution $Q^L \in \Delta\mathcal{P}^{rn}_{\text{interval}}$ over intervals. The adversary will seek to maximize the objective function over the set of (ρ, rn)-smooth distributions $Q^A \in \hat{\mathcal{Q}}_{\rho,rn}$. Recall that $v_\delta((\ell,u),y) = \text{Cov}((\ell,u),y) - (1-\delta)$. Therefore, fixing a distribution Q^L for the learner, there are terms in the objective function that are independent of the adversary's actions (roughly, those corresponding to the $(1-\delta)$ term), and hence irrelevant to the inner maximization problem (i.e the adversary's best response). We define the following quantity $\tilde{u}$ which eliminates these y-independent terms:

$$\begin{aligned}
\tilde{u}(Q^L, Q^A) &= \sum_{i\in\{0,\ldots,rn\}} Q^A\left(\frac{i}{rn}\right) \sum\nolimits_{(\ell,u)\in\mathcal{P}^{rn}_{\text{interval}}:\text{Cover}((\ell,u),\frac{i}{rn})=1} Q^L((\ell,u)) C^{\ell,u}_{t-1}, \\
&= \sum_{i\in\{0,\ldots,rn\}} Q^A\left(\frac{i}{rn}\right) W_i.
\end{aligned}$$

Observe that for any $Q^L \in \Delta\mathcal{P}_{\text{interval}}$:

$$\underset{Q^A \in \hat{\mathcal{Q}}_{\rho,rn}}{\text{argmax}} \left(\underset{\substack{\tilde{y} \sim Q^A, \\ (\tilde{\ell},\tilde{u}) \sim Q^L}}{\mathbb{E}} [u((\tilde{\ell},\tilde{u}),\tilde{y})] \right) = \underset{Q^A \in \hat{\mathcal{Q}}_{\rho,rn}}{\text{argmax}} \ \tilde{u}(Q^L, Q^A).$$

Hence, to derive a separation oracle, it suffices to find an algorithm which maximizes $\tilde{u}$ given a fixed distribution over intervals Q^L for the learner. This is how we proceed.

Observe that by the argument above, the adversary's problem is equivalent to solving:

$$\begin{aligned} \max_{Q^A} \sum\nolimits_{i \in \{0,\ldots,rn\}} Q^A\left(\frac{i}{rn}\right) W_i, \\ \sum\nolimits_{i \in \{0,\ldots,rn\}} Q^A\left(\frac{i}{rn}\right) = 1, \\ \forall i \in \{0,\ldots,rn\} : Q^A\left(\frac{i}{rn}\right) \leq \rho, \\ \forall i \in \{0,\ldots,rn\} : Q^A\left(\frac{i}{rn}\right) \geq 0. \end{aligned}$$

By observation, this is a fractional knapsack problem—the value of each item $i \in \{0,\ldots,rn\}$ is W_i, the quantity of each item i is ρ, and the total capacity is 1. Therefore the optimal solution is greed-induced.

To bound the runtime of Algorithm 22, first observe that checking that Q^L is a probability distribution takes time $O((rn)^2 \log rn)$. Now, we focus on the remaining constraints. Since the quantities $C_{t-1}^{\ell,u}$ are precomputed at the beginning of round t, the separation oracle computes W_i for each $i \in \{0,\ldots,rn\}$ in time $O((rn)^2)$, and hence we can compute all W_i's in time $O((rn))^3$. All that remains is to sort the indices W_i which takes time $O(rn \ln rn)$, which is a low order term. Altogether, this results in a runtime of $O((rn)^3)$ for Algorithm 22. □

Now, we verify that Algorithm 21 runs efficiently — to do so, we need to show that the Ellipsoid algorithm can efficiently (approximately) solve Linear Program 6.6.1.

Lemma 49. *Each run of the Ellipsoid algorithm within Algorithm 21 solves the LP to a desired accuracy $\epsilon > 0$ in runtime* $\mathrm{poly}(rn, \log|\mathcal{G}|, T, \log\frac{1}{\epsilon})$*. Consequently, Algorithm 21 runs in time* $\mathrm{poly}(rn, |\mathcal{G}|, T, \log\frac{1}{\epsilon})$*, where the dependence on $|\mathcal{G}|$ is $O(|\mathcal{G}|)$.*

Proof. To ensure the Ellipsoid has polynomial runtime, we need to satisfy the conditions of Theorem 37.

We first check that the feasible set of Linear Program 6.6.1 is a polyhedron, i.e. that it has *finitely many faces*. By Lemma 48 above, the adversary always has a *greed-induced* best-response Q^A constructed by Algorithm 22. Every distribution Q^A output by Algorithm 22 corresponds to selecting $\lfloor\frac{1}{\rho}\rfloor$ "full" buckets that will have probability ρ each and one bucket for the remaining probability mass, so there are at most $rn \cdot \binom{rn}{\lfloor\frac{1}{\rho}\rfloor} = O(rn \cdot 2^{rn})$ such distributions. The feasible set of Linear Program 6.6.1 is thus equivalently given by the corresponding finitely many $(O(rn \cdot 2^{rn}))$ constraints.

Thus, the feasible region of LP 6.6.1 is indeed a polyhedron; however, exponential terms in the coefficients of the constraints associated with the adversarial best-responses (which are due to our definition of the soft-max surrogate loss) prevent it from being *rational*. To fix this, we only keep $O(\log\frac{1}{\epsilon})$ bits of precision after the integer part of every coefficient of the original LP, resulting in a new LP whose coefficients are all rational and within $\pm\frac{\epsilon}{2}$ from their original values in LP 6.6.1. The new LP indeed has a rational polyhedron as its feasible region.

We now observe that Linear Program 6.6.1 has the form given in Lemma 42. This implies that by solving the just described rational LP corresponding to LP 6.6.1 *exactly*, we will obtain the desired ϵ-*approximate* solution to Linear Program 6.6.1. With this in mind, it remains to bound the bit complexity of the rational LP.

Consider any constraint of the rational LP. The coefficient of each variable $Q^L((\ell, u))$ has absolute value at most:

$$\begin{aligned}
&\max_{(\ell,u)\in\mathcal{P}_{\text{interval}}} \sum_{G\in\mathcal{G}} \exp(\eta V_{t-1}^{G,(\ell,u)}) - \exp(-\eta V_{t-1}^{G,(\ell,u)}) \\
&\leq |\mathcal{G}| 2 \exp\left(\eta \max_{G\in\mathcal{G},(\ell,u)\in\mathcal{P}_{\text{interval}}} \left|V_{t-1}^{G,(\ell,u)}\right|\right) \\
&\leq 2|\mathcal{G}|\exp(\eta T) \\
&\leq 2|\mathcal{G}|\exp(T).
\end{aligned}$$

Thus, every constraint in the rational LP has bit complexity at most:

$$O\left((rn)^2 \cdot \left(\log|\mathcal{G}| + T + \log\frac{1}{\epsilon}\right)\right),$$

where the $\log\frac{1}{\epsilon}$ term reflects the chosen precision. This is polynomial in $r, n, T, \log|\mathcal{G}|$, and $\log\frac{1}{\epsilon}$. Also, the objective function, which is simply γ, takes $O((rn)^2)$ bits to write down.

We may now apply Theorem 37 with the parameters

$$\begin{aligned}
q &= O((rn)^2), \\
\phi &= O\left((rn)^2(\log|\mathcal{G}| + T + \log\frac{1}{\epsilon})\right), \\
c &= O((rn)^2).
\end{aligned}$$

The runtime of the separation oracle (which, we note, applies to the rational LP just as it did for the original LP) is $O((rn)^3)$ by Lemma 48. Hence, the Ellipsoid algorithm will solve Linear Program 6.6.1 with accuracy ϵ in time $\text{poly}(rn, \log|\mathcal{G}|, T, \log\frac{1}{\epsilon})$.

Hence, Algorithm 21 has time complexity $\text{poly}(rn, |\mathcal{G}|, T, \log\frac{1}{\epsilon})$ — where the dependence on $|\mathcal{G}|$ is linear, because we precompute the $C_{t-1}^{\ell,u}$'s once at the beginning of each round t, taking time linear in $|\mathcal{G}|$, and the runtime of the Ellipsoid algorithm is polylogarithmic in $|\mathcal{G}|$. (We remark once more that the dependence on $|\mathcal{G}|$ can be reduced to a dependence on $|\mathcal{G}(x_t)|$ if $\mathcal{G}(x_t)$ is efficiently enumerable, and that this might be much smaller.) □

Finally, we need to demonstrate that the claimed multivalidity guarantees (which are a function of the chosen $\epsilon > 0$) indeed hold.

Lemma 50. *Algorithm 21 achieves the multivalidity guarantees stated in Theorem 40.*

The proof of this lemma involves repeating several calculations from Section 6.6.2 with an ϵ error term, and so is deferred to the Appendix. □

Appendix

6.A. Unboundedly Many Groups, Bounded Group Membership

In this section, we briefly sketch how we can modify our results so that we can handle the case that there are a “large number” of groups (i.e. $|\mathcal{G}|$ is infinite or larger than 2^T — a range in which the bounds we prove in the main body are vacuous). In this scenario, we maintain the assumption that any given $x \in \mathcal{X}$ appears in at most d groups, i.e. that $|\mathcal{G}(x)| \leq d$ for all $x \in \mathcal{X}$. As we have already noted, in this scenario, our running time dependence on $|\mathcal{G}|$ can be replaced with d — here we show that we can do the same in our convergence bounds.

The first step is to redefine our surrogate loss function L. The way it was previously defined, L_0 was already a quantity at the scale of $|\mathcal{G}|$, and so it would be hopeless to use it for infinite collections of groups. But a small modification solves this problem:

Definition 41 (Surrogate loss function)**.** *Fixing a transcript $\pi_s \in \Pi^*$ and a parameter $\eta \in [0, \frac{1}{2}]$, define a surrogate calibration loss function at day s as:*

$$L_s(\pi_s) = 1 + \sum_{\substack{G \in \mathcal{G}, \\ i \in [n]}} \left(\exp(\eta V_s^{G,i}) + \exp(-\eta V_s^{G,i}) - 2\right).$$

When the transcript π_s is clear from context, we will sometimes simply write L_s.

Observe that this modified function satisfies $L_0 = 1$, independently of the size of $|\mathcal{G}|$, and still allows us to tightly upper bound our calibration loss:

Observation 8. *For any transcript π_T, and any $\eta \in [0, \frac{1}{2}]$, we have that:*

$$\max_{G \in \mathcal{G}, i \in [n]} \left| V_T^{G,i} \right| \leq \frac{1}{\eta} \ln(L_T + 2dT) \leq \max_{G \in \mathcal{G}, i \in [n]} \left| V_T^{G,i} \right| + \frac{\ln(dT)}{\eta}.$$

This observation uses the fact that because (by assumption) $|\mathcal{G}(x_t)| \leq d$ for all t, after T time steps, there are at most dT quantities $V_T^{G,i}$ that are non-zero.

