

Aaron Roth

***Uncertain: Modern Topics
in Uncertainty Estimation***

INCOMPLETE WORKING DRAFT

Contents

1	Introduction	1
1.1	A Guide to the Monograph	3
2	Basic Setting and Definitions	5
2.1	In More Detail	6
2.1.1	The Batch Setting	7
2.1.2	The Sequential Setting	7
3	A Simple Goal: Marginal Estimation	9
3.1	Means	9
3.2	Quantiles	12
3.2.1	Generalizing From Data	18
3.3	Sequential Prediction	20
4	Conformal Prediction	29
4.1	Prediction Sets and Nonconformity Scores	29
4.1.1	Non-Conformity Scores	31
4.2	Marginal Coverage in Expectation	33
4.3	Calibration Set Conditional Bounds	35
4.4	Sequential Conformal Prediction	36
4.4.1	Sequential Marginal Coverage Guarantees	37
4.5	Things to Reflect On	38
5	Calibration	41
5.1	Introduction to Calibration	41
5.2	Calibrating a Model f	44
5.3	Interlude: What is Calibration Good For?	47
5.3.1	A Simple Model for Decision Making	47
5.4	Quantile Calibration	51
5.5	Sequential Prediction	53
5.5.1	Sequential (Mean) Calibration	54
5.5.2	Sequential Quantile Calibration	60
6	Multigroup Guarantees	67
6.1	Group Conditional Mean Consistency	68
6.2	Group Conditional Quantile Consistency	70

6.2.1	A More Direct Approach to Group Conditional Guarantees	72
6.2.1.1	Generalization	74
6.3	Multicalibration: Group Conditional Calibration	83
6.3.1	What is Multicalibration Good For?	87
6.4	Quantile Multicalibration	89
6.5	Out of Sample Generalization	92
6.5.1	Mean Multicalibration	92
6.5.2	Quantile Multicalibration	100
6.6	Loss Minimization with Group Conditional Accuracy Guarantees	101
6.6.1	Reducing Multi-Group Optimal Learning to (Marginal) PAC Learning	103
6.7	Sequential Prediction	106
6.7.1	Achieving Bucketed Calibration	106
6.7.2	Obtaining Bucketed Quantile Multicalibration	113
7	Interlude: Online Adversarial Optimization and the Minimax Theorem	119
7.1	Basic Definitions	119
7.2	The Multiplicative Weights Algorithm	121
7.2.1	Using Multiplicative Weights for Convex Optimization	123
7.3	Zero Sum Games	126
7.4	From Sequential Decision Making to The Minimax Theorem	127
7.5	Passing Arbitrary Sequential Statistical Tests	130
7.5.1	A Model	131
7.5.2	No Non-Trivial Statistical Tests	133
7.5.3	Implications	135
8	Efficient Sequential Adversarial Learning with Conditional Guarantees	137
8.1	Computing Minimax Strategies	138
8.2	A General Framework for Multiobjective Sequential Learning	141
8.3	Conditional Regret Guarantees	146
8.3.1	Action Independent Subsequences	149
8.3.1.1	Adaptive Regret	151
8.3.1.2	Group-wise Regret	152
8.3.2	General Subsequences	153
8.3.2.1	Swap Regret	156
8.3.2.2	Mixing and Matching Guarantees	157
8.4	Sequential Adversarial Multicalibration	157
8.4.1	A Bucketed Calibration Definition	158
8.4.2	Bounding Sequential Multicalibration Error	159

9 Multicalibration for Real Valued Functions: When Does Multicalibration Imply Accuracy?	165
9.1 Beyond Groups	165
9.2 Algorithmically Reducing Multicalibration to Regression . . .	169
9.3 Weak Learning, Multicalibration, and Boosting	172
10 Conformal Prediction	179
10.1 Prediction Sets and Nonconformity Scores	179
11 Distribution Shift	181
11.1 Likelihood Ratio Reweighting	181
11.2 Multicalibration under Distribution Shift	184
11.3 Why Calibration Under Distribution Shift is Useful	186
12 Sufficient Statistics for Optimization	191
12.1 Omnipredictors: Sufficient Statistics for Unconstrained Optimization	192
12.2 Sufficient Statistics for Constrained Optimization	197
12.2.1 Convex Optimization	198
12.2.2 f -estimated Optimization	200
12.2.3 Solving Optimization Problems Without Labelled Data	201
13 Ensembling, Model Multiplicity, and the Reference Class Problem	205
13.1 Reference Classes and Model Multiplicity	205
13.2 Model Ensembling	206
13.3 Sample Complexity	211
Bibliography	217
A Useful Probabilistic Inequalities	221
B Additional Tools for Online Convex Optimization	223
B.1 Large Action Spaces: Online Linear Optimization	223
B.2 Follow the Regularized Leader and Online Gradient Descent	229
B.3 Online Convex Optimization	233
B.4 From the Minimax Theorem to Sequential Decision Making	235



1

Introduction

CONTENTS

1.1	A Guide to the Monograph	3
-----	--------------------------------	---

When a life insurance adjuster, in calculating a premium to offer Alice, determines that there is a 3% chance that she will die within the next 12 months, what does such an estimate mean? Thinking about this question gets to the heart of a mismatch between how the foundations of probability theory are constructed, and how it is applied in the real world.

In the idealized world you may have learned about in an introductory probability theory class, there is a *distribution* from which samples can be drawn repeatedly and independently. Under this distribution, (many) random variables have expectations, and events have probabilities. Generally we do not have direct access to the distribution, but we can estimate these distributional parameters by sampling from it. For example, what is the probability that the quarter in my hand comes up heads? The hypothesized distribution (which abstracts away messy details of biology and physics) encodes some platonic ideal of this probability, and I can estimate it by repeatedly sampling from the distribution—flipping the coin—and recording the proportion of times that it comes up heads.

But the situation with Alice is different. After all, Alice has but one life to live, so we have no way of “repeatedly flipping the coin” to estimate the chance that she will die within the next 12 months. This isn’t just a question of inability to sample — this is an event that will only occur once even in principal, and the more one thinks about it, the less clear it is that there really is any sampling space at all that even makes the idea of this conditional probability coherent. What is the randomness that such a probability would be taken over? These kinds of events are called “individual probabilities”, and without making very strong assumptions they are fundamentally unknowable.

And yet. Life insurance adjusters estimate the probability that their clients will die within the next 12 months all the time, and make money doing so. Weather forecasters make a career of predicting the probability of rain tomorrow, even though there is only one tomorrow. In fact, any prediction problem that is defined on a large enough feature space so that you are unlikely to see any single example more than once faces this same “individual probability” difficulty. So what is going on?

In practical terms, the only things we can measure are averages (and related statistics) over sufficiently large subsets of our data, and so most guarantees are *marginal* — i.e. they correspond to averages over many different examples in our dataset or sampled from a distribution. For example, even though we cannot meaningfully estimate the probability that Alice will die within the next 12 months, we can estimate various error metrics of a model that we derive over the population. For the purposes of making money in life insurance, this might be good enough.

But marginal guarantees — averages over the whole population — can break down when you zoom in on structured sub-populations of interest. For example, the mortality model of our life insurance adjuster might be very well *calibrated* overall — informally, not systematically biased upwards or downwards given its predictions, as averaged over the population — but might be systematically biased upwards for middle aged professors living in Philadelphia. This need not violate marginal calibration guarantees if it is systematically biased downwards on some other demographic group.

There are many reasons we might care about these kinds of *conditional* rather than *marginal* guarantees. It might, for example, lead to equity concerns: systematic biases in mortality tables can lead to certain groups being systematically over-charged for life insurance! It is also a problem for using probabilistic estimates to inform decision making. In deciding whether or not to buy a year's worth of life insurance at a certain cost c , it would be useful for Alice to know the probability p that she will need it. Here, what is relevant to Alice's decision making is the probability p that she, Alice, will die in the next 12 months — not the marginal accuracy or calibration error of a mortality model averaged over everyone. If Alice is atypical — perhaps part of a medically relevant demographic group that the model has failed to learn, then using the model's estimate in (say) an expected utility calculation will lead to making the wrong decision.

Relying on *marginal* guarantees also makes our estimates vulnerable to (inevitable) distribution shift. If an insurance adjuster was somehow able to compute the true “individual” probabilities that each person would die within the next 12 months, they would be able to profitably sell life insurance to any set of individuals that they might encounter. On the other hand, if on the basis of the business they have been doing in Pennsylvania, they manage to find a high quality mortality model that is very well calibrated on the distribution of customers they have been serving, will they be able to easily use this model in California? Its not clear: because both the population living in California and the competitive market for life insurance in California will be different, they will inevitably encounter a customer base that differs in various ways from their Pennsylvania customers. It may be (and marginal calibration guarantees do not preclude) that their model will fail to be calibrated on this new distribution, which might mean that using their model to price life insurance will be a money losing proposition in California.

We may even end up with existential angst about the nature of prediction,

and the extent to which our models can be falsified by data. If individual probabilities are unknowable, and models are simply automating the estimation of a very large ensemble of individual probabilities, might it be that we can have two different models, that are both equally well supported by the data (e.g. have the same *marginal* error over the population and different subsets we can think of, are well calibrated over the population and different subsets we can think of, and pass any other statistical test at our disposal), and yet still end up frequently making different predictions from each other? If this is the case, then we might start worrying about acting on our predictions because of due process grounds. How can I justify e.g. denying Alice a loan on the basis of a prediction if there is another equally well supported model that would have made a different prediction that would have caused me to grant her the loan? If this situation arises frequently, how can we defend our predictions — and hence our actions — against the charge of arbitrariness?

These are some of the questions that we will investigate in this monograph. A common theme is that individual guarantees will be an impossible goal, and that marginal guarantees are often too weak to be useful — but that it will nevertheless be possible to meaningfully interpolate between these two extremes. And these meaningful interpolations — guarantees that hold *conditionally* on some, but not all information available to us — can be tailor made to particular downstream applications so as to be able to give us guarantees about e.g. decision making and distribution shift. We will find that although we cannot learn individual probabilities from data, we also cannot disagree about which of a set of candidate models is “correct”, because data is always sufficient to *compare* and falsify at least one model in a set of distinct models — so although we might never be able to learn the truth, at least we cannot disagree with one another in good faith. And we will study all of these problems both under strong but convenient non-parametric assumptions — that data is drawn iid from a fixed (but unknown) distribution — as well as in fully worst-case, adversarial environments, in which we make no assumptions at all about how data is generated or evolves over time. We’ll find that quite remarkably we can accomplish all of our goals in this worst-case setting.