We can now provide a modified bound on $\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu}_{s+1})$:

Lemma 51. *For any transcript $\pi_s \in \Pi^*$, any $x_{s+1} \in \mathcal{X}$, and any $\overline{\mu}_{s+1} \in \mathcal{P}_{mean}$ such that $\overline{\mu}_{s+1} \in B(i)$ for some $i \in [n]$:*

$$\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu}_{s+1}) \leq \eta \left(\mathop{\mathbb{E}}_{\tilde{y}_{s+1}} [\tilde{y}_{s+1}] - \overline{\mu}_{s+1} \right) C_s^i(x_{s+1}) + 2\eta^2 L_s + 4d\eta^2,$$

where for each $i \in [n]$:

$$C_s^i(x_{s+1}) \equiv \sum_{\mathcal{G}(x_{s+1})} \exp(\eta V_s^{G,i}) - \exp(-\eta V_s^{G,i}).$$

Proof. Fix any transcript $\pi_s \in \Pi^*$ (which defines L_s), feature vector $x_{s+1} \in \mathcal{X}$, and $\overline{\mu}_{s+1}$ such that $\overline{\mu}_{s+1} \in B(i)$ for some $i \in [n]$. By direct calculation, we obtain:

$$\begin{aligned}
&\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu}_{s+1}) \\
&= \mathop{\mathbb{E}}_{\tilde{y}_{s+1}} \Bigg[\sum_{G \in \mathcal{G}(x_{s+1})} \exp(\eta V_s^{G,i}) \big(\exp(\eta(\tilde{y}_{s+1} - \overline{\mu}_{s+1})) - 1 \big) \\
&+ \exp(-\eta V_s^{G,i}) \big(\exp(-\eta(\tilde{y}_{s+1} - \overline{\mu}_{s+1})) - 1 \big) \Bigg], \\
&\leq \mathop{\mathbb{E}}_{\tilde{y}_{s+1}} \left[\sum_{G \in \mathcal{G}(x_{s+1})} \exp(\eta V_s^{G,i}) \big(\eta(\tilde{y}_{s+1} - \overline{\mu}_{s+1}) + 2\eta^2 \big) + \exp(-\eta V_s^{G,i}) \big(-\eta(\tilde{y}_{s+1} - \overline{\mu}_{s+1}) + 2\eta^2 \big) \right], \\
&= \eta \left(\mathop{\mathbb{E}}_{\tilde{y}_{s+1}} [\tilde{y}_{s+1}] - \overline{\mu}_{s+1} \right) \sum_{G \in \mathcal{G}(x_{s+1})} \big(\exp(\eta V_s^{G,i}) - \exp(-\eta V_s^{G,i}) \big) \\
&+ 2\eta^2 \sum_{G \in \mathcal{G}(x_{s+1})} \big(\exp(\eta V_s^{G,i}) + \exp(-\eta V_s^{G,i}) \big), \\
&\leq \eta \left(\mathop{\mathbb{E}}_{\tilde{y}_{s+1}} [\tilde{y}_{s+1}] - \overline{\mu}_{s+1} \right) \left(\sum_{G \in \mathcal{G}(x_{s+1})} \exp(\eta V_s^{G,i}) - \exp(-\eta V_s^{G,i}) \right) + 2\eta^2 L_s + 4d\eta^2, \\
&= \eta \left(\mathop{\mathbb{E}}_{\tilde{y}_{s+1}} [\tilde{y}_{s+1}] - \overline{\mu}_{s+1} \right) C_s^i(x_{s+1}) + 2\eta^2 L_s + 4d\eta^2.
\end{aligned}$$

Here, the first inequality follows from the fact that for $0 < |x| < \frac{1}{2}$, $\exp(x) \leq 1+x+2x^2$. □

We can use this to provide a modified bound to Lemma 36.

Lemma 52. *For any transcript $\pi_s \in \Pi^*$, any $x_{s+1} \in \mathcal{X}$, and any $r \in \mathbb{N}$ there exists a distribution over predictions for the learner $Q^L_{s+1} \in \Delta\mathcal{P}^{rn}$, such that regardless of the adversary's choice of distribution of y_{s+1} over $\Delta\mathcal{Y}$, we have that:*

$$\mathop{\mathbb{E}}_{\overline{\mu}\sim Q^L_{s+1}}\left[\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu})\right] \leq L_s\left(\frac{\eta}{rn} + 2\eta^2\right) + 2d.$$

Proof. As in the proof of Lemma 36, we construct a zero-sum game between the learner and the adversary. Fix the transcript π_s and the feature vector x_{s+1}. We define the utility of this game to be the upper bound we proved on $\Delta_{s+1}(\pi_s, x_{s+1}, \overline{\mu})$ in Lemma 51. For each $\overline{\mu} \in \mathcal{P}^{rn}$ and each $y \in [0,1]$, we let:

$$u(\overline{\mu}, y) = \eta\,(y - \overline{\mu})\, C^{\overline{\mu}}_s(x_{s+1}) + 2\eta^2 L_s + 4d\eta^2.$$

We now establish the value of this game. Observe that for any strategy of the adversary (which fixes $\mathbb{E}[\tilde{y}]$), the learner can respond by playing $\overline{\mu}^* = \text{argmin}_{\overline{\mu}\in\mathcal{P}^{rn}} |\,\mathbb{E}[\tilde{y}] - \overline{\mu}|$, and that because of our discretization, $\min|\,\mathbb{E}[\tilde{y}] - \overline{\mu}^*| \leq \frac{1}{rn}$. Therefore, the value of the game is at most:

$$\begin{aligned}\max_{y\in[0,1]}\min_{\overline{\mu}^*\in\mathcal{P}^{rn}} u(\overline{\mu}^*, y) \quad &\leq \quad \max_{\overline{\mu}\in\mathcal{P}^{rn}} \frac{\eta}{rn}\left|C^{\overline{\mu}}_s(x_{s+1})\right| + 2\eta^2 L_s + 4d\eta^2, \\ &\leq \quad L_s\left(\frac{\eta}{rn} + 2\eta^2\right) + 2d.\end{aligned}$$

Here the latter inequality follows since $C^{\overline{\mu}}_s(x_{s+1}) \leq L_s + 2d$ for all $\overline{\mu} \in \mathcal{P}^{rn}$, by observation, and then since $\eta \in (0, \frac{1}{2})$ we have the bound. We can now apply the minimax theorem (Theorem 29) to conclude that there exists a fixed distribution $Q^L_{s+1} \in \mathcal{Q}^L$ for the learner that guarantees that simultaneously for *every* label $y \in [0,1]$ that might be chosen by the adversary:

$$\mathop{\mathbb{E}}_{\overline{\mu}\sim Q^L_{s+1}}[u(\overline{\mu}, y)] \leq L_s\left(\frac{\eta}{rn} + 2\eta^2\right) + 2d,$$

as desired. □

Corollary 10. *For every $r \in \mathbb{N}$, $s \in [T]$, $\pi_s \in \Pi^*$, and $x_{s+1} \in \mathcal{X}$ (which fixes L_s and Q^L_{s+1}), and any distribution over $\mathcal{Y}$:*

$$\mathop{\mathbb{E}}_{\overline{\mu}^L_{s+1}\sim Q_{s+1}}[\tilde{L}_{s+1}|\pi_s] = L_s + \mathop{\mathbb{E}}_{\overline{\mu}_{s+1}\sim Q^L_{s+1}}[\Delta_{s+1}(\pi_{s+1}, x_{s+1}, \overline{\mu}_{s+1})] \leq L_s\left(1 + \frac{\eta}{rn} + 2\eta^2\right) + 2d.$$

Lemma 52 shows that playing the minimax strategy of this zero-sum game (Algorithm 16) continues to provide a low value to the learner. We now show the counterpart of the first part of Theorem 30 for these modified bounds:

Theorem 41. *Consider a nonnegative random process $\tilde{X}_t$ adapted to the filtration $\mathcal{F}_t = \sigma(\pi_t)$, where $\tilde{X}_0$ is constant a.s. Suppose we have that for any period t, and any π_{t-1}, $\mathbb{E}[\tilde{X}_t|\pi_{t-1}] \leq X_{t-1}(1 + \eta c + 2\eta^2) + 2d$ for some $\eta \in [0, \frac{1}{2}], c \in [0, 1], d > 0$. Then we have that:*

$$\mathop{\mathbb{E}}_{\tilde{\pi}_T}[\tilde{X}_T] \leq (X_0 + 2dT)\exp\left(T\eta c + 2T\eta^2\right). \tag{6.12}$$

Proof. First, observe that:

$$\begin{aligned}
\underset{\tilde{\pi}_T}{\mathbb{E}}[\tilde{X}_T] &= \underset{\tilde{\pi}_{T-1}}{\mathbb{E}}\left[\mathbb{E}[\tilde{X}_T|\pi_{T-1}]\right], \\
&\leq \underset{\tilde{\pi}_{T-1}}{\mathbb{E}}\left[\mathbb{E}[\left(1+\eta c+2\eta^2\right)X_{T-1}+2d|\pi_{T-1}]\right] \\
&= \left(1+\eta c+2\eta^2\right)\underset{\tilde{\pi}_{T-1}}{\mathbb{E}}\left[\tilde{X}_{T-1}\right]+2d \\
&\ \vdots \\
&\leq X_0\left(1+\eta c+2\eta^2\right)^T+2d\sum_{t=0}^{T-1}(1+c\eta+2\eta^2)^t, \\
&\leq X_0\left(1+\eta c+2\eta^2\right)^T+2dT(1+c\eta+2\eta^2)^T \\
&= (X_0+2dT)\exp\left(T\ln\left(1+\eta c+2\eta^2\right)\right), \\
&\leq (X_0+2dT)\exp\left(T\eta c+2T\eta^2\right),
\end{aligned}$$

where the last inequality holds because $\ln(1+x) \leq x$ for any $x > -1$. This concludes the proof of (6.12). □

We are now ready to bound our multicalibration error. As a straightforward consequence of Corollary 10 and Theorem 41, we have the following Corollary.

Corollary 11. *Against any adversary, Algorithm 16 instantiated with discretization parameter r results in surrogate loss satisfying:*

$$\underset{\tilde{\pi}_T}{\mathbb{E}}[\tilde{L}_T] \leq (1+2dT)\exp\left(\frac{T\eta}{rn}+2T\eta^2\right).$$

Proof. Note that the first part of Theorem 41 applies to the process L with $L_0 = 1$ and $c = \frac{1}{rn}$. The bound follows by plugging these values into (6.12). □

Next, we can convert this into a bound on Algorithm 16's expected calibration error:

Theorem 42. *When Algorithm 16 is run using n buckets for calibration, discretization*

$r \in \mathbb{N}$, and $\eta = \sqrt{\frac{\ln(1+2dT)}{2T}}$, then against any adversary, its sequence of mean predictions are (α, n)-multicalibrated with respect to $\mathcal{G}$, where:

$$\mathbb{E}[\alpha] \leq \frac{1}{rn} + 2\sqrt{\frac{2\ln(1+4dT)}{T}}.$$

For $r = \frac{\sqrt{T}}{\epsilon n \sqrt{2\ln(1+4dT)}}$ this gives:

$$\mathbb{E}[\alpha] \leq (2+\epsilon)\sqrt{\frac{2}{T}\ln(1+4dT)}.$$

Here the expectation is taken over the randomness of the transcript π_T.

Proof. From Observation 2, it suffices to show that

$$\frac{1}{T}\mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\max_{G\in\mathcal{G}, i\in[n]} |\tilde{V}_T^{G,i}|\right] \leq \frac{1}{rn} + 2\sqrt{\frac{2\ln(1+4dT)}{T}}.$$

We begin by computing a bound on the (exponential of) the expectation of this quantity:

$$\begin{aligned}
\exp\left(\eta \mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\max_{G,i} |\tilde{V}_T^{G,i}|\right]\right) &\leq \mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\exp\left(\eta \max_{G,i} |\tilde{V}_T^{G,i}|\right)\right], \\
&= \mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\max_{G,i} \exp\left(\eta |\tilde{V}_T^{G,i}|\right)\right], \\
&\leq \mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\max_{G,i}\left(\exp\left(\eta \tilde{V}_T^{G,i}\right) + \exp\left(-\eta \tilde{V}_T^{G,i}\right)\right)\right], \\
&\leq \mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\sum_{\substack{G,i \\ G_T(i)\neq\phi}}\left(\exp\left(\eta \tilde{V}_T^{G,i}\right) + \exp\left(-\eta \tilde{V}_T^{G,i}\right)\right)\right], \\
&= \mathop{\mathbb{E}}_{\tilde{\pi}_T}[\tilde{L}_T + 2dT], \\
&\leq (1+2dT)\exp\left(\frac{T\eta}{rn} + 2T\eta^2\right) + 2dT, \\
&\leq (1+4dT)\exp\left(\frac{T\eta}{rn} + 2T\eta^2\right).
\end{aligned}$$

Here the first step is by Jensen's inequality and the second last one follows from Corollary 11.

Taking the logarithm of both sides and dividing by ηT, we have

$$\frac{1}{T}\mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\max_{G,i}|\tilde{V}_T^{G,i}|\right] \leq \frac{\ln(1+4dT)}{\eta T} + \frac{1}{rn} + 2\eta.$$

Choosing $\eta = \sqrt{\frac{\ln(1+4dT)}{2T}}$, we thus obtain the desired inequality

$$\frac{1}{T}\mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\max_{G,i}|\tilde{V}_T^{G,i}|\right] \leq \frac{1}{rn} + 2\sqrt{\frac{2\ln(1+4dT)}{T}}.$$

□

The corresponding high-probability bounds are omitted for brevity. They have the analogous dependence on dT replacing $|\mathcal{G}|$. Similar bounds can be obtained for the case of moment-multicalibration and multivalid intervals with the same approach.

6.B. Mean Conditioned Moment Multicalibrators Can Randomize Over Small Support

In Section 6.5.3, we derived a linear programming based algorithm for making mean conditioned moment multicalibrated predictors. Although we proved that we could reduce the pure strategy space of the learner from (r^2nn') to $4nn'$, a priori, the solutions we find via linear programming could have full support. Here we prove that this need not be the case — there always exists a basic feasible solution of the linear program that we solve that has support only over $k+1$ pure strategies for the learner.