1.1 A Guide to the Monograph

- In Chapter 2 we define some of the basic objects and settings we will be working with: Features, labels, distributions, means, and squared error, as well as the “batch” or “distributional” learning setting as well as the “sequential” or “online adversarial” learning settings.
- In Chapter 3



2

Basic Setting and Definitions

CONTENTS

2.1	In More Detail	6
2.1.1	The Batch Setting	7
2.1.2	The Sequential Setting	7

In this Chapter we will lay out two basic models that we will study in this monograph, together with some simple definitions that will underlie our study. A guiding principle throughout our study is that we would like to make *as few assumptions as possible*, so as to minimize the number of ways in which our assumptions might fail to hold! Of course we will need to make certain assumptions, and we will be transparent about them. At a high level, we will study two settings in this monograph:

1. The *Batch* or *Distributional* setting. Here we will imagine that there is some underlying joint distribution on features and labels — although we will assume nothing about the form of this distribution or the relationships it encodes between features and labels. We will assume that to train models, we have access to a set of data points sampled identically and independently (i.i.d.) from this distribution, and our goal will be to, from this data, learn predictive models that have good properties on the underlying distribution. The primary assumption we will make here is that at “test time” (i.e. when the model is deployed), it will face examples that are sampled i.i.d. from the same distribution that the model was trained on. This can be quite a big assumption of course, and one that we will eventually attempt to relax. But our approach will always be “non-parametric” in the sense that beyond assuming the existence of such a distribution, we will not assume that it has any particular simple (“parametric”) form.
2. The *Sequential* or *Online Adversarial* setting. Here we will dispense with all distributional assumptions, and instead assume that the algorithm encounters examples in an *arbitrary sequence*. In rounds t , the algorithm will observe features x_t , at which point it will need to commit to some prediction, only after which does it learn the corresponding outcome y_t . We assume that the sequence can be

generated in a worst-case way — even by an adversary who can adapt to the choices of our algorithm and whose goals are entirely opposed to our own. Our goals will be to make predictions so that on the empirical distribution on features, predictions, and labels in hindsight, our predictions have “good properties” — the very same set of good properties that we will aim to achieve in the Distributional setting. Our aim in modelling the sequence as adversarial is not necessarily that we believe we will be operating in truly adversarial environments (although sometimes we might be), but rather that this setting is more general than all other more benign kinds of distribution shift, and so positive results in this setting will carry over quite generally to less adversarial settings. Note that in this model there is no separation between training and deployment — the algorithm is making predictions in sequence, and will have to constantly adapt to changes in the sequence. The primary assumption in this setting is that at deployment time, our model will have access to the true labels y_t shortly after making its predictions at round t . Of course this assumption will not be satisfied in all applications, but it is easy to see that without observing outcomes, we cannot give the kinds of adversarial guarantees that necessarily need to adapt to changing relationships between features and labels.

In the following we make some of these modelling assumptions more precise.

2.1 In More Detail

We will consider prediction tasks over a domain $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$. Here \mathcal{X} represents the *feature* domain and \mathcal{Y} represents the label domain. Depending on the setting, the label domain might be real valued in either one or more dimensions ($\mathcal{Y} = \mathbb{R}, \mathcal{Y} = \mathbb{R}^d$)—the *regression* setting, binary valued ($\mathcal{Y} = \{0, 1\}$)—the *binary classification* setting, or consist of some larger finite unordered set—the *multiclass classification* setting. Frequently we will consider the regression setting in which the label domain is rescaled to the unit interval $\mathcal{Y} = [0, 1]$, understanding that the bounds we prove will carry over to intervals on different ranges up to scaling parameters.

We will sometimes (in the Batch/Distributional setting) assume the existence of a distribution $\mathcal{D} \in \Delta\mathcal{Z}$; here $\Delta\mathcal{Z}$ is the notation we use to represent the set of all distributions over \mathcal{Z} . Given such a distribution, we will write $\mathcal{D}_{\mathcal{X}}$ to denote the marginal distribution over features: $\mathcal{D}_{\mathcal{X}} \in \Delta\mathcal{X}$ induced by \mathcal{D} . We will write $\mathcal{D}_{\mathcal{Y}}(x) \in \Delta\mathcal{Y}$ to denote the conditional distribution over labels induced by \mathcal{D} when we condition on a particular feature vector x . $\mathcal{D}_{\mathcal{Y}}(x)$ cap-

tures all of the information about the label that is contained by the feature vector x , and is frequently the object that we are trying to approximate with our models and uncertainty quantification. Note that consistent with this notation is the possibility that x may fully determine y — in which case $\mathcal{D}_y(x)$ will be simply a point mass on the corresponding label. A model is just some function $f : \mathcal{X} \rightarrow [0, 1]$, and our (typically unattainable goal) is to find a model f^* that has the property that for all $x \in \mathcal{X}$, $f^*(x) = \mathbb{E}_{y \sim \mathcal{D}_y(x)}[y]$ is the *conditional label expectation given x* .

Suppose we try and solve this problem and come up with some model f . How can we evaluate whether f is any good? If we are in a regression setting and our goal is purely prediction, we might evaluate f via its *squared error* — i.e. the expected (squared) deviation of its prediction from the true label. This is the objective we would minimize if we were solving (e.g.) a least squares regression problem:

Definition 1 (Squared Error) *The squared error (also known as Brier score) of a predictor f on a distribution \mathcal{D} is:*

$$B(f, \mathcal{D}) = \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x) - y)^2]$$

We will sometimes elide the distribution \mathcal{D} when it is clear from context.

Squared error will be focal in many of our derivations, and as we will see, is closely related to the *mean* of a distribution in the sense that if one must make a constant prediction, then the squared error is minimized by predicting the mean of the label distribution. When we are interested in statistics other than means, we will make use of other loss functions.

2.1.1 The Batch Setting

In the *batch* setting, we are given a *batch* or *sample* of n datapoints D sampled i.i.d. from \mathcal{D} , which we will write as $D \in \mathcal{Z}^n$. We will want algorithms that use D to learn something useful about \mathcal{D} .

We will sometimes treat a sample D as if it is a distribution: sampling from it, taking expectations over it, etc. When we do this, we are identifying D with the discrete distribution that places weight $1/n$ on each example $(x, y) \in D$. For example, we can compute the squared error of a predictor over a sample D which evaluates to:

$$B(f, D) = \frac{1}{n} \sum_{(x,y) \in D} (f(x) - y)^2$$

2.1.2 The Sequential Setting

In the *sequential* setting, data is revealed to the algorithm one example at a time, and the algorithm must make predictions before learning the label

of each point. We will not always assume that the data is drawn from a distribution — often we will assume nothing about the data generation process at all, which might even be adversarial. In such cases our goals will pertain to the empirical performance of the predictions. At a high level the setting proceeds as follows, in rounds $t \in \{1, \dots, T\}$.

1. The *adversary* chooses a (distribution over) feature vectors $x_t \in \mathcal{X}$ and labels $y_t \in \mathcal{Y}$. The realized feature vector x_t is shown to the *learner*, but not the label.
2. The learner makes some prediction p_t .
3. Finally the learner observes the realized label y_t .

Here the prediction p_t could be anything — it could try to predict the label itself, or the label mean (in the case in which $\mathcal{Y} \subset \mathbb{R}^d$), or it could be a prediction *set*. We'll be more specific about our goals as we proceed.

An interaction for T rounds generates a *transcript* π , which just encodes the sequence of examples and predictions across the T rounds: $\pi = \{(x_t, p_t, y_t)\}_{t=1}^T$.

Any quantity that we can evaluate over a distribution \mathcal{D} , we can evaluate over a transcript π of length T simply by treating it as an empirical distribution over T points. For example, in the case of real valued labels and predictions we can compute the squared error of our predictions as:

$$B(\pi) = \frac{1}{T} \sum_{t=1}^T (p_t - y_t)^2$$

In general, we will be interested in the same goals in the sequential and distributional settings — its just that in the sequential setting, we will phrase our goals as distributional goals that we want to hold *ex-post*, in the worst case over adversaries, on the empirical distribution of the transcript that results from our algorithm interacting with the adversary.

3

A Simple Goal: Marginal Estimation

CONTENTS

3.1	Means	9
3.2	Quantiles	11
3.2.1	Generalizing From Data	18
3.3	Sequential Prediction	20
	References and Further Reading	27

We introduce the problem of learning a model that is faithful to the distribution in some formal sense with a goal that is extremely weak — too weak, on its face — in part as a straw man that will focus our attention on how we can meaningfully ask for stronger guarantees. Our initial aim will only be to find a model that matches the mean of a distribution. But we will also see that marginal guarantees like these are widely used for estimating other properties of a distribution — especially quantiles. We will talk about this at length when we get to *conformal prediction*, and we will think about ways in which we can strengthen those guarantees as well.

3.1 Means

Recall that in a regression setting in which $\mathcal{Y} \subseteq [0, 1]$, our goal is to learn a model f^* such that $f^*(x) = \mathbb{E}_{y \sim \mathcal{D}(x)}[y]$ — i.e. that correctly captures the conditional label mean for each $x \in \mathcal{X}$. The problem is, learning such a function from data sampled from the distribution (or even checking whether a given model has this property) is impossible! To see this, imagine that $\mathcal{D}_{\mathcal{X}}$ is a uniform distribution over an infinitely large domain; if infinities bother you, it will suffice that the domain is sufficiently larger than the number of samples we can reasonably take from the distribution. In such a setting, we will never observe exactly the same features x repeated in a finite set of samples. Thus, the following two settings defined over a binary label domain $\mathcal{Y} = \{0, 1\}$ will induce the same distribution of observations:

1. The label \mathcal{Y} is sampled i.i.d. from $\{0, 1\}$ *after* the features x are fixed — i.e. $\mathcal{D}(x)$ is uniform over $\{0, 1\}$ for all x .
2. For each x , $\mathcal{D}(x)$ is a point-mass on either 0 or 1 (i.e. conditioning on the features deterministically fixes the label), but which one is sampled uniformly from $\{0, 1\}$ for each x .

i.e. the only difference between settings 1 and 2 are when we flip the coins: In setting 1, first we sample x , and then we flip a coin to determine the label. In setting 2, we flip all of the coins up front — this is what determines the distribution. But the differences only manifest themselves if we start seeing the same examples x over and over again. In setting 1, the labels will be uniform even conditional on x , but in setting 2, once we condition on x we always see the same label. In the first case, $f^*(x) = 0.5$ for all x . In the second case, $f^*(x) \in \{0, 1\}$ for all x . Yet we have no empirical method to distinguish these two cases.