Lemma 53. *For any game with objective function (6.7), there exists a minimax strategy for the learner $\hat{Q}^L \in \hat{\mathcal{Q}}^L_{r,n,n'}$, such that $|support(\hat{Q}^L)| \leq k+1$.*

Proof. Suppose that Q^* is a minimax strategy for the learner.

Observe that the adversary's best response in this problem is straightforward: we have that $\psi_\ell = 1$ if $\sum_{\overline{\mu},\overline{m}^k} F_\ell^{\overline{\mu},\overline{m}^k} Q^*(\overline{\mu},\overline{m}^k) > 0$, that $\psi_\ell = 0$ if $\sum_{\overline{\mu},\overline{m}^k} F_\ell^{\overline{\mu},\overline{m}^k} Q^*(\overline{\mu},\overline{m}^k) < 0$, and otherwise the adversary is indifferent. Define

$$
\begin{aligned}
L_{+} &= \{\ell \in [k] : \sum_{\overline{\mu}, \overline{m}^k} F_{\ell}^{\overline{\mu}, \overline{m}^k} Q^*(\overline{\mu}, \overline{m}^k) > 0\}, \\
L_{-} &= \{\ell \in [k] : \sum_{\overline{\mu}, \overline{m}^k} F_{\ell}^{\overline{\mu}, \overline{m}^k} Q^*(\overline{\mu}, \overline{m}^k) < 0\}, \\
L_{=} &= \{\ell \in [k] : \sum_{\overline{\mu}, \overline{m}^k} F_{\ell}^{\overline{\mu}, \overline{m}^k} Q^*(\overline{\mu}, \overline{m}^k) = 0\}.
\end{aligned}
$$

Note that $L_{+} \cup L_{-} \cup L_{=} = [k]$.

Since Q^* is a minimax strategy, it must solve the following linear program, which corresponds to minimizing the learner's objective value over all strategies Q which engender the same best response for the adversary as Q^*:

$$
\begin{aligned}
&\min_{Q \in \hat{\mathcal{Q}}^L_{r,n,n'}} \sum_{\overline{\mu}, \overline{m}^k} Q(\overline{\mu}, \overline{m}^k) \left(\overline{\mu} C_s^{\overline{\mu}, \overline{m}^k} + \overline{m}^k D_s^{\overline{\mu}, \overline{m}^k} - \hat{\mu}_i^k D_s^{\overline{\mu}, \overline{m}^k} \right) \\
&\text{subject to:} \\
&\forall \ell \in L_{+} : \sum_{\overline{\mu}, \overline{m}^k} F_{\ell}^{\overline{\mu}, \overline{m}^k} Q(\overline{\mu}, \overline{m}^k) \geq 0, \\
&\forall \ell \in L_{-} : \sum_{\overline{\mu}, \overline{m}^k} F_{\ell}^{\overline{\mu}, \overline{m}^k} Q(\overline{\mu}, \overline{m}^k) \leq 0, \\
&\forall \ell \in L_{=} : \sum_{\overline{\mu}, \overline{m}^k} F_{\ell}^{\overline{\mu}, \overline{m}^k} Q(\overline{\mu}, \overline{m}^k) = 0, \\
&\qquad\qquad \sum_{\overline{\mu}, \overline{m}^k} Q(\overline{\mu}, \overline{m}^k) = 1, \\
&\qquad\qquad Q \geq 0.
\end{aligned}
$$

Further, any solution to this LP must also be a minimax strategy for the learner. Observe that this has $k+1$ linear constraints. Any such linear program has a basic feasible solution: so there exists a solution $\hat{Q}^L$ (viewed as a vector) with exactly the number of non-zero entries as the number of binding constraints, i.e. $\leq k+1$, as desired.[26] This is exactly the

[26] As an aside, we point out that this also implies the square submatrix with rows corresponding to binding

statement of the Lemma. □

6.C. Proofs from Section 6.4

Theorem 30. *Consider a nonnegative random process $\tilde{X}_t$ adapted to the filtration $\mathcal{F}_t = \sigma(\pi_t)$, where $\tilde{X}_0$ is constant a.s. Suppose we have that for any period t, and any π_{t-1}, $\mathbb{E}[\tilde{X}_t|\pi_{t-1}] \leq X_{t-1}(1+\eta c+2\eta^2)$ for some $\eta \in [0, \frac{1}{2}], c \in [0,1]$. Then we have that:*

$$\mathop{\mathbb{E}}_{\tilde{\pi}_T}[\tilde{X}_T] \leq X_0 \exp\left(T\eta c + 2T\eta^2\right). \tag{6.3}$$

Further, define a process $\tilde{Z}_t$ adapted to the same filtration by $\tilde{Z}_t = Z_{t-1} + \ln \tilde{X}_t - \mathbb{E}[\ln(\tilde{X}_t)|\pi_{t-1}]$. Suppose that $|Z_t - Z_{t-1}| \leq 2\eta$, where $Z_0 = 0$ a.s. Then, with probability $1-\lambda$,

$$\ln(X_T(\pi_T)) \leq \ln(X_0) + T\left(\eta c + 2\eta^2\right) + \eta\sqrt{8T \ln\left(\frac{1}{\lambda}\right)}. \tag{6.4}$$

constraints and corresponding to non-zero variables is of full rank. Textbook treatments that we are aware of consider either LPs with all inequality constraints or all equality constraints. So for completeness we include the following argument. Convert the LP above into a LP in standard form $\min c^T x$ s.t. $Ax = b, x \geq 0$ by adding/subtracting non-negative slack variables to the inequality constraints L_+, L_-. This is a system of $k+1$ linear equality constraints in $4nn' + |L_-| + |L_+| + 1$ variables. We know that there exists an optimal of this LP that is a Basic feasible solution (BFS) (see e.g. Theorem 4.7 of [93]), i.e. an optimal solution with exactly $k+1$ non-zero variable with the corresponding $(k+1) \times (k+1)$ sub-matrix of A, denoted $\hat{A}$, of full rank. By observation, the number of non-zero Q's in this BFS must equal the number of constraints that bind at equality in the original LP (any non-zero slack variable will correspond to a slack constraint in the original). The sub-matrix of $\bar{A}$ corresponding to the non-zero Q's as columns and binding constraints of the original LP as rows must be of full rank, because these rows have all 0's in the columns corresponding to the slack variables in $\bar{A}$.

Proof. First, observe that:

$$\begin{aligned}
\underset{\tilde{\pi}_T}{\mathbb{E}}[\tilde{X}_T] &= \underset{\tilde{\pi}_{T-1}}{\mathbb{E}}\left[\mathbb{E}[\tilde{X}_T|\pi_{T-1}]\right],\\
&\leq \underset{\tilde{\pi}_{T-1}}{\mathbb{E}}\left[\mathbb{E}[\left(1+\eta c+2\eta^2\right) X_{T-1}|\pi_{T-1}]\right]\\
&= \left(1+\eta c+2\eta^2\right) \underset{\tilde{\pi}_{T-1}}{\mathbb{E}}\left[\tilde{X}_{T-1}\right],\\
&\ \ \vdots\\
&\leq X_0\left(1+\eta c+2\eta^2\right)^T,\\
&= X_0\exp\left(T\ln\left(1+\eta c+2\eta^2\right)\right),\\
&\leq X_0\exp\left(T\eta c+2T\eta^2\right),
\end{aligned}$$

where the last inequality holds because $\ln(1+x) \leq x$ for any $x > -1$. This concludes the proof of (6.3).

Towards demonstrating the high-probability bound 6.4, we first show the following statement.

Lemma 54. *For any π_T, we have*

$$\sum_{t=1}^{T}\left(\underset{\tilde{\pi}_t}{\mathbb{E}}\left[\ln(\tilde{X}_t)\middle|\pi_{t-1}\right] - \ln(X_{t-1}(\pi_{t-1}))\right) \leq T\left(\eta c+2\eta^2\right).$$

Proof. Fixing π_T and taking any $t \leq T$, we have

$$\begin{aligned}
\underset{\tilde{\pi}_t}{\mathbb{E}}\left[\ln(\tilde{X}_t)|\pi_{t-1}\right] &\leq \ln\left(\underset{\tilde{\pi}_t}{\mathbb{E}}[\tilde{X}_t|\pi_{t-1}]\right), && \text{(Jensen's inequality)}\\
&\leq \ln(X_{t-1}(\pi_{t-1})) + \ln\left(1+c\eta+2\eta^2\right), && \text{(by assumption)}\\
&\leq \ln(X_{t-1}(\pi_{t-1})) + \left(c\eta+2\eta^2\right). && (\ln(1+x) \leq x \text{ for any } x > -1)
\end{aligned}$$

Summing over every round $t \in [T]$ gives us the result. □

Now observe that for any π_{t-1}, we have $\mathbb{E}[\tilde{Z}_t|\pi_{t-1}] = Z_{t-1}$, so the process $\tilde{Z}_t$ is a martingale. Further, its increments are bounded by assumption. Recall Azuma's inequality for martingales with bounded increments (see e.g. [18]):

Lemma 55 (Azuma's Inequality)**.** *For any martingale $\{\tilde{Z}_t\}_{t=1}^T$ with $|Z_t - Z_{t-1}| \leq c$ a.s., for all T it holds*

$$\Pr\left[\tilde{Z}_T - \tilde{Z}_0 \geq \epsilon\right] \leq \exp\left(-\frac{\epsilon^2}{2c^2T}\right).$$

By assumption, we may apply Azuma's inequality with $c = 2\eta$, and we obtain

$$\Pr_{\tilde{\pi}_T}\left[\sum_{t=1}^{T}\left(\ln(X_t(\pi_t)) - \mathop{\mathbb{E}}_{\tilde{\pi}_t}[\ln X_t(\tilde{\pi}_t)|\pi_{t-1}]\right) \geq \epsilon\right] \leq \exp\left(-\frac{\epsilon^2}{8\eta^2T}\right).$$

So, with probability $1-\lambda$, it holds that

$$\begin{aligned}
&\sum_{t=1}^{T}\left(\ln(X_t(\pi_t)) - \mathop{\mathbb{E}}_{\tilde{\pi}_t}[\ln X_t(\tilde{\pi}_t)|\pi_{t-1}]\right) \leq \eta\sqrt{8T\ln\left(\frac{1}{\lambda}\right)} \\
\Longrightarrow\, &\ln(X_T(\pi_T)) \leq \ln(X_0) + \left(\sum_{t=1}^{T}\mathop{\mathbb{E}}_{\tilde{\pi}_t}\left[\ln(X_t(\tilde{\pi}_t))|\pi_{t-1}\right] - \ln(X_{t-1}(\pi_{t-1}))\right) + \eta\sqrt{8T\ln\left(\frac{1}{\lambda}\right)} \\
\Longrightarrow\, &\ln(X_T(\pi_T)) \leq \ln(X_0) + T\left(\eta c + 2\eta^2\right) + \eta\sqrt{8T\ln\left(\frac{1}{\lambda}\right)},
\end{aligned}$$

where the last inequality follows from Lemma 54. □

Lemma 37. *At any round $t \in [T]$ and for any realized transcript π_t, $|Z_t - Z_{t-1}| \leq 2\eta$.*

Proof. Observe that

$$\begin{aligned}
|Z_t - Z_{t-1}| &= |\ln(L_t(\pi_t)) - \mathbb{E}\left[\ln(L_t(\tilde{\pi}_t))|\pi_{t-1}\right]| \\
&= \left|\mathbb{E}\left[\ln\left(\frac{L_t(\pi_t)}{L_t(\tilde{\pi}_t)}\right)\middle|\pi_{t-1}\right]\right|
\end{aligned}$$

Note that for any π_t,

$$L_t(\pi_t) = L_{t-1}(\pi_{t-1}) + \Delta_t(\pi_{t-1}, x_t, y_t, \overline{\mu}_t)$$

where

$$\Delta_t(\pi_{t-1}, x_t, y_t, \overline{\mu}_t)$$
$$= \sum_{\mathcal{G}(x_t)} \exp(\eta V_{t-1}^{G,B^{-1}(\overline{\mu}_t)}) \left(\exp(\eta(y_t - \overline{\mu}_t)) - 1\right) + \exp(-\eta V_{t-1}^{G,B^{-1}(\overline{\mu}_t)}) \left(\exp(-\eta(y_t - \overline{\mu}_t)) - 1\right).$$