So without making strong assumptions about the underlying distribution, we cannot set our sights on learning f^* . But we can begin with a minimal sanity check: *marginal mean consistency*:

Definition 2 A model $f : \mathcal{X} \rightarrow [0, 1]$ has marginal mean consistency error α if:

$$\left| \mathbb{E}_{x \sim \mathcal{D}_X} [f(x)] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y] \right| = \alpha$$

If $\alpha = 0$ we'll just say that f satisfies marginal mean consistency.

This minimal sanity check is an example of a *marginal* guarantee because it depends on f only through an unconditional expectation $\mathbb{E}[f(x)]$, rather than constraining the behavior of f conditional on any property of x . In other words, it's just an average over all inputs to f . f^* satisfies marginal mean consistency, so if our model f does not, this means that our model f must not be f^* . Of course, failure to satisfy marginal mean consistency is easy to fix: Let:

$$\Delta = \mathbb{E}_{(x,y) \sim \mathcal{D}} [y] - \mathbb{E}_{x \sim \mathcal{D}_X} [f(x)] \quad \text{and} \quad \hat{f}(x) = f(x) + \Delta.$$

It is easy to see that \hat{f} satisfies marginal mean consistency:

Lemma 3.1.1 $\hat{f}(x) = f(x) + \Delta$ satisfies marginal mean consistency.

Proof 1

$$\begin{aligned} \mathbb{E}_{x \sim \mathcal{D}_X} [\hat{f}(x)] &= \mathbb{E}_{x \sim \mathcal{D}_X} [f(x)] + \Delta \\ &= \mathbb{E}_{x \sim \mathcal{D}_X} [f(x)] + \mathbb{E}_{(x,y) \sim \mathcal{D}} [y] - \mathbb{E}_{x \sim \mathcal{D}_X} [f(x)] \\ &= \mathbb{E}_{(x,y) \sim \mathcal{D}} [y] \end{aligned}$$

as desired.

What is less obvious is that \hat{f} is more accurate than f — as measured by its squared error.

Lemma 3.1.2 Fix any distribution \mathcal{D} , let $f : \mathcal{X} \rightarrow [0, 1]$ be any model, let $\Delta = \mathbb{E}_{(x,y) \sim \mathcal{D}}[y] - \mathbb{E}_{x \sim \mathcal{D}_X}[f(x)]$, and let $\hat{f}(x) = f(x) + \Delta$. Then over the distribution \mathcal{D} :

$$B(\hat{f}, \mathcal{D}) = B(f, \mathcal{D}) - \Delta^2$$

Proof 2 We can directly compute:

$$\begin{aligned} B(f, \mathcal{D}) - B(\hat{f}, \mathcal{D}) &= \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[(f(x) - y)^2 - (\hat{f}(x) - y)^2 \right] \\ &= \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[f(x)^2 - 2f(x)y + y^2 - \hat{f}(x)^2 + 2\hat{f}(x)y - y^2 \right] \\ &= \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[f(x)^2 - 2f(x)y - (f(x) + \Delta)^2 + 2(f(x) + \Delta)y \right] \\ &= \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[f(x)^2 - 2f(x)y - f(x)^2 - 2\Delta f(x) - \Delta^2 + 2f(x)y + 2\Delta y \right] \\ &= \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[-2\Delta f(x) - \Delta^2 + 2\Delta y \right] \\ &= 2\Delta \mathbb{E}_{(x,y) \sim \mathcal{D}} [y - f(x)] - \Delta^2 \\ &= 2\Delta^2 - \Delta^2 \\ &= \Delta^2 \end{aligned}$$

So not only is it easy to fix a model that does not satisfy marginal mean consistency, it is always in our interest to do so if we care about accuracy: the fix is strictly accuracy improving (as measured by squared error).

Remark 3.1.1 A simple corollary of Lemma 3.1.2 is that amongst all constants, squared error is minimized at the mean of a distribution:

$$\arg \min_{c \in \mathbb{R}} B(c, \mathcal{D}) = \mathbb{E}_{(x,y) \sim \mathcal{D}} [y]$$

So one way you could compute the mean of a distribution is to solve a squared error regression problem: find the constant that minimizes the expected squared error on that distribution. Another way of saying this is to say that “squared error elicits the mean of a distribution”. Other loss functions elicit other properties of distributions, and this will be important. Not all distributional properties are elicited by any regression loss however, and this will also be important. In the next section we’ll study quantiles, another important distributional property that is also “elicited” by a regression loss.

3.2 Quantiles

The *mean* of a real-valued distribution may be different than its *median*, which is a point such that half of the probability mass of the distribution corresponds to points below it, and half of the probability mass corresponds to points above it. The median is an example of a *quantile* — a threshold such that an q fraction of the mass of a distribution corresponds to points below the threshold. The median is simply the q -quantile for $q = 0.5$.

Rather than asking for a model that matches the mean of a distribution marginally, we can ask for a model that matches a target quantile of a distribution marginally. For simplicity, we will assume that all marginal label distributions $\mathcal{D}(x)$ are continuous to avoid notational difficulties that arise from point masses. This will be without loss of generality up to adding (arbitrarily) small perturbations to labels if necessary.

Definition 3 Fix any $0 \leq q \leq 1$. τ is a q -quantile of a label distribution if:

$$\Pr_y[y \leq \tau] = q$$

We also write $Q(\tau) = q$.

Once again, our goal might be to produce a model $f : \mathcal{X} \rightarrow [0, 1]$ that on each input x outputs a value $f(x)$ that is an q -quantile of the conditional label distribution $\mathcal{D}(x)$. This will generally be impossible, but we can define marginal quantile consistency as a simple sanity check analogue of marginal mean consistency.

Definition 4 A model $f : \mathcal{X} \rightarrow [0, 1]$ has marginal quantile consistency error α with respect to a target quantile q if:

$$\left| \Pr_{(x,y) \sim \mathcal{D}}[y \leq f(x)] - q \right| = \alpha$$

If $\alpha = 0$ we'll say that f satisfies marginal quantile consistency for target quantile q .

Just as squared error score will play a key role in our analysis of models that aim to match distributional means, *pinball loss* will play a key role in our analysis of models that aim to match distributional quantiles.

Definition 5 The pinball loss function for target quantile q is:

$$L_q(\tau, y) = \begin{cases} (y - \tau)q & y > \tau \\ (\tau - y)(1 - q) & y \leq \tau \end{cases}$$

Given a data distribution \mathcal{D} and a function $f : \mathcal{X} \rightarrow [0, 1]$, write:

$$PB_q(f, \mathcal{D}) = \mathbb{E}_{(x,y) \sim \mathcal{D}} [L_q(f(x), y)]$$

We will sometimes elide the distribution \mathcal{D} when it is clear from context.

Observe that for $q = 1/2$, this is simply (a scaling of) the absolute value difference function: $L_{1/2}(\tau, y) = \frac{1}{2}|\tau - y|$. Just as the constant that minimizes the squared error on a distribution is its mean, the constant that minimizes the absolute loss is its median, and in general, the constant that minimizes the pinball loss for a target quantile q is a q -quantile:

Lemma 3.2.1 *For any continuous distribution over y and any $0 \leq q \leq 1$*

$$\tau_q = \arg \min_{\tau \in [0,1]} \mathbb{E}_y [L_q(\tau, y)]$$

is a q -quantile.

Proof 3 *Since the distribution is continuous, this is a continuous convex function and takes its minimum at a point that has a (sub)derivative equal to 0. Thus We can calculate a (sub)derivative of the function:*

$$\begin{aligned} \frac{d \mathbb{E}_y [L_q(\tau, y)]}{d\tau} &= \mathbb{E}_y [(1 - q) \mathbb{1}[y \leq \tau] - q \mathbb{1}[y > \tau]] \\ &= \mathbb{E}_y [\mathbb{1}[y \leq \tau] - q] \\ &= \Pr_y [y \leq \tau] - q \end{aligned}$$

Thus by inspection we see that there is a sub-derivative that takes value 0 for exactly values τ_q for which $\Pr_y [y \leq \tau_q] = q$ — i.e. q -quantiles of the distribution.

It will also be useful for us later on to have an analogue of Lemma 3.1.2 for pinball loss — i.e. a lemma that says that if we start with a model f that is far from satisfying marginal quantile consistency, and then apply a shift so that it does, that we make quantifiable progress in terms of reducing the expected pinball loss for the function.

Suppose that $f : \mathcal{X} \rightarrow [0, 1]$ has marginal quantile consistency error α . Let $\Delta \in \mathbb{R}$ be such that: $\Pr_{(x,y) \sim \mathcal{D}} [y \leq f(x) + \Delta] = q$. Such a value Δ is guaranteed to exist since we have assumed that the conditional label distributions $\mathcal{D}(x)$ are continuous, and so $\Pr_{(x,y) \sim \mathcal{D}} [y \leq f(x) + \Delta]$ is a continuous monotonically increasing function taking values in the full range $[0, 1]$. Let $\hat{f}(x) = f(x) + \Delta$. By construction, \hat{f} satisfies marginal quantile consistency with respect to target quantile q . It also has improved pinball loss. But in order to claim that the pinball loss has improved by an amount that we can bound away from 0, we will need to assume that the conditional label distributions has bounded

probability density — or equivalently that its cumulative distribution function is Lipschitz-continuous. This is a quantitative version of the qualitative assumption that the distribution is (uniformly) continuous. Once again this assumption can be enforced, if necessary, by adding noise to observed labels.

Definition 6 A conditional label distribution $\mathcal{D}(x)$ is ρ -Lipschitz continuous (or just ρ -Lipschitz) if for all $0 \leq \tau \leq \tau' \leq 1$:

$$\Pr_{y \sim \mathcal{D}(x)}[y \leq \tau'] - \Pr_{y \sim \mathcal{D}(x)}[y \leq \tau] \leq \rho(\tau' - \tau)$$

A distribution over labelled examples \mathcal{D} is ρ -Lipschitz if for each $x \in \mathcal{X}$, $\mathcal{D}(x)$ is ρ -Lipschitz.

The above definition is actually somewhat stronger than we need right now — we don't need the Lipschitz condition simultaneously for each conditional label distribution $\mathcal{D}(x)$, but only marginally over the whole distribution — but this stronger condition will be useful for us later on.

Lemma 3.2.2 Fix any distribution over labeled examples \mathcal{D} that is ρ -Lipschitz. Fix any model $f : \mathcal{X} \rightarrow [0, 1]$ that has marginal consistency error α with respect to target quantile q , and let $\hat{f}(x) = f(x) + \Delta$ with Δ chosen such that \hat{f} satisfies marginal quantile consistency for quantile q . Then:

$$PB_q(\hat{f}) \leq PB_q(f) - \frac{\alpha^2}{2\rho}$$

and

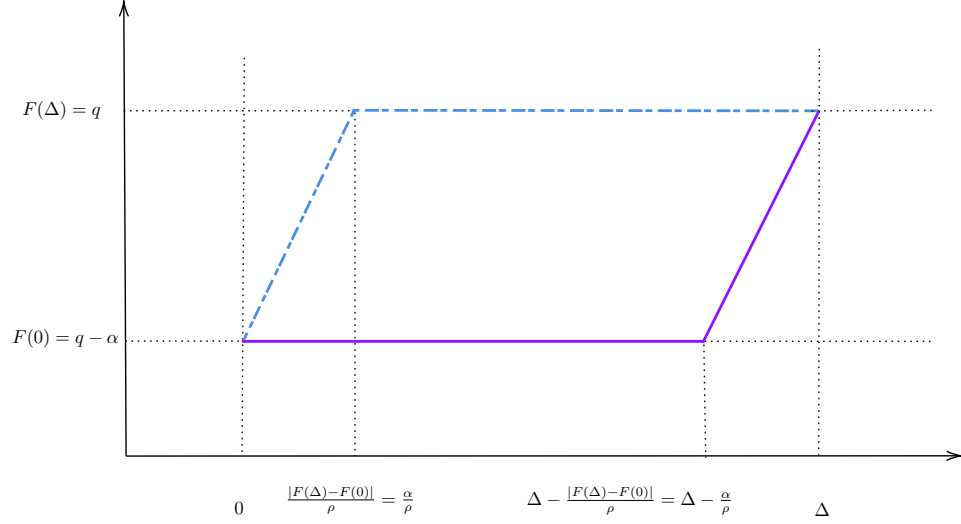
$$PB_q(f) \leq PB_q(\hat{f}) + |\Delta|\alpha - \frac{\alpha^2}{2\rho}$$

Proof 4 As in the proof of Lemma 3.2.1, we can compute:

$$\begin{aligned} \frac{dPB_q(f(x) + \tau)}{d\tau} &= \mathbb{E}_{x \sim \mathcal{D}_X} \left[\frac{d\mathbb{E}_{y \sim \mathcal{D}(x)}[L_q(f(x) + \tau, y)]}{d\tau} \right] \\ &= \mathbb{E}_{x \sim \mathcal{D}_X} \left[\Pr_{y \sim \mathcal{D}(x)}[y \leq f(x) + \tau] - q \right] \\ &= \Pr_{(x,y) \sim \mathcal{D}}[y \leq f(x) + \tau] - q \end{aligned}$$

We can now compute:

$$\begin{aligned} PB_q(\hat{f}(x)) - PB_q(f(x)) &= PB_q(f(x) + \Delta) - PB_q(f(x)) \\ &= \int_0^\Delta \frac{dPB_q(f(x) + \tau)}{d\tau} d\tau \\ &= \int_0^\Delta \left(\Pr_{(x,y) \sim \mathcal{D}}[y \leq f(x) + \tau] - q \right) d\tau \\ &= \begin{cases} \int_0^\Delta \Pr_{(x,y) \sim \mathcal{D}}[y \leq f(x) + \tau] d\tau - |\Delta|q & \Delta \geq 0 \\ \int_0^\Delta \Pr_{(x,y) \sim \mathcal{D}}[y \leq f(x) + \tau] d\tau + |\Delta|q & \Delta < 0 \end{cases} \end{aligned}$$

**FIGURE 3.1**

Upper and lower bounding the local area under the curve when $\Delta > 0$. Here $F(\tau) = \Pr_{(x,y) \sim \mathcal{D}}[y \leq f(x) + \tau]$

$\Pr_{(x,y) \sim \mathcal{D}}[y \leq f(x) + \tau]$ is a non-negative function that is increasing in τ , and so if $\Delta < 0$ (i.e. if initially $f(x)$ is over-predicting the q 'th quantile), then we have that $\int_0^\Delta \Pr_{(x,y) \sim \mathcal{D}}[y \leq f(x) + \tau] d\tau$ evaluates to the negative of the area under the CDF of the distribution between $f(x) + \Delta$ and $f(x)$.

Similarly the integral takes positive value if $\Delta > 0$ and corresponds to the area under the CDF between $f(x)$ and $f(x) + \Delta$. First we consider the case in which $\Delta > 0$. We need to bound $\int_0^\Delta \Pr_{(x,y) \sim \mathcal{D}}[y \leq f(x) + \tau] d\tau$. Here we will use the Lipschitz condition to upper bound the maximum possible area under the CDF. The worst case is that the CDF of the label distribution increases as quickly as possible at a linear rate from $q - \alpha$ to q between $\tau = 0$ and $\tau = \alpha/\rho$, and then maintains a constant value at q from $\tau = \alpha/\rho$ to $\tau = \Delta$ (See Figure 3.1). Calculating the area under this worst case curve, we have:

$$\begin{aligned}
 \int_0^\Delta \Pr_{(x,y) \sim \mathcal{D}}[y \leq f(x) + \tau] d\tau &\leq \left(\Delta - \frac{\alpha}{\rho}\right) \cdot q + \frac{\alpha}{\rho}(q - \alpha) + \frac{\alpha}{\rho} \cdot \frac{\alpha}{2} \\
 &= q\Delta - \frac{q\alpha}{\rho} + \frac{q\alpha}{\rho} - \frac{\alpha^2}{\rho} + \frac{\alpha^2}{2\rho} \\
 &= q\Delta - \frac{\alpha^2}{2\rho}
 \end{aligned}$$

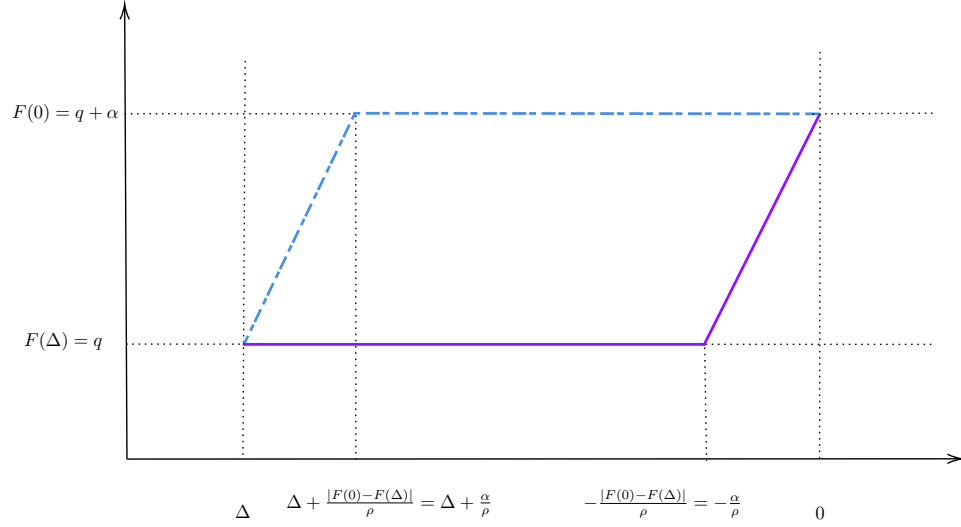


FIGURE 3.2

Upper and lower bounding the local area under the curve when $\Delta < 0$. Here $F(\tau) = \Pr_{(x,y) \sim \mathcal{D}}[y \leq f(x) + \tau]$

Combining with above, we have that:

$$PB_q(\hat{f}) - PB_q(f) \leq q|\Delta| - \frac{\alpha^2}{2\rho} - q|\Delta| = -\frac{\alpha^2}{2\rho}$$

Next we can lower bound the area under the CDF. Again by the Lipschitz condition, the smallest area under the CDF that respects the Lipschitz condition arises if the CDF remains constant taking value $q - \alpha$ from $\tau = 0$ to $\tau = \Delta - \frac{\alpha}{\rho}$ before increasing at a linear rate to q from $\tau = \Delta - \frac{\alpha}{\rho}$ to $\tau = \Delta$. See figure 3.1. In this case the area is:

$$\begin{aligned} \int_0^\Delta \Pr_{(x,y) \sim \mathcal{D}}[y \leq f(x) + \tau] d\tau &\geq \Delta(q - \alpha) + \frac{\alpha^2}{2\rho} \\ &= \Delta q - \Delta\alpha + \frac{\alpha^2}{2\rho} \end{aligned}$$

Combining with the above we have that:

$$PB_q(\hat{f}) - PB_q(f) \geq |\Delta|q - |\Delta|\alpha + \frac{\alpha^2}{2\rho} - |\Delta|q = -|\Delta|\alpha + \frac{\alpha^2}{2\rho}$$

In the remaining case in which $\Delta < 0$, our worst cases are reversed (we need to maximize the area under the curve to lower bound the integral and minimize the area under the curve to upper bound the integral). Once again, the CDF

that minimizes the area under the curve subject to the Lipschitz constraint behaves as follows (See figure 3.2): The CDF remains constant at q between $\tau = \Delta$ and $\tau = -\alpha/\rho$, before increasing as quickly as possible at a linear rate up to value $q + \alpha$ between $\tau = -\alpha/\rho$ and $\tau = 0$. In this case we have that:

$$\begin{aligned} \int_0^\Delta \Pr_{(x,y) \sim \mathcal{D}}[y \leq f(x) + \tau] d\tau &\leq -\left(|\Delta|q + \frac{\alpha^2}{2\rho}\right) \\ &= -q|\Delta| - \frac{\alpha^2}{2\rho} \end{aligned}$$

Again combining with above we get that:

$$PB_q(\hat{f}) - PB_q(f(x)) \leq -q|\Delta| - \frac{\alpha^2}{2\rho} + q|\Delta| = -\frac{\alpha^2}{2\rho}$$

Finally, the CDF that maximizes the area under the curve subject to the Lipschitz constraint increases at a linear rate from $\tau = \Delta$ to $\tau = \Delta + \alpha/\rho$ from value q to value $q + \alpha$, and then remains constant at $q + \alpha$ from $\tau = \Delta + \frac{\alpha}{\rho}$ to $\tau = 0$. Computing the area under this curve, we get:

$$\begin{aligned} \int_0^\Delta \Pr_{(x,y) \sim \mathcal{D}}[y \leq f(x) + \tau] d\tau &\geq -\left(\frac{\alpha^2}{2\rho} + |\Delta|q + \alpha\left(|\Delta| - \frac{\alpha}{\rho}\right)\right) \\ &= \frac{\alpha^2}{2\rho} - |\Delta|q - |\Delta|\alpha \end{aligned}$$

Together with the above we have that:

$$PB_q(\hat{f}) - PB_q(f(x)) \geq \frac{\alpha^2}{2\rho} - |\Delta|q - |\Delta|\alpha + q|\Delta| = \frac{\alpha^2}{2\rho} - |\Delta|\alpha$$

which completes the proof of the lemma.