Since $y_t - \overline{\mu}_t$ must lie in $[-1, 1]$, we have that:

$$(\exp(-\eta) - 1)L_{t-1}(\pi_{t-1}) \leq \Delta_t(\pi_{t-1}, x_t, y_t, \overline{\mu}_t) \leq (\exp(\eta) - 1)L_{t-1}(\pi_{t-1})$$

which implies

$$\exp(-\eta)L_{t-1}(\pi_{t-1}) \leq L_t(\pi_t) \leq \exp(\eta)L_{t-1}(\pi_{t-1}).$$

Hence, for any two transcripts π_t, π_t' which are equal over the first $t-1$ periods, we have

$$\left|\ln\left(\frac{L_t(\pi_t)}{L_t(\pi_t')}\right)\right| \leq \ln\left(\frac{\exp(\eta)}{\exp(-\eta)}\right) = 2\eta.$$

Therefore, $\left|\mathbb{E}\left[\ln\left(\frac{L_t(\pi_t)}{L_t(\tilde{\pi}_t)}\right)\middle|\pi_{t-1}\right]\right| \leq 2\eta$ as desired. □

6.D. Proofs from Section 6.5

Theorem 34. *When Algorithm 18 is run using bucketing coarseness parameters n and n', discretization parameter $r \in \mathbb{N}$, and $\eta = \sqrt{\frac{\ln(4|\mathcal{G}|n \cdot n')}{2T}} \in (0, 1/2)$, then against any adversary, its sequence of mean-moment predictions is (α, β, n, n')-mean-conditioned moment multicalibrated with respect to $\mathcal{G}$, where $\beta = (k+1)\alpha + \frac{k}{2n}$ and:*

$$\mathbb{E}[\alpha] \leq \frac{1}{rn} + \frac{1}{rn'} + 2\sqrt{\frac{2\ln(4|\mathcal{G}|n \cdot n')}{T}}.$$

For $r = \frac{\sqrt{T}(n+n')}{\varepsilon n\cdot n'\cdot\sqrt{2\ln(4|\mathcal{G}|n\cdot n')}}$*, this gives:*

$$\mathbb{E}[\alpha] \le (2+\varepsilon)\sqrt{\frac{2}{T}\ln\left(4|\mathcal{G}|n\cdot n'\right)}.$$

Here the expectation is taken over the randomness of the transcript π_T*.*

Proof. From Observation 5, it suffices to show that:

$$\begin{aligned}
\frac{1}{T}\mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\max_{G\in\mathcal{G},i\in[n],j\in[n']}|\tilde{V}_T^{G,i,j}|\right] &\le \frac{1}{rn}+\frac{1}{rn'}+2\sqrt{\frac{2\ln(4|\mathcal{G}|n\cdot n')}{T}},\\
\frac{1}{T}\mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\max_{G\in\mathcal{G},i\in[n],j\in[n']}|\tilde{M}_T^{G,i,j}|\right] &\le \frac{1}{rn}+\frac{1}{rn'}+2\sqrt{\frac{2\ln(4|\mathcal{G}|n\cdot n')}{T}}.
\end{aligned}$$

We begin by computing a bound on the (exponential of) the expectation of the first quantity:

$$\begin{aligned}
&\exp\left(\eta\mathop{\mathbb{E}}_{\tilde{\pi}_T}[\max_{G,i,j}|\tilde{V}_T^{G,i,j}|]\right)\\
&\le \mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\exp\left(\eta\max_{G,i,j}|\tilde{V}_T^{G,i,j}|\right)\right]\\
&= \mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\max_{G,i,j}\exp\left(\eta|\tilde{V}_T^{G,i,j}|\right)\right]\\
&\le \mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\max_{G,i,j}\left(\exp\left(\eta\tilde{V}_T^{G,i,j}\right)+\exp\left(-\eta\tilde{V}_T^{G,i,j}\right)\right)\right]\\
&\le \mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\sum_{G,i,j}\left(\exp\left(\eta\tilde{V}_T^{G,i,j}\right)+\exp\left(-\eta\tilde{V}_T^{G,i,j}\right)+\exp\left(\eta\tilde{M}_T^{G,i,j}\right)+\exp\left(-\eta\tilde{M}_T^{G,i,j}\right)\right)\right]\\
&= \mathop{\mathbb{E}}_{\tilde{\pi}_T}[\tilde{L}_T]\\
&\le 4|\mathcal{G}|n\cdot n'\cdot\exp\left(\frac{T\eta}{rn}+\frac{T\eta}{rn'}+2T\eta^2\right).
\end{aligned}$$

Here the first inequality follows from Jensen's inequality and the last one follows from Corollary 8. Taking the log of both sides and dividing by ηT we obtain

$$\frac{1}{T}\mathop{\mathbb{E}}_{\tilde{\pi}_T}[\max_{G,i}|\tilde{V}_T^{G,i}|] \le \frac{\ln(4|\mathcal{G}|n\cdot n')}{\eta T}+\frac{1}{rn}+\frac{1}{rn'}+2\eta.$$

Choosing $\eta = \sqrt{\frac{\ln(4|\mathcal{G}|n\cdot n')}{2T}}$, we have

$$\frac{1}{T}\mathop{\mathbb{E}}_{\tilde{\pi}_T}[\max_{G,i}|\tilde{V}_T^{G,i}|] \leq \frac{1}{rn} + \frac{1}{rn'} + 2\sqrt{\frac{2\ln(4|\mathcal{G}|n\cdot n')}{T}}.$$

Repeating the same steps, we get an identical bound for $\frac{1}{T}\mathbb{E}_{\tilde{\pi}_T}[\max_{G\in\mathcal{G},i\in[n],j\in[n']}|\tilde{M}_T^{G,i,j}|]$.

□

Now, given $\tilde{L}$, define $\tilde{Z}$ analogously to the second part of Theorem 30. Next, we can show that the increments of $\tilde{Z}$ thus defined, at any round t, can be bounded.

Lemma 56. *At any round $t \in [T]$ and for any realized transcript π_t, $|Z_t - Z_{t-1}| \leq 2\eta$.*

Proof. Observe that

$$\begin{aligned}|Z_t - Z_{t-1}| &= |\ln(L_t(\pi_t)) - \mathbb{E}\left[\ln(L_t(\tilde{\pi}_t))|\pi_{t-1}\right]| \\ &= \left|\mathbb{E}\left[\ln\left(\frac{L_t(\pi_t)}{L_t(\tilde{\pi}_t)}\right)\middle|\pi_{t-1}\right]\right|\end{aligned}$$

Note that for any π_t,

$$L_t(\pi_t) = L_{t-1}(\pi_{t-1}) + \Delta_t(\pi_{t-1}, x_t, y_t, \overline{\mu}_t, \overline{m}_t^k)$$

where:

$$\begin{aligned}&\Delta_t(\pi_{t-1}, x_t, y_t, \overline{\mu}_t, \overline{m}_t^k) \\ &= \sum_{\mathcal{G}(x_t)} \exp(\eta V_{t-1}^{G,B^{-1}(\overline{\mu}_t),B^{-1}(\overline{m}_t^k)})\left(\exp(\eta(y_t - \overline{\mu}_t)) - 1\right) \\ &+ \exp(-\eta V_{t-1}^{G,B^{-1}(\overline{\mu}_t),B^{-1}(\overline{m}_t^k)})\left(\exp(-\eta(y_t - \overline{\mu}_t)) - 1\right), \\ &+ \sum_{\mathcal{G}(x_t)} \exp(\eta M_{t-1}^{G,B^{-1}(\overline{\mu}_t),B^{-1}(\overline{m}_t^k)})\left(\exp(\eta((y_t - \hat{\mu}_{\overline{\mu}_t})^k - \overline{m}_t^k)) - 1\right) \\ &+ \exp(-\eta M_{t-1}^{G,B^{-1}(\overline{\mu}_t),B^{-1}(\overline{m}_t^k)})\left(\exp(-\eta((y_t - \hat{\mu}_{\overline{\mu}_t})^k - \overline{m}_t^k)) - 1\right).\end{aligned}$$

Since $(y_t - \overline{\mu}_t)$ and $((y_t - \hat{\mu}_{\overline{\mu}_t})^k - \overline{m}_t^k)$ must lie in $[-1, 1]$, we have that:

$$(\exp(-\eta) - 1)L_{t-1}(\pi_{t-1}) \leq \Delta_t(\pi_{t-1}, x_t, y_t, \overline{\mu}_t, \overline{m}_t^k) \leq (\exp(\eta) - 1)L_{t-1}(\pi_{t-1})$$

which implies:

$$\exp(-\eta)L_{t-1}(\pi_{t-1}) \leq L_t(\pi_t) \leq \exp(\eta)L_{t-1}(\pi_{t-1}).$$

Therefore, for any two π_t, π_t' such that the corresponding transcripts for the first $t-1$ periods is the same, we have

$$\left|\ln\left(\frac{L_t(\pi_t)}{L_t(\pi_t')}\right)\right| \leq \ln\left(\frac{\exp(\eta)}{\exp(-\eta)}\right) = 2\eta.$$

Therefore we have $\left|\mathbb{E}\left[\ln\left(\frac{L_t(\pi_t)}{L_t(\tilde{\pi}_t)}\right)\middle|\pi_{t-1}\right]\right| \leq 2\eta$ as desired. □

Theorem 35. *When Algorithm 18 is run using bucketing coarseness parameters n and n', discretization $r \in \mathbb{N}$ and $\eta = \sqrt{\frac{\ln(4|\mathcal{G}|n \cdot n')}{2T}} \in (0, 1/2)$, then against any adversary, with probability $1-\lambda$ over the randomness of the transcript, its sequence of predictions is (α, β, n, n')-mean-conditioned moment multicalibrated with respect to $\mathcal{G}$ for $\beta = (k+1)\alpha + \frac{k}{2n}$ and:*

$$\alpha \leq \frac{1}{rn} + \frac{1}{rn'} + 4\sqrt{\frac{2}{T}\ln\left(\frac{4|\mathcal{G}|n \cdot n'}{\lambda}\right)}.$$

For $r = \frac{\sqrt{T}(n+n')}{\epsilon n \cdot n'\sqrt{2\ln(4|\mathcal{G}|n \cdot n'/\lambda)}}$, this gives:

$$\alpha \leq (4+\epsilon)\sqrt{\frac{2}{T}\ln\left(\frac{4|\mathcal{G}|n \cdot n'}{\lambda}\right)}.$$

Proof. By Lemma 56, the second part of Theorem 30 applies, and plugging in $L_0 = 4|\mathcal{G}|n \cdot n'$ and $c = \frac{1}{rn} + \frac{1}{rn'}$, we have that, with probability $(1-\lambda)$ over the randomness of the transcript:

$$\ln(L_T(\pi_T)) \leq \ln(4|\mathcal{G}|n \cdot n) + T\left(\frac{\eta}{rn} + \frac{\eta}{rn'} + 2\eta^2\right) + \eta\sqrt{8T\ln\left(\frac{1}{\lambda}\right)}.$$

Now, note that

$$\begin{aligned}
&\exp\left(\eta \max_{G,i,j} |V_T^{G,i,j}|\right) \\
&= \max_{G,i,j} \exp\left(\eta |V_T^{G,i,j}|\right), \\
&\le \max_{G,i,j} \left(\exp\left(\eta V_T^{G,i,j}\right) + \exp\left(-\eta V_T^{G,i,j}\right)\right), \\
&\le \sum_{G,i,j} \left(\exp\left(\eta V_T^{G,i,j}\right) + \exp\left(-\eta V_T^{G,i,j}\right) + \exp\left(\eta M_T^{G,i,j}\right) + \exp\left(-\eta M_T^{G,i,j}\right)\right), \\
&= L_T(\pi_T).
\end{aligned}$$

By an analogous argument we have that $\exp\left(\eta \max_{G,i,j} |M_T^{G,i,j}|\right) \le L_T(\pi_T)$.Taking log on both sides and dividing both sides by ηT, we get

$$\frac{1}{T} \max_{G,i} |V_T^{G,i,j}| \le \frac{1}{\eta T} \ln(L_T(\pi_T)) \le \frac{\ln(4|\mathcal{G}|n \cdot n')}{\eta T} + \frac{1}{rn} + \frac{1}{rn'} + 2\eta + \sqrt{\frac{8 \ln\left(\frac{1}{\lambda}\right)}{T}}.$$

Choosing $\eta = \sqrt{\frac{\ln(4|\mathcal{G}|n \cdot n')}{2T}}$, we obtain:

$$\begin{aligned}
\frac{1}{T} \max_{G,i,j} |V_T^{G,i,j}| &\le \frac{1}{rn} + \frac{1}{rn'} + 2\sqrt{\frac{2\ln(4|\mathcal{G}|n \cdot n')}{T}} + \sqrt{\frac{8 \ln\left(\frac{1}{\lambda}\right)}{T}} \\
&\le \frac{1}{rn} + \frac{1}{rn'} + 4\sqrt{\frac{2}{T} \ln\left(\frac{4|\mathcal{G}|n \cdot n'}{\lambda}\right)},
\end{aligned}$$

and, by an analogous argument,

$$\frac{1}{T} \max_{G,i,j} |M_T^{G,i,j}| \le \frac{1}{rn} + \frac{1}{rn'} + 4\sqrt{\frac{2}{T} \ln\left(\frac{4|\mathcal{G}|n \cdot n'}{\lambda}\right)},$$

as desired. □

Lemma 42. *Consider a linear program of the following form, with variables $x \in \mathbb{R}^m$, $\gamma \in \mathbb{R}$*

for some m:

$$\textit{Minimize } \gamma, \quad \textit{subject to:} \quad Ax \leq \gamma \mathbf{1}^m, x \cdot \mathbf{1}^m = 1, x \geq 0.$$

Here, $\mathbf{1}^m \in \mathbb{R}^m$ is the all-ones vector, and $A = (a_{ji})$ is a finite matrix with real entries.