So once again, if a model fails to satisfy marginal quantile consistency, it is easy to fix the model, and once again, doing so is accuracy improving — this time as measured via Pinball loss.

We'll make one more observation: we proved Lemma 3.2.2 under the assumption that the underlying distribution \mathcal{D} was ρ -Lipschitz. But eventually, when we want to apply similar arguments to algorithms run on *data* sampled from some underlying distribution, we will face the problem that the empirical distribution over a finite dataset is discrete, and hence cannot be Lipschitz at fine enough resolutions. But we observe that Lemma 3.2.2 actually only speaks to updates Δ that are applied to fix marginal consistency error of scale α — and if the underlying distribution is ρ -Lipschitz, we must have that $\Delta \geq \frac{\alpha}{\rho}$. So we really only require that the underlying distribution is Lipschitz at scales larger than $\frac{\alpha}{\rho}$. Here we define the condition we need — Lipschitzness only at large enough scales:

Definition 7 Fix $\rho, r > 0$. A conditional label distribution $\mathcal{D}(x)$ is (ρ, r) -Lipschitz continuous (or just (ρ, r) -Lipschitz) if for all $0 \leq \tau \leq \tau' \leq 1$ such that $\tau' - \tau \geq r$:

$$\Pr_{y \sim \mathcal{D}(x)}[y \leq \tau'] - \Pr_{y \sim \mathcal{D}(x)}[y \leq \tau] \leq \rho(\tau' - \tau)$$

A distribution over labelled examples \mathcal{D} is (ρ, r) -Lipschitz if for each $x \in \mathcal{X}$, $\mathcal{D}(x)$ is (ρ, r) -Lipschitz.

Using the insight that our proof of Lemma 3.2.2 really only used the condition of $(\rho, \frac{\alpha}{\rho})$ -Lipschitz continuity, we have the following lemma:

Lemma 3.2.3 Fix any $\rho, \alpha > 0$. Fix any distribution over labeled examples \mathcal{D} that is $(\rho, \frac{\alpha}{\rho})$ -Lipschitz. Fix any model $f : \mathcal{X} \rightarrow [0, 1]$ that has marginal consistency error α with respect to target quantile q , and let $\hat{f}(x) = f(x) + \Delta$ with Δ chosen such that \hat{f} satisfies marginal quantile consistency for quantile q . Then:

$$PB_q(\hat{f}) \leq PB_q(f) - \frac{\alpha^2}{2\rho}$$

and

$$PB_q(f) \leq PB_q(\hat{f}) + |\Delta|\alpha - \frac{\alpha^2}{2\rho}$$

3.2.1 Generalizing From Data

Thus far we have been acting as if we have direct access to the data distribution \mathcal{D} , and in particular, given a fixed model f can compute the quantity Δ such that $\hat{f}(x) = f(x) + \Delta$ satisfies our marginal consistency desideratum with respect to either means or quantiles. But of course we generally will not have direct access to \mathcal{D} , and will instead have only a *sample* $D \sim \mathcal{D}^n$ of n points drawn i.i.d. from \mathcal{D} . What we will generally do (now, and for more complex algorithms in later chapters) is run our algorithms on the *empirical distribution* over our sample D , and then prove that the guarantees that our algorithms have on D carry over (with small loss) to \mathcal{D} .

Theorem 1 Fix any model f and distribution \mathcal{D} , and let $D \sim \mathcal{D}^n$ consist of n samples drawn i.i.d. from \mathcal{D} . Let Δ be such that $\hat{f}(x) = f(x) + \Delta$ satisfies marginal mean consistency on D . Then with probability $1 - \delta$ over the draw of D , \hat{f} has marginal mean consistency error at most α on \mathcal{D} , for:

$$\alpha \leq \sqrt{\frac{2 \log(2/\delta)}{n}}$$

Proof 5 This is an application of Hoeffding's inequality (Theorem 67) which we quote here in its first use:

Let X_1, \dots, X_n be independent random variables bounded such that for each i , $a_i \leq X_i \leq b_i$. Let $S_n = \sum_{i=1}^n X_i$ denote their sum. Then for all $t > 0$:

$$\Pr[|S_n - \mathbb{E}[S_n]| \geq t] \leq 2 \exp\left(\frac{-2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right)$$

In our case, we have that $\Delta = \frac{1}{n} \sum_{(x,y) \in D} (y - f(x))$, and each term $\frac{1}{n}(y - f(x))$ is bounded such that:

$$-\frac{1}{n} \leq \frac{1}{n}(y - f(x)) \leq \frac{1}{n}$$

We also have that $\mathbb{E}_D \Delta = \mathbb{E}_{(x,y) \sim \mathcal{D}}[y - f(x)]$. Thus we can apply Hoeffding's inequality to conclude that:

$$\Pr\left[|\Delta - \mathbb{E}_{(x,y) \sim \mathcal{D}}[y - f(x)]| \geq t\right] \leq 2 \exp\left(\frac{-nt^2}{2}\right)$$

Setting the right hand side to be at most δ and solving for t , we find that:

$$\Pr\left[|\Delta - \mathbb{E}_{(x,y) \sim \mathcal{D}}[y - f(x)]| \geq \sqrt{\frac{2 \log(2/\delta)}{n}}\right] \leq \delta$$

Finally, recall that by definition, \hat{f} has marginal mean consistency error:

$$\begin{aligned} \left| \mathbb{E}_{(x,y) \sim \mathcal{D}}[\hat{f}(x) - y] \right| &= \left| \mathbb{E}_{(x,y) \sim \mathcal{D}}[f(x) + \Delta - y] \right| \\ &\leq \left| \Delta - \mathbb{E}_{(x,y) \sim \mathcal{D}}[y - f(x)] \right| \\ &\leq \sqrt{\frac{2 \log(2/\delta)}{n}} \end{aligned}$$

where the last inequality holds with probability $1 - \delta$, as established by Hoeffding's inequality.

Theorem 2 Fix any model f and distribution \mathcal{D} , and let $D \sim \mathcal{D}^n$ consist of n samples drawn i.i.d. from \mathcal{D} . Let Δ be such that $\hat{f}(x) = f(x) + \Delta$ has marginal quantile consistency error α' with respect to some target quantile q on D . Then with probability $1 - \delta$ over the draw of D , \hat{f} has marginal quantile consistency error at most α with respect to target quantile q on \mathcal{D} , for:

$$\alpha \leq \alpha' + \sqrt{\frac{\log(2/\delta)}{2n}}$$

Proof 6 This is an application of the DKW (Dvoretzky–Kiefer–Wolfowitz) inequality (Theorem 70) which we quote here in its first use:

Let $\mathcal{D} \in \mathcal{Z}^n$ be any distribution and let $D \sim \mathcal{D}^n$ consist of n points sampled i.i.d. from D . Let $F(c) = \Pr_{(x,y) \sim \mathcal{D}}[y \leq c]$ denote the CDF of the label distribution induced by \mathcal{D} , and let $\hat{F}_D(c) = \frac{1}{n} \sum_{(x,y) \in D} \mathbb{1}[y \leq c]$ denote the CDF of the empirical label distribution induced by D . Then for every $t > 0$:

$$\Pr \left[\sup_{c \in \mathbb{R}} |F(c) - \hat{F}_D(c)| \geq t \right] \leq 2 \exp(-2nt^2)$$

Consider the distribution \mathcal{D}' which is derived from \mathcal{D} by replacing the label y of each example (x, y) with the label $y' = y - f(x)$. We apply the DKW inequality to this distribution. By definition, Δ is chosen such that

$$\left| \Pr_{(x,y) \sim D'} [y \leq f(x) + \Delta] - q \right| = \alpha'$$

rearranging, this is:

$$|\hat{F}_D(\Delta) - q| \leq \alpha'$$

Applying the DKW inequality with $t = \sqrt{\frac{\log(2/\delta)}{2n}}$, we have that with probability $1 - \delta$:

$$|\hat{F}_D(\Delta) - F(\Delta)| \leq \sqrt{\frac{\log(2/\delta)}{2n}}$$

And so $|F(\Delta) - q| \leq \alpha' + \sqrt{\frac{\log(2/\delta)}{2n}}$. Expanding out the definition of $F(\Delta)$ we have that

$$\begin{aligned} \alpha' + \sqrt{\frac{\log(2/\delta)}{2n}} &\geq \left| \Pr_{(x,y) \sim \mathcal{D}} [y - f(x) \leq \Delta] - q \right| \\ &= \left| \Pr_{(x,y) \sim \mathcal{D}} [y \leq f(x) + \Delta] - q \right| \\ &= \left| \Pr_{(x,y) \sim \mathcal{D}} [y \leq \hat{f}(x)] - q \right| \end{aligned}$$

The result is that we can simply proceed as if our sample is our underlying distribution when we aim for marginal consistency — and our marginal consistency error on the underlying distribution is guaranteed to be larger than our empirical marginal consistency error by at most ϵ with probability $1 - \delta$, whenever $n \geq \Omega\left(\frac{\log(1/\delta)}{\epsilon^2}\right)$.

3.3 Sequential Prediction

What about when we are in a sequential prediction setting, and there is no distribution? Even when examples are selected by an adversary, we can still

talk about marginal mean and quantile consistency (and all of the other distributional measures that we will introduce in later chapters). We will always evaluate these guarantees ex-post, over the empirical distribution over the transcript.

Definition 8 Fix a transcript $\pi = \{(x_t, p_t, y_t)\}_{t=1}^T$ consisting of examples $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$ and predictions $p_t \in [0, 1]$. The transcript satisfies marginal mean consistency with error α if:

$$\left| \frac{1}{T} \sum_{t=1}^T p_t - \frac{1}{T} \sum_{t=1}^T y_t \right| = \alpha$$

It satisfies marginal quantile consistency with respect to a target quantile q and error α if:

$$\left| \frac{1}{T} \sum_{t=1}^T \mathbb{1}[y_t \leq p_t] - q \right| = \alpha$$

Our goal in sequential prediction settings will generally be to derive algorithms that guarantee that against *any adversary*, they generate a transcript π that with high probability (or with certainty) satisfies some notion of statistical consistency. An adversary is simply an (arbitrary) algorithm mapping prefixes of a transcript of the first $t - 1$ rounds to a (distribution over) outcomes (x_t, y_t) at the next round, for every t . Similarly a prediction algorithm (the thing we will want to develop) is simply an algorithm mapping prefixes of the transcript of the first $t - 1$ rounds, together with the features x_t at the current round, to a prediction p_t for the current round — once again for every t . We'll want algorithms that can make guarantees in the worst case over adversaries.