Take any $\epsilon > 0$. Modify the above linear program by replacing matrix A with matrix $\tilde{A} = (\tilde{a}_{ji})$, where each $\tilde{a}_{ji}$ is a rational number within $\pm\frac{\epsilon}{2}$ from a_{ji}, obtained by truncating a_{ji} to $O(\log \frac{1}{\epsilon})$ bits of precision. Then, any optimal solution $(x^{*,r}, \gamma^{*,r})$ of the resulting rational linear program is an ϵ-approximately optimal feasible solution of the original linear program.

Proof. Let (x^*, γ^*) be the optimal solution of the original LP. Consider the constraint of the original (resp. rational) LP associated with any row j of matrix A (resp. $\tilde{A}$). This constraint is written as $\sum_i a_{ji}x_i \leq \gamma$ in the original LP, and $\sum_i \tilde{a}_{ji}x_i \leq \gamma$ in the rational LP. Here and below, i ranges over $[m]$. Now, we have that

$$\sum_i \tilde{a}_{ji}x_i^* \leq \sum_i \left(a_{ji} + \frac{\epsilon}{2}\right) x_i^* = \sum_i a_{ji}x_i^* + \frac{\epsilon}{2}\sum_i x_i^* \leq \gamma^* + \frac{\epsilon}{2}\sum_i x_i^* = \gamma^* + \frac{\epsilon}{2}.$$

Since this holds for any row j of the matrix, then setting $x = x^*$ achieves value at most $\gamma^* + \frac{\epsilon}{2}$ with respect to the rational LP.

Conversely, consider an optimal solution $(x^{*,r}, \gamma^{*,r})$ of the rational LP — by the above, we immediately have $\gamma^{*,r} \leq \gamma^* + \frac{\epsilon}{2}$. We claim it achieves value at most $\gamma^* + \epsilon$ with respect to the original LP. Indeed, for any matrix row j,

$$\begin{aligned}
&\sum_i a_{ji}x_i^{*,r} \leq \sum_i \left(\tilde{a}_{ji} + \frac{\epsilon}{2}\right) x_i^{*,r} \\
&= \sum_i \tilde{a}_{ji}x_i^{*,r} + \frac{\epsilon}{2}\sum_i x_i^{*,r} \\
&= \sum_i \tilde{a}_{ji}x_i^{*,r} + \frac{\epsilon}{2} \leq \gamma^{*,r} + \frac{\epsilon}{2} \leq \left(\gamma^* + \frac{\epsilon}{2}\right) + \frac{\epsilon}{2} = \gamma^* + \epsilon.
\end{aligned}$$

Therefore, by solving the rational LP, we obtain an ϵ-approximate solution to the original

LP, as desired. □

Lemma 43. *Algorithm 19 achieves the multivalidity guarantees specified in Theorem 36.*

Proof. We briefly argue that the additive ϵ-approximation to the (shifted and rescaled) value of the game results in the claimed dependence of the multivalidity guarantees on ϵ. When the learner achieves an ϵ approximation to the value of the game at each round, the statement of Corollary 7 becomes:

$$\mathop{\mathbb{E}}_{Q_{s+1}^L}[\tilde{L}_{s+1}|\pi_s] \leq L_s\left(1+\frac{\eta}{rn}+\frac{\eta}{rn'}+2\eta^2\right)+\eta\epsilon \leq L_s\left(1+\frac{\eta}{rn}+\frac{\eta}{rn'}+2\eta^2\right)+\epsilon.$$

Indeed, recall that the linear program that we solve at each round solves for the value of the game that has been shifted by $2\eta^2 L_s$ *and divided by* η. For the second inequality, recall that $\eta < 1$.

Now, using the telescoping argument from the first part of the proof of Theorem 30, we obtain

$$\begin{aligned}
&\exp\left(\eta \mathop{\mathbb{E}}_{\tilde{\pi}_T}[\max_{G,(i,j)}|\tilde{V}_T^{G,(i,j)}|]\right)\\
&\leq 4|\mathcal{G}|n\cdot n'\left((1+\frac{\eta}{rn}+\frac{\eta}{rn'}+2\eta^2\right)^T+\epsilon\sum_{t=0}^{T-1}\left(1+\frac{\eta}{rn}+\frac{\eta}{rn'}+2\eta^2\right)^t,\\
&\leq 4|\mathcal{G}|n\cdot n'\left((1+\frac{\eta}{rn}+\frac{\eta}{rn'}+2\eta^2\right)^T+\epsilon T\left(1+\frac{\eta}{rn}+\frac{\eta}{rn'}+2\eta^2\right)^T,\\
&= (4|\mathcal{G}|n\cdot n'+\epsilon T)\exp\left(T\ln\left(1+\frac{\eta}{rn}+\frac{\eta}{rn'}+2\eta^2\right)\right),\\
&\leq (4|\mathcal{G}|n\cdot n'+\epsilon T)\exp\left(\frac{T\eta}{rn}+\frac{T\eta}{rn'}+2T\eta^2\right).
\end{aligned}$$

Taking logs and dividing by ηT, we get

$$\frac{1}{T}\mathop{\mathbb{E}}_{\tilde{\pi}_T}[\max_{G,(i,j)}|\tilde{V}_T^{G,(i,j)}|] \leq \frac{\ln(4|\mathcal{G}|n\cdot n'+\epsilon T)}{\eta T}+\frac{1}{rn}+\frac{1}{rn'}+2\eta.$$

Setting the two terms involving η equal, we have:

$$\eta = \sqrt{\frac{\ln(4|\mathcal{G}|n \cdot n' + \epsilon T)}{2T}}.$$

For this choice of η, we obtain the following *in-expectation* multivalidity guarantee (and the same guarantee for the M's):

$$\frac{1}{T} \mathop{\mathbb{E}}_{\tilde{\pi}_T}[\max_{G,(i,j)} |\tilde{V}_T^{G,(i,j)}|] \leq \frac{1}{rn} + \frac{1}{rn'} + 2\sqrt{\frac{2\ln(4|\mathcal{G}|n \cdot n' + \epsilon T)}{T}}.$$

Now, setting $\epsilon = \frac{\epsilon'}{T}$ for any desired $\epsilon' > 0$, we obtain the guarantee (and same for the M's) that

$$\frac{1}{T} \mathop{\mathbb{E}}_{\tilde{\pi}_T}[\max_{G,(i,j)} |\tilde{V}_T^{G,(i,j)}|] \leq \frac{1}{rn} + \frac{1}{rn'} + 2\sqrt{\frac{2\ln(4|\mathcal{G}|n \cdot n' + \epsilon')}{T}} \quad \text{if we set } \eta = \sqrt{\frac{\ln(4|\mathcal{G}|n \cdot n' + \epsilon')}{2T}},$$

and the resulting runtime will be polynomial in T and $\log \frac{1}{\epsilon}$ and thus polynomial in T and $\log \frac{1}{\epsilon'}$.

Now, we show the *high-probability* multivalidity guarantee. In the proof of Theorem 30, the statement of Lemma 54 changes to:

Lemma 57. *For any π_T, we have*

$$\sum_{t=1}^{T} \left(\mathop{\mathbb{E}}_{\tilde{\pi}_t} \left[\ln(\tilde{X}_t) \middle| \pi_{t-1} \right] - \ln(X_{t-1}(\pi_{t-1})) \right) \leq T \left(\eta c + 2\eta^2 + \epsilon \right).$$

Proof. Fixing π_T and taking any $t \leq T$, we have

$$\begin{aligned}
&\mathop{\mathbb{E}}_{\tilde{\pi}_t}\left[\ln(\tilde{X}_t)|\pi_{t-1}\right] \\
&\leq \ln\left(\mathop{\mathbb{E}}_{\tilde{\pi}_t}[\tilde{X}_t|\pi_{t-1}]\right) \\
&\leq \ln\left(X_{t-1}(\pi_{t-1})\cdot\left(1+c\eta+2\eta^2\right)+\epsilon\right) \\
&\leq \ln\left(X_{t-1}(\pi_{t-1})\cdot\left(1+c\eta+2\eta^2\right)\right)+\frac{\epsilon}{X_{t-1}(\pi_{t-1})\cdot(1+c\eta+2\eta^2)} \\
&\leq \ln(X_{t-1}(\pi_{t-1}))+\ln\left(1+c\eta+2\eta^2\right)+\epsilon \\
&\leq \ln(X_{t-1}(\pi_{t-1}))+\left(c\eta+2\eta^2+\epsilon\right).
\end{aligned}$$

The first inequality follows from Jensen's inequality, the second from the fact that we computed an ϵ-approximation, the third from $\ln(x+y) \leq \ln(x) + \frac{y}{x}$ for $x, y \geq 0$, the fourth from the fact that loss satisfies $X_{t-1}(\pi_{t-1}) \geq 1$, and the last from the fact that $\ln(1+x) \leq x$ for any $x > -1$. Summing over every round $t \in [T]$ gives us the result. □

Thus, the statement of the second part of Theorem 30 becomes that with probability $1-\lambda$,

$$\ln(X_T(\pi_T)) \leq \ln(X_0) + T\left(\eta c + 2\eta^2 + \epsilon\right) + \eta\sqrt{8T\ln\left(\frac{1}{\lambda}\right)}.$$

Now, applying it to the setting at hand, we obtain:

$$\ln(L_T(\pi_T)) \leq \ln(4|\mathcal{G}|n\cdot n') + T\left(\eta\left(\frac{1}{rn}+\frac{1}{rn'}\right)+2\eta^2+\epsilon\right)+\eta\sqrt{8T\ln\left(\frac{1}{\lambda}\right)}.$$

Thus, taking log on both sides and dividing both sides by ηT, we get

$$\frac{1}{T}\max_{G,i,j}|V_T^{G,(i,j)}| \leq \frac{1}{\eta T}\ln(L_T(\pi_T)) \leq \frac{\ln(4|\mathcal{G}|n\cdot n')}{\eta T}+\frac{1}{rn}+\frac{1}{rn'}+2\eta+\frac{\epsilon}{\eta}+\sqrt{\frac{8\ln\left(\frac{1}{\lambda}\right)}{T}}.$$

Choosing $\eta = \sqrt{\frac{\ln(4|\mathcal{G}|n \cdot n') + \epsilon T}{2T}}$, we obtain (and the same holds for the M's):

$$\begin{aligned}\frac{1}{T}\max_{G,i,j}|V_T^{G,i,j}| &\leq \frac{1}{rn} + \frac{1}{rn'} + 2\sqrt{\frac{2(\ln(4|\mathcal{G}|n \cdot n') + \epsilon T)}{T}} + \sqrt{\frac{8\ln\left(\frac{1}{\lambda}\right)}{T}} \\ &\leq \frac{1}{rn} + \frac{1}{rn'} + 4\sqrt{\frac{2}{T}\ln\left(\frac{4|\mathcal{G}|n \cdot n'}{\lambda}\right)} + 2\epsilon,\end{aligned}$$

as desired. □

6.E. Proofs from Section 6.6

Lemma 44. *For every transcript $\pi_s \in \Pi^*$, every $x_{s+1} \in \mathcal{X}$, and every $(\overline{\ell}_{s+1}, \overline{u}_{s+1}) \in B_n(i,j)$ we have that:*

$$\Delta_{s+1}(\pi_s, x_{s+1}, (\overline{\ell}_{s+1}, \overline{u}_{s+1})) \leq \left(\eta(\mathop{\mathbb{E}}_{\tilde{y}_{s+1}}[v_\delta((\overline{\ell}_{s+1}, \overline{u}_{s+1}), \tilde{y}_{s+1})])\right) C_s^{i,j}(x_{s+1}) + 2\eta^2 L_s,$$

where for each $i \leq j \in [n]$, we have defined

$$C_s^{i,j}(x_{s+1}) \equiv \sum_{\mathcal{G}(x_{s+1})} \exp(\eta V_s^{G,(i,j)}) - \exp(-\eta V_s^{G,(i,j)}).$$

When x_{s+1} is clear from context, for notational economy, we will elide it and simply write $C_s^{i,j}$.