In general, solving these problems in the adversarial sequential setting is only more difficult than solving them in the batch, distributional setting, in a formal sense: If we have algorithms that promise consistency guarantees in the sequential setting, then by running them on data that is in fact drawn i.i.d. from some distribution, we can also obtain the same guarantees in the batch setting. The reverse is not true — the sequential setting is generally strictly harder.

Here we give a warm-up version of this style of theorem, just for mean marginal consistency. In this case, it is easy to more directly get the same kinds of out of sample guarantees — but more later we will see more sophisticated versions of this kind of online-to-offline reduction. It will be an application of Hoeffding's inequality, which we state as Theorem 67.

Theorem 3 Suppose we have an algorithm \mathcal{A} that when run against any adversary for T rounds generates a transcript π that satisfies marginal mean consistency with error at most α . Suppose we have some model $f : \mathcal{X} \rightarrow [0, 1]$ and a data distribution \mathcal{D} , and consider the following procedure to simulate an adversary. At each round t we:

1. Sample $(\hat{x}_t, \hat{y}) \sim \mathcal{D}$
2. Feed algorithm \mathcal{A} the sample $(x_t, y_t) = (\hat{x}, \hat{y}_t - f(\hat{x}))$

This results in some transcript $\pi = \{(x_t, p_t, y_t)\}_{t=1}^T$. Let $\Delta = \frac{1}{T} \sum_{t=1}^T p_t$ and let $\hat{f}(x) = f(x) + \Delta$. Then for any $\delta > 0$, with probability $1 - \delta$, \hat{f} satisfies marginal mean consistency with error α' for:

$$\alpha' \leq \alpha + \sqrt{\frac{2 \log(2/\delta)}{T}}$$

Proof 7 Since π is promised to satisfy marginal mean consistency with error at most α , we know that:

$$\left| \frac{1}{T} \sum_{t=1}^T p_t - \frac{1}{T} \sum_{t=1}^T y_t \right| \leq \alpha$$

Let:

$$\bar{\Delta} = \frac{1}{T} \sum_{t=1}^T (\hat{y}_t - f(\hat{x}_t))$$

Plugging in the definitions of Δ and y_t we have that:

$$|\Delta - \bar{\Delta}| \leq \alpha$$

Note also that since (\hat{x}_t, \hat{y}_t) are sampled i.i.d. from \mathcal{D} , we have that:

$$\mathbb{E}[\bar{\Delta}] = \mathbb{E}_{(x,y) \sim \mathcal{D}} [y - f(x)]$$

We can now apply Hoeffding's inequality (Theorem 67) to the quantity $\bar{\Delta} = \frac{1}{T} \sum_{t=1}^T (\hat{y}_t - f(\hat{x}_t))$. Each term in the sum is bounded between $-1/T \leq \frac{1}{T}(\hat{y}_t - f(\hat{x}_t)) \leq 1/T$ and so we have for any $\epsilon > 0$:

$$\Pr \left[\left| \bar{\Delta} - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y - f(x)] \right| \geq \epsilon \right] \leq 2 \exp \left(\frac{-2T\epsilon^2}{4} \right)$$

The right hand side is at most δ when we have:

$$\epsilon \geq \sqrt{\frac{2 \log(2/\delta)}{T}}$$

We therefore have that with probability $1 - \delta$:

$$\begin{aligned}
\left| \mathbb{E}_{(x,y) \sim \mathcal{D}} [\hat{f}(x) - y] \right| &= \left| \mathbb{E}_{(x,y) \sim \mathcal{D}} [f(x) + \Delta - y] \right| \\
&\leq \left| \mathbb{E}_{(x,y) \sim \mathcal{D}} [f(x) + \bar{\Delta} - y] \right| + \alpha \\
&= \left| \mathbb{E}_{(x,y) \sim \mathcal{D}} [f(x) + \mathbb{E}[\bar{\Delta}] + (\bar{\Delta} - \mathbb{E}[\bar{\Delta}]) - y] \right| + \alpha \\
&= \left| \mathbb{E}_{(x,y) \sim \mathcal{D}} [(\bar{\Delta} - \mathbb{E}[\bar{\Delta}])] \right| + \alpha \\
&\leq \alpha + \sqrt{\frac{2 \log(2/\delta)}{T}}
\end{aligned}$$

as desired.

Next, we'll see a simple algorithm that can guarantee marginal mean consistency with error on the order of $O(1/T)$ on *any* sequence of length T — i.e. without assuming that the data points come from a distribution. The algorithm will be silly on its face as a prediction algorithm — always predicting that today's outcome will be equal to yesterday's outcome. Its excellent performance (as measured by marginal mean consistency) tells us something about the weakness of marginal guarantees.

Algorithm 1 Online-Marginal-Mean-Predictor

```

Let  $y_0 = 0$ 
for  $t = 1$  to  $T$  do
  Observe  $x_t$  (and ignore it!)
  Predict  $p_t = y_{t-1}$ 
  Observe  $y_t$ .

```

If we imagine using this algorithm to predict weather, then what it does is the following: If it rained yesterday, it predicts a 100% chance of rain today. Otherwise it predicts a 0% chance of rain. And yet:

Theorem 4 For any sequence of examples of length T , $\{(x_t, y_t)\}_{t=1}^T$ Online-Marginal-Mean-Predictor (Algorithm 1) produces a transcript that satisfies marginal mean consistency with error α for $\alpha \leq 1/T$.

Proof 8 Using the fact that $p_t = y_{t-1}$ (and $y_0 = 0$) we compute:

$$\begin{aligned}
\left| \frac{1}{T} \sum_{t=1}^T p_t - \frac{1}{T} \sum_{t=1}^T y_t \right| &= \left| \frac{1}{T} \sum_{t=1}^T y_{t-1} - \frac{1}{T} \sum_{t=1}^T y_t \right| \\
&= \frac{1}{T} |y_0 - y_T| \\
&\leq \frac{1}{T}
\end{aligned}$$

Now let's do the same for quantiles. First we argue that obtaining marginal quantile consistency in the online setting is sufficient to obtain marginal quantile consistency on a distribution, and then show a simple deterministic algorithm for obtaining marginal quantile consistency in the online adversarial setting. There will be one major difference, which is that to convert an online sequence of predictions to an offline quantile predictor, we cannot simply average the predicted quantiles as we did with predicted means (because the relationship between the numeric value of quantiles and their inverse CDF value is not linear). Instead, we will *randomize* over the sequence of predictions, which will result in an offline randomized quantile predictor.

Theorem 5 *Suppose we have an algorithm \mathcal{A} that when run against any adversary for T rounds generates a transcript π that satisfies marginal quantile consistency with error at most α for some target quantile q . Suppose we have some model $f : \mathcal{X} \rightarrow [0, 1]$ and a data distribution \mathcal{D} , and consider the following procedure to simulate an adversary. At each round t we:*

1. Sample $(\hat{x}_t, \hat{y}_t) \sim \mathcal{D}$
2. Feed algorithm \mathcal{A} the sample $(x_t, y_t) = (\hat{x}_t, \hat{y}_t - f(\hat{x}_t))$

This results in some transcript $\pi = \{(x_t, p_t, y_t)\}_{t=1}^T$. Let Δ be the random variable that takes value in $\{p_1, \dots, p_T\}$ uniformly at random (i.e. $\Delta = p_1$ with probability $1/T$, $\Delta = p_2$ with probability $1/T$, etc.) Let $\hat{f}(x)$ be the randomized predictor defined as $\hat{f}(x) + \Delta$. Then for any $\delta > 0$, with probability $1 - \delta$, \hat{f} satisfies marginal quantile consistency with error α' with respect to target quantile q for:

$$\alpha' \leq \alpha + \sqrt{\frac{2 \ln(2/\delta)}{T}}$$

In other words:

$$\left| \Pr_{(x,y) \sim \mathcal{D}, \Delta} [y \leq f(x) + \Delta] - q \right| \leq \alpha'$$

Proof 9 *Since π is promised to satisfy marginal quantile consistency w.r.t. quantile q with error at most α , we know that:*

$$\left| \frac{1}{T} \sum_{t=1}^T \mathbb{1}[y_t \leq p_t] - q \right| \leq \alpha$$

Plugging in the definition of y_t we have that:

$$\left| \frac{1}{T} \sum_{t=1}^T \mathbb{1}[\hat{y}_t - f(\hat{x}_t) \leq p_t] - q \right| \leq \alpha$$

Let \mathcal{D}' be the label distribution induced by outputting the label $y - f(x)$ for $(x, y) \sim \mathcal{D}$ and let F denote its CDF: $F(x) = \Pr_{y \sim \mathcal{D}'} [y \leq x]$. We want to

be able to say that $\frac{1}{T} \sum_{t=1}^T F(p_t) \approx q$, but we have a problem: the indicators $\mathbb{1}[y_t \leq p_t]$ are not independent random variables even though the y_t are, since each p_t is potentially chosen as a function of all previous labels y_1, \dots, y_{t-1} . Hence we cannot apply Hoeffding's inequality. But all is not lost! We will need Azuma's inequality (Theorem 69) which we quote here before its first use:

Let X_1, \dots, X_n be random variables (not necessarily independent) bounded such that for each i , $|X_i| \leq c_i$. Let $X_{<i}$ denote the prefix X_1, X_2, \dots, X_{i-1} . Then for all $t > 0$:

$$\Pr \left[\left| \sum_{i=1}^n X_i - \sum_{i=1}^n \mathbb{E}[X_i | X_{<i}] \right| \geq t \right] \leq 2 \exp \left(\frac{-t^2}{2 \sum_{i=1}^n c_i^2} \right)$$

Recall that for a sequential prediction algorithm, p_t can be chosen as a function of past examples — but must be independent of the current example y_t . Hence we do have that $\mathbb{E}_{y_t}[\mathbb{1}[y_t \leq p_t] | y_{<t}] = F(p_t)$. For us, the random variables are $1/T(\mathbb{1}[y_t \leq p_t])$ which are bounded by $c_t = 1/T$. Thus we can apply Azuma's inequality with $t = \sqrt{\frac{2 \ln(2/\delta)}{T}}$ to conclude that:

$$\Pr \left[\left| \frac{1}{T} \sum_{i=1}^T \mathbb{1}[y_t \leq p_t] - \frac{1}{T} \sum_{t=1}^T F(p_t) \right| \geq \sqrt{\frac{2 \ln(2/\delta)}{T}} \right] \leq \delta$$

Combining this with our guarantee of marginal quantile consistency, with probability $1 - \delta$ we have that:

$$\left| \frac{1}{T} \sum_{t=1}^T F(p_t) - q \right| \leq \alpha + \sqrt{\frac{2 \ln(2/\delta)}{T}}$$

Finally we can compute:

$$\begin{aligned} \left| \Pr_{(x,y) \sim \mathcal{D}, \Delta} [y \leq f(x) + \Delta] - q \right| &= \left| \frac{1}{T} \sum_{t=1}^T \Pr_{(x,y) \sim \mathcal{D}} [y \leq f(x) + p_t] - q \right| \\ &= \left| \frac{1}{T} \sum_{t=1}^T F(p_t) - q \right| \\ &\leq \alpha + \sqrt{\frac{2 \ln(2/\delta)}{T}} \end{aligned}$$

where the last inequality holds with probability $1 - \delta$ over the draws of $\{(x_t, y_t)\}_{t=1}^T$.

Next, we give our algorithm for making predictions that satisfy online marginal quantile consistency for any target quantile q against any adversarially chosen sequence of examples. The algorithm takes as input a “learning

rate” parameter η , and can be viewed and analyzed as online gradient descent on the pinball loss. But the specific form of the resulting update also lends itself to a very simple analysis showing that its quantile error tends to 0 at a rate of $1/T$, just as our algorithm for obtaining marginal mean consistency does.

Algorithm 2 Online-Marginal-Quantile-Predictor(q, η)

Let $p_1 = 0$
for $t = 1$ to T **do**
 Observe x_t (and ignore it!)
 Predict p_t
 Observe y_t .
 Let $p_{t+1} = p_t + \eta(q - \mathbb{1}[y_t \leq p_t])$

Theorem 6 *For any sequence of examples of length T , any target quantile $q \in [0, 1]$ and any update parameter $\eta > 0$, Online-Marginal-Quantile-Predictor(q, η) (Algorithm 2) produces a transcript that satisfies marginal quantile consistency with error α for $\alpha \leq \frac{1+\eta}{\eta T}$*

Proof 10 *Examining the update rule $p_{t+1} = p_t + \eta(q - \mathbb{1}[y_t \leq p_t])$ and solving for $\mathbb{1}[y_t \leq p_t]$, we see:*

$$\mathbb{1}[y_t \leq p_t] = q - \frac{p_{t+1} - p_t}{\eta}$$

So, we can compute:

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^T \mathbb{1}[y_t \leq p_t] &= q - \frac{1}{\eta T} \sum_{t=1}^T (p_{t+1} - p_t) \\ &= q - \frac{p_{T+1} - p_1}{\eta T} \\ &= q - \frac{p_{T+1}}{\eta T} \end{aligned}$$

Next observe that for all t , $|p_t - p_{t+1}| \leq \eta$, and since $y_t, q \in [0, 1]$, if $p_t \geq 1$, it must be that $\mathbb{1}[y_t \leq p_t] = 1$ then $p_{t+1} < p_t$ and similarly if $p_t \leq 0$ then $p_{t+1} > p_t$. Hence we must have for all t that:

$$-\eta \leq p_t \leq 1 + \eta$$

So we have:

$$\left| \frac{1}{T} \sum_{t=1}^T \mathbb{1}[y_t \leq p_t] - q \right| = \left| \frac{p_{T+1}}{\eta T} \right| \leq \frac{1 + \eta}{\eta T}$$

Remark 3.3.1 *In fact, there is an even simpler algorithm that can guarantee*

marginal quantile consistency against an adversary, with error tending to 0 at a rate of $1/T$. For a q fraction of rounds, predict $p_t = 1$, and for a $1 - q$ fraction of rounds predict $p_t = 0$. Because we know that $y_t \in [0, 1]$, we have that on the q fraction of rounds for which $p_t = 1$, $\mathbb{1}[y_t \leq p_t] = 1$ and for the remaining $1 - q$ fraction of rounds, $\mathbb{1}[y_t \leq p_t] = 0$. Hence we can satisfy marginal quantile consistency in an entirely data independent way, which should make us suspicious of marginal guarantees on their own and make us ask for something stronger in addition — either from theory or from empirical experiments.

References and Further Reading

Lemma 3.2.2 (bounding the change in pinball loss as a function of the change in predicted quantile under a Lipschitz condition on the distribution) is adapted from Jung et al. [2022]. Algorithm 2 and its analysis are adapted from Gibbs and Candes [2021], who derive it in the context of conformal prediction (which we will see next in Chapter 4).



4

Conformal Prediction

CONTENTS

4.1	Prediction Sets and Nonconformity Scores	29
4.1.1	Non-Conformity Scores	31
4.2	Marginal Coverage in Expectation	33
4.3	Calibration Set Conditional Bounds	35
4.4	Sequential Conformal Prediction	36
4.4.1	Sequential Marginal Coverage Guarantees	37
4.5	Things to Reflect On	38
	References and Further Reading	39

Most of this monograph will be devoted to going beyond marginal consistency properties — but it is worth noting that even the simple marginal consistency properties that we have studied so far can be useful. In this Chapter we’ll explore their application in a popular and powerful framework for uncertainty quantification called “conformal prediction”.

Of course, our reasons for wanting to strengthen marginal guarantees will apply with equal force to conformal prediction. But as we will see, conformal prediction is fundamentally about quantile estimation, and so the techniques that we develop for offering stronger-than-marginal guarantees for quantile estimators will directly apply to conformal prediction as well. We will revisit this in Chapter 10 when we return to conformal prediction to apply what we have learned in the interim.

4.1 Prediction Sets and Nonconformity Scores

Suppose we have a distribution $\mathcal{D} \in \Delta\mathcal{Z}$ (although we will also consider the sequential prediction setting in which there need not be any distribution). Our goal is to be able to produce *prediction sets* as a function of observed features x that are likely to contain the corresponding label y . More specifically, we want to be able to find a function $\mathcal{T} : \mathcal{X} \rightarrow 2^{\mathcal{Y}}$ mapping unlabelled examples x to *subsets* of labels $\mathcal{T}(x)$ that have the property that the true label is contained

**FIGURE 4.1**

Images x about which we might have uncertainty about their labels y .

within $\mathcal{T}(x)$ with some specified level of confidence $1 - \delta$:

$$\Pr[y \in \mathcal{T}(x)] \approx 1 - \delta$$

We leave unspecified for now what distribution this probability is taken over, because we will consider a spectrum of guarantees of increasing strength, mirroring our treatment of mean and quantile estimation. For example, we can ask for marginal guarantees, group conditional guarantees, calibrated guarantees, or ask for guarantees that hold empirically on adversarially chosen sequences. In this chapter we will restrict attention to marginal guarantees, but in Chapter 10 we will return and consider stronger guarantees. Prediction sets can take different forms: when we are facing a regression problem ($\mathcal{Y} = \mathbb{R}$) it is natural (but not necessary) for a prediction set to take the form of an *interval*: $\mathcal{T}(x) = [a, b]$ for some $a < b \in \mathbb{R}$. On the other hand, for a multiclass classification problem (when \mathcal{Y} is some unordered discrete set), prediction sets correspond to subsets of labels — e.g. we might have $\mathcal{T}(x) = \{\text{Blueberry Muffin, Chihuahua}\}$ for x representing images from Figure 10.1.

Prediction sets are a very attractive way to quantify uncertainty: their size represents a quantitative *degree* of uncertainty. For example, if $\mathcal{T}(x)$ is a singleton, this represents certainty at the specified $1 - \delta$ level in a particular point prediction. But the contents of the set also provides insight into *where* the uncertainty lies. For example in a classification problem, there might be a high degree of uncertainty in the specific label, but a well crafted prediction set might nevertheless tell us that our uncertainty is concentrated in a region that corresponds to the same downstream action. Say, in a computer vision setting, we might be unsure of the breed of dog in front of us—so $\mathcal{T}(x)$ contains half a dozen different labels, corresponding to different breeds—but despite this uncertainty in the specifics, this prediction set gives us a high degree of confidence in what action to take—apply the breaks.

The main difficulty with thinking about producing prediction sets is that they are very high dimensional objects: In a k -label multiclass classification setting, there are 2^k different prediction sets. The main idea in *conformal prediction* is to reduce these high dimensional prediction sets to one-dimensional objects using a *non-conformity score function* $s : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, and to use this score function to define a 1-dimensional parametric family of prediction sets.

4.1.1 Non-Conformity Scores

A “non-conformity score function” $s(x, y)$ is typically built from some *model* h for making point predictions. As a running example, let’s imagine that we are in the regression setting ($\mathcal{Y} = \mathbb{R}$) and we have solved a linear regression problem to produce a model $h : \mathcal{X} \rightarrow \mathcal{Y}$ that makes point predictions. Intuitively, the “non-conformity score” $s(x, y)$ is supposed to communicate some measure by which the label y differs from the prediction of the model $h(x)$. The simplest (often too simple) non-conformity score in this setting is:

$$s(x, y) = |h(x) - y|$$

which simply measures the deviation of the label y from the point prediction $h(x)$.

Any non-conformity score function s can be used to parameterize a (now one dimensional) family of prediction sets $\mathcal{T}_s : \mathcal{X} \times \mathbb{R} \rightarrow 2^{\mathcal{Y}}$ as follows:

$$\mathcal{T}_s(x, \tau) = \{\hat{y} : s(x, \hat{y}) \leq \tau\}$$

The prediction set $\mathcal{T}(x, \tau)$ simply contains all labels \hat{y} that would produce nonconformity score at most τ when paired with x : $s(x, \hat{y}) \leq \tau$. In the case of our simple regression running example, this would simply correspond to the interval centered at the point prediction $h(x)$ that has width 2τ : $\mathcal{T}_s(x, \tau) = [h(x) - \tau, h(x) + \tau]$. Although simple, a clear disadvantage of this non-conformity score is that for fixed τ , every prediction interval $\mathcal{T}(x, \tau)$ has the same width — so for methods that use a fixed value of τ — which roughly speaking are those methods that promise only marginal coverage — the prediction intervals do not give us any insight into *which examples* we have more uncertainty about compared to others.