Proof. We calculate:

$$\Delta_{s+1}(\pi_s, x_{s+1}, (\overline{\ell}_{s+1}, \overline{u}_{s+1}))$$
$$= \mathop{\mathbb{E}}_{\tilde{y}_{s+1}}\Bigg[\sum_{\mathcal{G}(x_{s+1})} \exp(\eta V_s^{G,(i,j)})\left(\exp(\eta v_\delta((\overline{\ell}_{s+1}, \overline{u}_{s+1}), \tilde{y}_{s+1})) - 1\right)$$
$$+ \exp(-\eta V_s^{G,(i,j)})\left(\exp(-\eta v_\delta((\overline{\ell}_{s+1}, \overline{u}_{s+1}), \tilde{y}_{s+1}) - 1\right)\Bigg]$$
$$\leq \mathop{\mathbb{E}}_{\tilde{y}_{s+1}}\Bigg[\sum_{\mathcal{G}(x_{s+1})} \exp(\eta V_s^{G,(i,j)})\left(\eta v_\delta((\overline{\ell}_{s+1}, \overline{u}_{s+1}), \tilde{y}_{s+1}) + 2\eta^2\right)$$
$$+ \exp(-\eta V_s^{G,(i,j)})\left(-\eta v_\delta((\overline{\ell}_{s+1}, \overline{u}_{s+1}), \tilde{y}_{s+1}) + 2\eta^2\right)\Bigg]$$
$$= \eta(\mathop{\mathbb{E}}_{\tilde{y}_{s+1}}[v_\delta((\overline{\ell}_{s+1}, \overline{u}_{s+1}), \tilde{y}_{s+1})])C_s^{i,j} + 2\eta^2 \sum_{\mathcal{G}(x_{s+1})} \exp(\eta V_s^{G,(i,j)}) + \exp(-\eta V_s^{G,(i,j)})$$
$$\leq \eta(\mathop{\mathbb{E}}_{\tilde{y}_{s+1}}[v_\delta((\overline{\ell}_{s+1}, \overline{u}_{s+1}), \tilde{y}_{s+1})])C_s^{i,j} + 2\eta^2 L_s,$$

as desired. Here the first inequality follows from the fact that for $0 < |x| < \frac{1}{2}$, $\exp(x) \leq 1 + x + 2x^2$, the following equality from organizing terms and the final inequality by noting that $\sum_{\mathcal{G}(x_{s+1})} \exp(\eta V_s^{G,(i,j)}) + \exp(-\eta V_s^{G,(i,j)}) \leq L_s$ by definition of L. □

Theorem 38. *When Algorithm 20 is run using n buckets, discretization parameter r and $\eta = \sqrt{\frac{\ln(2|\mathcal{G}|n^2)}{2T}} \in (0, 1/2)$, then against any adversary constrained to playing (ρ, rn)-smooth distributions, its sequence of interval predictions is α-multivalid with respect to $\mathcal{G}$ in expectation over the randomness of the transcript π_T, where:*

$$\mathbb{E}[\alpha] \leq \rho + 2\sqrt{\frac{2\ln(2|\mathcal{G}|n^2)}{T}}.$$

Proof. From Observation 6, it suffices to show that $\frac{1}{T}\mathbb{E}_{\pi_T}[\max|V_T^{G,(i,j)}|] \leq \alpha$.

We begin by computing a bound on the (exponential of) the expectation of this quantity:

$$\begin{aligned}
\exp\left(\eta \mathop{\mathbb{E}}_{\tilde{\pi}_T}[\max_{G,(i,j)} |\tilde{V}_T^{G,(i,j)}|]\right) &\leq \mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\exp\left(\eta \max_{G,(i,j)} |\tilde{V}_T^{G,(i,j)}|\right)\right] \\
&= \mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\max_{G,(i,j)} \exp\left(\eta |\tilde{V}_T^{G,(i,j)}|\right)\right] \\
&\leq \mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\max_{G,(i,j)} \left(\exp\left(\eta \tilde{V}_T^{G,(i,j)}\right) + \exp\left(-\eta V_T^{G,(i,j)}\right)\right)\right] \\
&\leq \mathop{\mathbb{E}}_{\tilde{\pi}_T}\left[\sum_{G,(i,j)} \left(\exp\left(\eta \tilde{V}_T^{G,(i,j)}\right) + \exp\left(-\eta \tilde{V}_T^{G,(i,j)}\right)\right)\right] \\
&= \mathop{\mathbb{E}}_{\tilde{\pi}_T}[\tilde{L}_T(\tilde{\pi}_T)] \\
&\leq 2|\mathcal{G}|n^2 \exp\left(T\eta\rho + 2T\eta^2\right).
\end{aligned}$$

Here the first inequality follows from Jensen's inequality and the last one follows from Lemma 47. Taking the log of both sides and dividing by ηT we obtain:

$$\frac{1}{T} \mathop{\mathbb{E}}_{\tilde{\pi}_T}[\max_{G,(i,j)} |\tilde{V}_T^{G,(i,j)}|] \leq \frac{\ln(2|\mathcal{G}|n^2)}{\eta T} + \rho + 2\eta.$$

Choosing $\eta = \sqrt{\frac{\ln(2|\mathcal{G}|n^2)}{2T}}$ we obtain:

$$\frac{1}{T} \mathop{\mathbb{E}}_{\tilde{\pi}_T}[\max_{G,(i,j)} |\tilde{V}_T^{G,(i,j)}|] \leq \rho + 2\sqrt{\frac{2\ln(2|\mathcal{G}|n^2)}{T}},$$

as desired. □

Now, given $\tilde{L}$, define $\tilde{Z}$ analogously to the second part of Theorem 30. Next, we can show that the increments of $\tilde{Z}$ thusly defined, at any round t, can be bounded.

Lemma 58. *At any round $t \in [T]$ and for any realized transcript π_t, $|Z_t - Z_{t-1}| \leq 2\eta$.*

Proof. Observe that

$$\begin{aligned}|Z_t - Z_{t-1}| &= |\ln(L_t(\pi_t)) - \mathbb{E}\left[\ln(L_t(\tilde{\pi}_t))|\pi_{t-1}\right]| \\ &= \left|\mathbb{E}\left[\ln\left(\frac{L_t(\pi_t)}{L_t(\tilde{\pi}_t)}\right)\middle|\pi_{t-1}\right]\right|\end{aligned}$$

Note that for any π_t,

$$L_t(\pi_t) = L_{t-1}(\pi_{t-1}) + \Delta_t(\pi_{t-1}, x_t, y_t, (\ell_t, \mu_t))$$

where:

$$\begin{aligned}&\Delta_t(\pi_{t-1}, x_t, y_t, (\ell_t, u_t)) \\ &= \sum_{\mathcal{G}(x_t)} \exp(\eta V_{t-1}^{G,B_n^{-1}(\ell_t,u_t)})\left(\exp(\eta v_\delta((\ell_t, u_t), y_t)) - 1\right) \\ &\quad + \exp(-\eta V_{t-1}^{G,B_n^{-1}(\ell_t,u_t)})\left(\exp(-\eta v_\delta((\ell_t, u_t), y_t) - 1\right).\end{aligned}$$

Since $v_\delta((\ell_t, u_t), y_t)$ must lie in $[-1, 1]$ (actually $[-(1-\delta), \delta]$), we have that:

$$(\exp(-\eta) - 1)L_{t-1}(\pi_{t-1}) \leq \Delta_t(\pi_{t-1}, x_t, y_t, (\ell_t, u_t)) \leq (\exp(\eta) - 1)L_{t-1}(\pi_{t-1})$$

which implies:

$$\exp(-\eta)L_{t-1}(\pi_{t-1}) \leq L_t(\pi_t) \leq \exp(\eta)L_{t-1}(\pi_{t-1}).$$

Therefore, for any two π_t, π_t' such that the corresponding transcripts for the first $t-1$ periods are the same, we have

$$\left|\ln\left(\frac{L_t(\pi_t)}{L_t(\pi_t')}\right)\right| \leq \ln\left(\frac{\exp(\eta)}{\exp(-\eta)}\right) = 2\eta.$$

Therefore we have $\left|\mathbb{E}\left[\ln\left(\frac{L_t(\pi_t)}{L_t(\tilde{\pi}_t)}\right)\middle|\pi_{t-1}\right]\right| \leq 2\eta$ as desired. □

Theorem 39. *When Algorithm 20 is run using n buckets, discretization parameter r and*

$\eta = \sqrt{\frac{\ln(2|\mathcal{G}|n^2)}{2T}} \in (0, 1/2)$*, then against any adversary who is constrained to playing* (ρ, rn)*-smooth distributions, its sequence of interval predictions is* α*-multivalid with respect to* $\mathcal{G}$ *with probability* $1 - \lambda$ *over the randomness of the transcript* π_T*:*

$$\alpha \leq \rho + 4\sqrt{\frac{2}{T} \ln\left(\frac{2|\mathcal{G}|n^2}{\lambda}\right)}.$$

Proof. By Lemma 58, the second part of Theorem 30 applies, and plugging in $L_0 = 2|\mathcal{G}|n^2$ and $c = \rho$, we have that, with probability $(1 - \lambda)$ over the randomness of the transcript:

$$\ln(L_T(\pi_T)) \leq \ln(2|\mathcal{G}|n^2) + T\left(\eta\rho + 2\eta^2\right) + \eta\sqrt{8T \ln\left(\frac{1}{\lambda}\right)}.$$

Now, note that

$$\begin{aligned}
\exp\left(\eta \max_{G,i,j} |V_T^{G,(i,j)}|\right) &= \max_{G,i,j} \exp\left(\eta |V_T^{G,(i,j)}|\right), \\
&\leq \max_{G,i,j} \left(\exp\left(\eta V_T^{G,(i,j)}\right) + \exp\left(-\eta V_T^{G,(i,j)}\right)\right), \\
&\leq \sum_{G,i,j} \left(\exp\left(\eta V_T^{G,(i,j)}\right) + \exp\left(-\eta V_T^{G,(i,j)}\right)\right), \\
&= L_T(\pi_T).
\end{aligned}$$

Taking log on both sides and dividing both sides by ηT, we get

$$\frac{1}{T} \max_{G,i,j} |V_T^{G,(i,j)}| \leq \frac{1}{\eta T} \ln(L_T(\pi_T)) \leq \frac{\ln(2|\mathcal{G}|n^2)}{\eta T} + \rho + 2\eta + \sqrt{\frac{8 \ln\left(\frac{1}{\lambda}\right)}{T}}.$$

Choosing $\eta = \sqrt{\frac{\ln(2|\mathcal{G}|n^2)}{2T}}$, we obtain

$$\begin{aligned}
\frac{1}{T} \max_{G,i,j} |V_T^{G,i,j}| &\leq \rho + 2\sqrt{\frac{2 \ln(2|\mathcal{G}|n^2)}{T}} + \sqrt{\frac{8 \ln\left(\frac{1}{\lambda}\right)}{T}} \\
&\leq \rho + 4\sqrt{\frac{2}{T} \ln\left(\frac{2|\mathcal{G}|n^2}{\lambda}\right)},
\end{aligned}$$

as desired. □

Lemma 50. *Algorithm 21 achieves the multivalidity guarantees stated in Theorem 40.*

Proof. We briefly argue that the additive ϵ-approximation to the (shifted and rescaled) value of the game results in the claimed dependence of the multivalidity guarantees on ϵ. When the learner achieves an ϵ approximation to the value of the game at each round, the statement of Corollary 9 becomes:

$$\mathop{\mathbb{E}}_{(\ell,u)\sim Q^L_{s+1}}[\tilde{L}_{s+1}|\pi_s] \leq L_s\left(1+\eta\rho+2\eta^2\right)+\eta\epsilon \leq L_s\left(1+\eta\rho+2\eta^2\right)+\epsilon.$$

Indeed, recall that the linear program that we solve at each round solves for the value of the game that has been shifted by $2\eta^2 L_s$ *and divided by* η. For the second inequality, recall that $\eta < 1$.

Now, using the telescoping argument from the first part of the proof of Theorem 30, we obtain

$$\begin{aligned}\exp\left(\eta\mathop{\mathbb{E}}_{\tilde{\pi}_T}[\max_{G,(i,j)}|\tilde{V}_T^{G,(i,j)}|]\right) \leq& 2|\mathcal{G}|n^2\left(1+\eta\rho+2\eta^2\right)^T+\epsilon\sum_{t=0}^{T-1}(1+\eta\rho+2\eta^2)^t,\\ \leq& 2|\mathcal{G}|n^2\left(1+\eta\rho+2\eta^2\right)^T+\epsilon T(1+\eta\rho+2\eta^2)^T,\\ =&(2|\mathcal{G}|n^2+\epsilon T)\exp\left(T\ln\left(1+\eta\rho+2\eta^2\right)\right),\\ \leq&(2|\mathcal{G}|n^2+\epsilon T)\exp\left(T\eta\rho+2T\eta^2\right),\end{aligned}$$

Taking logs and dividing by ηT, we get

$$\frac{1}{T}\mathop{\mathbb{E}}_{\tilde{\pi}_T}[\max_{G,(i,j)}|\tilde{V}_T^{G,(i,j)}|] \leq \frac{\ln(2|\mathcal{G}|n^2+\epsilon T)}{\eta T}+\rho+2\eta.$$

Setting the two terms involving η equal, we have:

$$\eta=\sqrt{\frac{\ln(2|\mathcal{G}|n^2+\epsilon T)}{2T}}.$$

For this choice of η, we obtain the following *in-expectation* multivalidity guarantee:

$$\frac{1}{T}\mathop{\mathbb{E}}_{\tilde{\pi}_T}[\max_{G,(i,j)}|\tilde{V}_T^{G,(i,j)}|] \leq \rho + 2\sqrt{\frac{2\ln(2|\mathcal{G}|n^2+\epsilon T)}{T}}.$$

Now, setting $\epsilon = \frac{\epsilon'}{T}$ for any desired $\epsilon' > 0$, we obtain the guarantee that

$$\frac{1}{T}\mathop{\mathbb{E}}_{\tilde{\pi}_T}[\max_{G,(i,j)}|\tilde{V}_T^{G,(i,j)}|] \leq \rho + 2\sqrt{\frac{2\ln(2|\mathcal{G}|n^2+\epsilon')}{T}} \quad \text{if we set } \eta = \sqrt{\frac{\ln(2|\mathcal{G}|n^2+\epsilon')}{2T}},$$

and the resulting runtime will be polynomial in T and $\log\frac{1}{\epsilon}$ and thus polynomial in T and $\log\frac{1}{\epsilon'}$.

Now, we show the *high-probability* multivalidity guarantee. In the proof of Theorem 30, the statement of Lemma 54 changes to:

Lemma 57. *For any π_T, we have*

$$\sum_{t=1}^{T}\left(\mathop{\mathbb{E}}_{\tilde{\pi}_t}\left[\ln(\tilde{X}_t)\middle|\pi_{t-1}\right] - \ln(X_{t-1}(\pi_{t-1}))\right) \leq T\left(\eta c + 2\eta^2 + \epsilon\right).$$

We show this updated claim in the proof of Lemma 43 of Section 6.5.3.

Thus, the statement of the second part of Theorem 30 becomes that with probability $1-\lambda$,

$$\ln(X_T(\pi_T)) \leq \ln(X_0) + T\left(\eta c + 2\eta^2 + \epsilon\right) + \eta\sqrt{8T\ln\left(\frac{1}{\lambda}\right)}.$$

Now, applying it to the setting at hand, we obtain:

$$\ln(L_T(\pi_T)) \leq \ln(2|\mathcal{G}|n^2) + T\left(\eta\rho + 2\eta^2 + \epsilon\right) + \eta\sqrt{8T\ln\left(\frac{1}{\lambda}\right)}.$$

Thus, taking log on both sides and dividing both sides by ηT, we get

$$\frac{1}{T}\max_{G,i,j}|V_T^{G,(i,j)}| \leq \frac{1}{\eta T}\ln(L_T(\pi_T)) \leq \frac{\ln(2|\mathcal{G}|n^2)}{\eta T} + \rho + 2\eta + \frac{\epsilon}{\eta} + \sqrt{\frac{8\ln\left(\frac{1}{\lambda}\right)}{T}}.$$

Choosing $\eta = \sqrt{\frac{\ln(2|\mathcal{G}|n^2)+\epsilon T}{2T}}$, we obtain:

$$\begin{aligned}\frac{1}{T}\max_{G,i,j}|V_T^{G,i,j}| &\leq \rho + 2\sqrt{\frac{2(\ln(2|\mathcal{G}|n^2)+\epsilon T)}{T}} + \sqrt{\frac{8\ln\left(\frac{1}{\lambda}\right)}{T}} \\ &\leq \rho + 4\sqrt{\frac{2}{T}\ln\left(\frac{2|\mathcal{G}|n^2}{\lambda}\right) + 2\epsilon},\end{aligned}$$

as desired. □

BIBLIOGRAPHY

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 2312–2320, 2011. URL http://papers.nips.cc/paper/4417-improved-algorithms-for-linear-stochastic-bandits. 32, 33

[2] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna M. Wallach. A reductions approach to fair classification. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 60–69. PMLR, 2018. URL http://proceedings.mlr.press/v80/agarwal18a.html. 1, 75, 124

[3] Alekh Agarwal, Miroslav Dudík, and Zhiwei Steven Wu. Fair regression: Quantitative definitions and reduction-based algorithms. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 120–129. PMLR, 2019. URL http://proceedings.mlr.press/v97/agarwal19d.html. 1

[4] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. propublica, may 23, 2016, 2016. 1, 96

[5] Sanjeev Arora, Elad Hazan, and Satyen Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012. 42, 45, 55

[6] Rina Foygel Barber, Emmanuel J Candes, Aaditya Ramdas, and Ryan J Tibshirani. The limits of distribution-free conditional predictive inference. *arXiv preprint arXiv:1903.04684*, 2019. 126, 127, 194

[7] Raef Bassily, Kobbi Nissim, Adam Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman. Algorithmic stability for adaptive data analysis. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 1046–1059, 2016. 144

[8] Yahav Bechavod, Christopher Jung, and Zhiwei Steven Wu. Metric-free individual fairness in online learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/80b618ebcac7aa97a6dac2ba65cb7e36-Abstract.html. 42

[9] Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, 50(1):3–44, 2021. 4

[10] Xuanyu Cao and K. J. Ray Liu. Online convex optimization with time-varying constraints and bandit feedback. *IEEE Trans. Autom. Control.*, 64(7):2665–2680, 2019. doi: 10.1109/TAC.2018.2884653. URL https://doi.org/10.1109/TAC.2018.2884653. 48, 49

[11] Nicolò Cesa-Bianchi, Yoav Freund, David Haussler, David P. Helmbold, Robert E. Schapire, and Manfred K. Warmuth. How to use expert advice. *J. ACM*, 44(3): 427–485, May 1997. ISSN 0004-5411. doi: 10.1145/258128.258179. URL https://doi.org/10.1145/258128.258179. 42, 45, 55

[12] Victor Chernozhukov, Kaspar Wüthrich, and Zhu Yinchu. Exact and robust conformal inference methods for predictive machine learning with dependent data. In *Conference On Learning Theory*, pages 732–749, 2018. 194

[13] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data*, 5(2):153–163, 2017. 2, 7, 193

[14] Alexandra Chouldechova and Aaron Roth. A snapshot of the frontiers of fairness in machine learning. *Communications of the ACM*, 63(5):82–89, 2020. 127, 195

[15] Vincent Conitzer, Walter Sinnott-Armstrong, Jana Schaich Borg, Yuan Deng, and Max Kramer. Moral decision making frameworks for artificial intelligence. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematics (ISAIM)*, 2018. 2

[16] Sam Corbett-Davies and Sharad Goel. The measure and mismeasure of fairness: A critical review of fair machine learning. *CoRR*, abs/1808.00023, 2018. URL http://arxiv.org/abs/1808.00023. 1

[17] A Philip Dawid. The well-calibrated bayesian. *Journal of the American Statistical Association*, 77(379):605–610, 1982. 118, 125, 185, 192

[18] Devdatt P Dubhashi and Alessandro Panconesi. *Concentration of measure for the analysis of randomized algorithms.* Cambridge University Press, 2009. 269

[19] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard S. Zemel. Fairness through awareness. In Shafi Goldwasser, editor, *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pages 214–226. ACM, 2012. doi: 10.1145/2090236.2090255. URL https://doi.org/10.1145/2090236.2090255. 1, 2, 7, 8, 13, 15, 16, 76, 77, 127

[20] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 117–126, 2015. 144

[21] Cynthia Dwork, Michael P Kim, Omer Reingold, Guy N Rothblum, and Gal Yona. Learning from outcomes: Evidence-based rankings. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 106–125. IEEE, 2019. 126, 193

[22] Cynthia Dwork, Michael P Kim, Omer Reingold, Guy N Rothblum, and Gal Yona. Outcome indistinguishability. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1095–1108, 2021. 193

[23] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. On agnostic learning of parities, monomials, and halfspaces. *SIAM Journal on Computing*, 39(2):606–645, 2009. 75

[24] Vitaly Feldman, Venkatesan Guruswami, Prasad Raghavendra, and Yi Wu. Agnostic learning of monomials by halfspaces is hard. *SIAM Journal on Computing*, 41(6): 1558–1590, 2012. 75

[25] Dean P Foster. A proof of calibration via blackwell's approachability theorem. *Games and Economic Behavior*, 29(1-2):73–78, 1999. 192

[26] Dean P Foster and Sergiu Hart. Smooth calibration, leaky forecasts, finite recall, and nash dynamics. *Games and Economic Behavior*, 109:271–293, 2018. 193

[27] Dean P Foster and Sergiu Hart. Forecast hedging and calibration. *Journal of Political Economy*, 129(12):3447–3490, 2021. 192

[28] Dean P Foster and Sham M Kakade. Calibration via regression. In *2006 IEEE Information Theory Workshop-ITW'06 Punta del Este*, pages 82–86. IEEE, 2006. 193

[29] Dean P Foster and Rakesh V Vohra. Asymptotic calibration. *Biometrika*, 85(2): 379–390, 1998. 125, 185, 186, 188, 192

[30] Dean P Foster, Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Complexity-based approach to calibration with checking rules. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 293–314, 2011. 193

[31] Yoav Freund and Robert E Schapire. Game theory, on-line prediction and boosting. In *COLT*, volume 96, pages 325–332. Citeseer, 1996. 83, 104

[32] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, 1997. doi: 10.1006/jcss.1997.1504. URL https://doi.org/10.1006/jcss.1997.1504. 42, 45, 55

[33] Drew Fudenberg and David K Levine. An easier way to calibrate. *Games and economic behavior*, 29(1-2):131–137, 1999. 186, 188, 189, 190, 192

[34] Drew Fudenberg and David K Levine. Conditional universal consistency. *Games and Economic Behavior*, 29(1-2):104–130, 1999. 193

[35] Stephen Gillen, Christopher Jung, Michael J. Kearns, and Aaron Roth. Online learning with an unknown fairness metric. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 2605–2614, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/50905d7b2216bfeccb5b41016357176b-Abstract.html. 14

[36] Chirag Gupta, Aleksandr Podkopaev, and Aaditya Ramdas. Distribution-free binary classification: prediction sets, confidence intervals and calibration. *Advances in Neural Information Processing Systems*, 33, 2020. 195

[37] Swati Gupta and Vijay Kamble. Individual fairness in hindsight. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*, pages 805–806, 2019. doi: 10.1145/3328526.3329605. URL https://doi.org/10.1145/3328526.3329605. 12, 16

[38] Varun Gupta, Christopher Jung, Georgy Noarov, Mallesh M. Pai, and Aaron Roth. Online multivalid learning: Means, moments, and prediction intervals. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 82:1–82:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi: 10.4230/LIPIcs.ITCS.2022.82. URL https://doi.org/10.4230/LIPIcs.ITCS.2022.82. 206

[39] Sara Hajian and Josep Domingo-Ferrer. A methodology for direct and indirect discrimination prevention in data mining. *IEEE transactions on knowledge and data engineering*, 25(7):1445–1459, 2012. 7

[40] Moritz Hardt and Guy N Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 61–70. IEEE, 2010. 144

[41] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3315–3323, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/9d2682367c3935defcb1f9e247a97c0d-Abstract.html. 1, 7

[42] Sergiu Hart. Calibrated forecasts: The minimax proof. *oral communication*, 2020. URL http://www.ma.huji.ac.il/~hart/papers/calib-minmax.pdf. 188, 192

[43] Úrsula Hébert-Johnson, Michael P. Kim, Omer Reingold, and Guy N. Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1944–1953.

PMLR, 2018. URL http://proceedings.mlr.press/v80/hebert-johnson18a.html. 4, 7, 12, 118, 119, 120, 124, 125, 126, 127, 130, 133, 134, 144, 152, 186, 187, 189, 190, 193, 197, 198, 206

[44] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963. 34

[45] Christina Ilvento. Metric learning for individual fairness. In Aaron Roth, editor, *1st Symposium on Foundations of Responsible Computing, FORC 2020, June 1-3, 2020, Harvard University, Cambridge, MA, USA (virtual conference)*, volume 156 of *LIPIcs*, pages 2:1–2:11. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi: 10.4230/LIPIcs.FORC.2020.2. URL https://doi.org/10.4230/LIPIcs.FORC.2020.2. 12, 77

[46] Shahin Jabbari, Matthew Joseph, Michael Kearns, Jamie Morgenstern, and Aaron Roth. Fairness in reinforcement learning. In *International conference on machine learning*, pages 1617–1626. PMLR, 2017. 13

[47] Prateek Jain, Brian Kulis, Inderjit S Dhillon, and Kristen Grauman. Online metric learning and fast similarity search. In *Advances in neural information processing systems*, pages 761–768, 2009. 13

[48] Rodolphe Jenatton, Jim Huang, and Cédric Archambeau. Adaptive algorithms for online convex optimization with long-term constraints. In *International Conference on Machine Learning*, pages 402–411. PMLR, 2016. 48

[49] Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. *Advances in neural information processing systems*, 29, 2016. 13, 127

[50] Matthew Joseph, Michael J. Kearns, Jamie Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 325–333, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/eb163727917cbba1eea208541a643e74-Abstract.html. 1, 8, 13, 76

[51] Matthew Joseph, Michael Kearns, Jamie Morgenstern, Seth Neel, and Aaron Roth. Meritocratic fairness for infinite and contextual bandits. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 158–163, 2018. 127

[52] Christopher Jung, Katrina Ligett, Seth Neel, Aaron Roth, Saeed Sharifi-Malvajerdi, and Moshe Shenfeld. A new analysis of differential privacy's generalization guarantees. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. 144

[53] Christopher Jung, Changhwa Lee, Mallesh Pai, Aaron Roth, and Rakesh Vohra. Moment multicalibration for uncertainty estimation. In *Conference on Learning Theory*, pages 2634–2678. PMLR, 2021. 190, 194

[54] Anson Kahng, Min Kyung Lee, Ritesh Noothigattu, Ariel D. Procaccia, and Christos-Alexandros Psomas. Statistical foundations of virtual democracy. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pages 3173–3182, 2019. 2

[55] Sham M Kakade and Dean P Foster. Deterministic calibration and nash equilibrium. In *International Conference on Computational Learning Theory*, pages 33–48. Springer, 2004. 193

[56] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowl. Inf. Syst.*, 33(1):1–33, 2011. doi: 10.1007/s10115-011-0463-8. URL https://doi.org/10.1007/s10115-011-0463-8. 1, 7

[57] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. An empirical study of rich subgroup fairness for machine learning. In *Proceedings of the conference on fairness, accountability, and transparency*, pages 100–109, 2019. 127, 195

[58] Michael J Kearns and Umesh Virkumar Vazirani. *An introduction to computational learning theory*. MIT press, 1994. 105

[59] Michael J. Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 2569–2577. PMLR, 2018. URL http://proceedings.mlr.press/v80/kearns18a.html. 1, 2, 4, 7, 12, 75, 97, 124, 127, 152, 195

[60] Michael Kim, Omer Reingold, and Guy Rothblum. Fairness through computationally-bounded awareness. *Advances in Neural Information Processing Systems*, 31, 2018. 12, 77, 127, 195

[61] Michael P Kim, Amirata Ghorbani, and James Zou. Multiaccuracy: Black-box post-processing for fairness in classification. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 247–254, 2019. 124, 126, 127, 152, 193

[62] Jyrki Kivinen and Manfred K. Warmuth. Exponentiated gradient versus gradient descent for linear predictors. *Information and Computation*, 132:1–63, 1997. 83, 87, 90

[63] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*, 2016. 1, 2, 7, 193

[64] Felicitas Kraemer, Kees Van Overveld, and Martin Peterson. Is there an ethics of algorithms? *Ethics and information technology*, 13(3):251–260, 2011. 2

[65] Brian Kulis et al. Metric learning: A survey. *Foundations and Trends® in Machine Learning*, 5(4):287–364, 2013. 13

[66] Preethi Lahoti, Krishna P. Gummadi, and Gerhard Weikum. Operationalizing individual fairness with pairwise fair representations. *CoRR*, abs/1907.01439, 2019. URL http://arxiv.org/abs/1907.01439. 77

[67] Min Kyung Lee, Daniel Kusbit, Anson Kahng, Ji Tae Kim, Xinran Yuan, Allissa Chan, Daniel See, Ritesh Noothigattu, Siheon Lee, Alexandros Psomas, and Ariel D. Procaccia. Webuildai: Participatory framework for algorithmic governance. *Proc. ACM Hum. Comput. Interact.*, 3(CSCW):181:1–181:35, 2019. 2

[68] Ehud Lehrer. Any inspection is manipulable. *Econometrica*, 69(5):1333–1347, 2001. 125, 193

[69] Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523):1094–1111, 2018. 191, 192

[70] Yang Liu, Goran Radanovic, Christos Dimitrakakis, Debmalya Mandal, and David C Parkes. Calibrated fairness in bandits. *arXiv preprint arXiv:1707.01875*, 2017. 13

[71] Ilan Lobel, Renato Paes Leme, and Adrian Vladu. Multidimensional binary search for contextual decision-making. In *Proceedings of the 2017 ACM Conference on Economics and Computation, EC '17, Cambridge, MA, USA, June 26-30, 2017*, page 585, 2017. doi: 10.1145/3033274.3085100. URL http://doi.acm.org/10.1145/3033274.3085100. 11, 13, 19, 20, 22, 23

[72] Mehrdad Mahdavi, Rong Jin, and Tianbao Yang. Trading regret for efficiency: online convex optimization with long term constraints. *J. Mach. Learn. Res.*, 13:2503–2528, 2012. URL http://dl.acm.org/citation.cfm?id=2503322. 48, 49

[73] Arvind Narayanan. Translation tutorial: 21 fairness definitions and their politics. In *Proc. Conf. Fairness Accountability Transp., New York, USA*, volume 1170, 2018. 1

[74] Seth Neel, Aaron Roth, and Zhiwei Steven Wu. How to use heuristics for differential privacy. *arXiv preprint arXiv:1811.07765*, 2018. 111, 112

[75] Ritesh Noothigattu, Snehalkumar (Neil) S. Gaikwad, Edmond Awad, Sohan Dsouza, Iyad Rahwan, Pradeep Ravikumar, and Ariel D. Procaccia. A voting-based system for ethical decision making. In *Proceedings of the 32nd Conference on Artificial Intelligence, (AAAI)*, pages 1587–1594, 2018. 2

[76] David Oakes. Self-calibrating priors do not exist. *Journal of the American Statistical Association*, 80(390):339–339, 1985. 125, 193

[77] Robin Pemantle and Yuval Peres. Concentration of lipschitz functionals of determinantal and other strong rayleigh measures. *Combinatorics, Probability and Computing*, 23(1):140–160, 2014. 111, 112

[78] Thomas K Philips and Randolph Nelson. The moment bound is tighter than chernoff's bound for positive tail probabilities. *The American Statistician*, 49(2):175–178, 1995. 156

[79] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. *arXiv preprint arXiv:1709.02012*, 2017. 4, 193

[80] Mingda Qiao and Gregory Valiant. Stronger calibration lower bounds via sidestepping. *arXiv preprint arXiv:2012.03454*, 2020. 194

[81] Guy N. Rothblum and Gal Yona. Probably approximately metric-fair learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 5666–5674, 2018. URL http://proceedings.mlr.press/v80/yona18a.html. 11, 76, 127

[82] Alvaro Sandroni, Rann Smorodinsky, and Rakesh V Vohra. Calibration with many checking rules. *Mathematics of operations Research*, 28(1):141–153, 2003. 125, 186, 193

[83] Jeanette P Schmidt, Alan Siegel, and Aravind Srinivasan. Chernoff–hoeffding bounds for applications with limited independence. *SIAM Journal on Discrete Mathematics*, 8(2):223–250, 1995. 156

[84] Alexander Schrijver. *Theory of Linear and Integer Programming*. John Wiley & Sons, Inc., USA, 1986. ISBN 0471908541. 238

[85] Eliran Shabat, Lee Cohen, and Yishay Mansour. Sample complexity of uniform convergence for multicalibration. *arXiv preprint arXiv:2005.01757*, 2020. 126, 193

[86] Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(Mar):371–421, 2008. 117, 126, 184, 194

[87] Saeed Sharifi-Malvajerdi, Michael Kearns, and Aaron Roth. Average individual fairness: Algorithms, generalization and experiments. In *Advances in Neural Information Processing Systems*, pages 8242–8251, 2019. 127

[88] Maurice Sion et al. On general minimax theorems. *Pacific Journal of mathematics*, 8(1):171–176, 1958. 83

[89] Arun Sai Suggala and Praneeth Netrapalli. Online non-convex learning: Following the perturbed leader is optimal. *CoRR*, abs/1903.08110, 2019. URL http://arxiv.org/abs/1903.08110. 42, 45

[90] Vasilis Syrgkanis, Akshay Krishnamurthy, and Robert E. Schapire. Efficient algorithms for adversarial contextual learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 2159–2168, 2016. URL http://proceedings.mlr.press/v48/syrgkanis16.html. 42, 45, 53, 59, 60, 62, 70

[91] Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in Neural Information Processing Systems*, 32:2530–2540, 2019. 195

[92] Sahil Verma and Julia Rubin. Fairness definitions explained. In Yuriy Brun, Brittany Johnson, and Alexandra Meliou, editors, *Proceedings of the International Workshop on Software Fairness, FairWare@ICSE 2018, Gothenburg, Sweden, May 29, 2018*, pages 1–7. ACM, 2018. doi: 10.1145/3194770.3194776. URL https://doi.org/10.1145/3194770.3194776. 1

[93] Rakesh V Vohra. *Advanced mathematical economics*. Routledge, 2004. 267

[94] Peng-Jun Wan, Ding-Zhu Du, Panos Pardalos, and Weili Wu. Greedy approximations for minimum submodular cover with submodular cost. *Computational Optimization and Applications*, 45(2):463–474, 2010. 182, 183

[95] Pak-Hang Wong. Democratizing algorithmic fairness. *Philosophy & Technology*, 33: 225–244, 2020. doi: 10.1145/3290605.3300830. URL http://doi.acm.org/10.1145/3290605.3300830. 2

[96] Ariana Yaptangco. Male tennis pros confirm serena's penalty was sexist and admit to saying worse on the court. *Elle*, 2018. https://www.elle.com/culture/a23051870/male-tennis-pros-confirm-serenas-penalty-was-sexist-and-admit-to-saying-worse-on-the-court/. 71

[97] Hao Yu and Michael J. Neely. A low complexity algorithm with o($\sqrt{\text{t}}$) regret and O(1) constraint violations for online convex optimization with long term constraints. *J. Mach. Learn. Res.*, 21:1:1–1:24, 2020. URL http://jmlr.org/papers/v21/16-494.html. 48

[98] Hao Yu, Michael J. Neely, and Xiaohan Wei. Online convex optimization with stochastic constraints. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1428–1438, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/da0d1111d2dc5d489242e60ebcbaf988-Abstract.html. 48

[99] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich, editors, *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, pages 1171–1180. ACM, 2017. doi: 10.1145/3038912.3052660. URL https://doi.org/10.1145/3038912.3052660. 1, 7

[100] Rich Zemel, Yu Wu, Kevin Swersky, Toni Pitassi, and Cynthia Dwork. Learning fair representations. In *International conference on machine learning*, pages 325–333. PMLR, 2013. 12, 77

[101] Shengjia Zhao, Tengyu Ma, and Stefano Ermon. Individual calibration with randomized forecasting. In *International Conference on Machine Learning*, pages 11387–11397. PMLR, 2020. 125

[102] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, 2003. 83, 87, 88, 133, 134