There are many other non-conformity scores that are in wide use. For example, rather than training a regression model h that aims to predict the mean of the conditional label distribution $\mathcal{D}_{\mathcal{Y}}(x)$ (as linear regression does), we could train quantile regression models $h_{\delta/2}(x)$, $h_{1-\delta/2}(x)$ that try and predict the $\delta/2$ and $1 - \delta/2$ quantiles of the conditional label distribution $\mathcal{D}_{\mathcal{Y}}(x)$ instead. Then a natural non-conformity score would be:

$$s(x, y) = \max(h_{\delta/2}(x) - y, y - h_{1-\delta/2}(x))$$

This score starts with the candidate interval that directly arises from the

quantile regression method $[h_{\delta/2}(x), h_{1-\delta/2}(x)]$, and measures how far the label y is from the interval — taking a positive value when y falls outside of the interval and a negative value when it falls inside. If the interval is correct, then the $1 - \delta$ quantile of the nonconformity score distribution will be 0 — picking threshold $\tau = 0$ will get the target marginal coverage. But if the interval induced by the quantile regression method is not correct, then choosing different thresholds τ can systematically widen or shorten the prediction interval by τ on each end: $\mathcal{T}_s(x, \tau) = [h_{\delta/2}(x) - \tau, h_{1-\delta/2}(x) + \tau]$. This non-conformity score has the advantage that even for a fixed value of τ , the prediction intervals $\mathcal{T}_s(x, \tau)$ can have very different widths, depending on the predictions of the models $h_{\delta/2}(x)$ and $h_{1-\delta/2}(x)$. More generally, if we suspect that the underlying data generating process might have some parametric structure (for example, perhaps we think it is an ordinary least squares model), we can estimate its parameters to the best of our ability and let our non-conformity score be $s(x, y) = \max(h_{\delta/2}(x) - y, y - h_{1-\delta/2}(x))$, where now $h_{\delta/2}(x)$ and $h_{1-\delta/2}(x)$ are quantiles of the conditional label distribution, conditional on x , given our parameter estimate — or in a Bayesian setting, under our posterior distribution on parameter estimates. If our modelling assumptions turn out to be correct, then a $1 - \delta$ quantile of the non-conformity score distribution will be 0 — but we will have a way to proceed even if our modelling assumptions fail.

What about for multi-class classification problems, in which $\mathcal{D}_Y(x)$ is a discrete distribution over k possible labels? To build intuition, suppose we were given the true conditional distribution over labels given x : For each label $\hat{y} \in [k]$, $p_x^*(\hat{y}) = \Pr[y = \hat{y}|x]$. Let $\pi_{p_x^*}$ be the permutation on labels that puts them in decreasing sorted order by their underlying probability: so $p_x^*(\pi_{p_x^*}(1)) \geq p_x^*(\pi_{p_x^*}(2)) \geq \dots \geq p_x^*(\pi_{p_x^*}(k))$. How would we find the smallest prediction set that contains the true label with probability at least $1 - \delta$? We would greedily start adding labels to our prediction set in order of their probabilities (highest probability to lowest) until the cumulative probability of the labels in our prediction set exceeded $1 - \delta$. To say this more formally, for each $t \in [k]$, let $C(t, p_x^*) = \sum_{i=1}^{t-1} p_x^*(\pi_{p_x^*}(i))$ denote the cumulative probability of the top t labels in likelihood sorted order. We would choose the prediction set:

$$\mathcal{T}(x) = \{\hat{y} : C(\pi_{p_x^*}^{-1}(\hat{y}), p_x^*) < 1 - \delta\}$$

Now suppose we have a method that gives us a score function $p_x : \mathcal{Y} \rightarrow [0, 1]$ for each example x . We might *hope* that p_x is the true probability distribution over labels, but we have no strong reason to believe that it is. For example, p_x might be the softmax outputs of the final layer of a neural network. We can nevertheless define the same quantities with respect to p_x : π_{p_x} is the permutation that places the labels in descending order according to p_x : $p_x(\pi_{p_x}(1)) \geq \dots \geq p_x(\pi_{p_x}(k))$, and $C(t, p_x) = \sum_{i=1}^{t-1} p_x(\pi_{p_x}(i))$ denotes the cumulative “probability” of the top t labels according to p_x . We can then

define a non-conformity score:

$$s(x, y) = C(\pi_{p_x}^{-1}(y), p_x)$$

In the event that p_x really is the true conditional label distribution conditional on x , then using this non-conformity score, the prediction sets $\mathcal{T}_s(x, \tau) = \{\hat{y} : C(\pi_{p_x}^{-1}(\hat{y}), p_x) \leq \tau\}$ are the minimum size prediction sets with coverage probability τ — and even if they are not, there exists some τ that leads to coverage with the target coverage probability.

There are plenty of other non-conformity scores that one could consider. But for the rest of this chapter, we won't worry about what the non-conformity score is — the techniques we discuss will offer “coverage” guarantees for *any* choice of nonconformity score. This is both a strength and a weakness; the generality of these methods means we are free to explore non-conformity scores to our heart's content. But of course, the cost is that beyond coverage, we will be able to offer very little in terms of promising that the generated prediction sets are *useful* — much of the art of conformal prediction comes down to designing an appropriate score function.

4.2 Marginal Coverage in Expectation

In this section we will consider the problem of using a sample of data $D \sim \mathcal{D}^n$ (that we will call a *calibration set*) to produce prediction sets $\mathcal{T}(x)$ that obtain the following coverage guarantee on *new samples* $(x, y) \sim \mathcal{D}$ that are not contained in D .

$$1 - \delta \leq \Pr_{D \sim \mathcal{D}^n, (x, y) \sim \mathcal{D}} [y \in \mathcal{T}(x)] \leq 1 - \delta + \frac{1}{n+1}$$

This is a marginal coverage guarantee because the probability is over x as well as y , and is unconditioned. We call it a marginal guarantee in *expectation* because the probability is also taken over the calibration set D , and so could be expressed as:

$$1 - \delta \leq \mathbb{E}_{D \sim \mathcal{D}^n} \left[\Pr_{(x, y) \sim \mathcal{D}} [y \in \mathcal{T}(x)] \right] \leq 1 - \delta + \frac{1}{n+1}$$

This is in contrast to theorems we will see later that have high probability guarantees over the randomness of the calibration set D . Nevertheless, this guarantee is very simple to obtain, and has a very mild (inverse linear) dependence on n which makes it attractive.

Algorithm 3 SplitConformal(D, s, δ)

Let τ be the smallest value such that:

$$\sum_{i=1}^n \mathbb{1}[s(x_i, y_i) \leq \tau] \geq (1 - \delta)(n + 1)$$

i.e. τ is an empirical $\frac{\lceil (1-\delta)(n+1) \rceil}{n}$ quantile of D .

Output the function:

$$\mathcal{T}_D(x) = \{\hat{y} : s(x, \hat{y}) \leq \tau\}$$

The algorithm is simple, and given in Algorithm 3. Informally, it takes as input a calibration dataset D (of any size), a non-conformity score s (which must be defined independently of the calibration dataset D), and a target miscoverage rate δ . It computes a threshold τ that comes as close as possible to being an empirical $(1 - \delta)$ -quantile of the set of non-conformity scores induced by S on D (up to a bias correction term of roughly $\frac{n+1}{n}$), and then outputs the function $\mathcal{T}(x) \equiv \mathcal{T}_s(x, \tau)$ that uses the fixed threshold τ for every example x . As we have done previously in our discussion of quantile estimation, we will assume that the distribution on which we want to compute quantiles (which in this case is the induced distribution on non-conformity scores) is *continuous*, which simplifies things. Recall that we can always enforce this assumption by adding arbitrarily small amounts of noise from any continuous distribution to the non-conformity scores.

Theorem 7 Fix any distribution $\mathcal{D} \in \Delta Z$, any $0 \leq \delta \leq 1$ and any non-conformity score $s : \mathcal{Z} \rightarrow \mathbb{R}$. Assume the induced distribution on non-conformity scores $s(x, y)$ for $(x, y) \sim \mathcal{D}$ is continuous. Let $D \sim \mathcal{D}^n$ be a dataset of n points sampled i.i.d. from \mathcal{D} . Then for the function $\mathcal{T}_D(x)$ output by SplitConformal(D, s, δ) (Algorithm 3) we have that:

$$1 - \delta \leq \mathbb{E}_{D \sim \mathcal{D}^n} \left[\Pr_{(x, y) \sim \mathcal{D}} [y \in \mathcal{T}_D(x)] \right] \leq 1 - \delta + \frac{1}{n + 1}$$

In fact, the only property we will use about the distribution from which D and (x, y) are jointly drawn is that it is exchangeable, which means permutation invariant — we will not need the stronger property that the points are drawn i.i.d.

Proof 11 (Proof of Theorem 7) Because we have assumed that the non-conformity score distribution on $s(x, y)$ is continuous, with probability 1, there are no ties amongst the non-conformity scores in D — i.e. for all $i \neq j$, $s(x_i, y_i) \neq s(x_j, y_j)$. Renumber the datapoints in D in increasing order of their nonconformity scores — i.e. such that $s(x_1, y_1) < s(x_2, y_2) < \dots < s(x_n, y_n)$. Let i^* be the unique index such that $s(x_{i^*}, y_{i^*}) = \tau$. $i^* = \lceil (1 - \delta)(n + 1) \rceil$.

Imagine the dataset $D' = D \cup (x, y)$ containing $n + 1$ elements. Consider the event $y \in \mathcal{T}_D(x)$. This occurs exactly when $s(x, y) < \tau$, which is exactly the event that the pair (x, y) occurs before the pair (x_{i^*}, y_{i^*}) when we sort the $n + 1$ points in D' by their non-conformity scores. But since all of the points in D' are exchangeable, by symmetry it must be that point (x, y) will have rank that is uniformly random in $\{1, 2, \dots, n + 1\}$ when put in sorted order within D' . Thus the event that $y \in \mathcal{T}_D(x)$ is the event that (x, y) has rank that is less than i^* , which is:

$$\begin{aligned} \Pr_{D, (x, y)} [y \in \mathcal{T}_D(x)] &= \frac{i^*}{n + 1} \\ &= \frac{[(1 - \delta)(n + 1)]}{n + 1} \\ &\geq \frac{(1 - \delta)(n + 1)}{n + 1} \\ &= (1 - \delta) \end{aligned}$$

We can similarly calculate:

$$\begin{aligned} \Pr_{D, (x, y)} [y \in \mathcal{T}_D(x)] &= \frac{i^*}{n + 1} \\ &= \frac{[(1 - \delta)(n + 1)]}{n + 1} \\ &\leq \frac{(1 - \delta)(n + 1) + 1}{n + 1} \\ &= (1 - \delta) + \frac{1}{n + 1} \end{aligned}$$

which completes the proof.

4.3 Calibration Set Conditional Bounds

In general, we will have a single calibration set that we use to produce prediction sets for many test examples. As a result, bounds like the one we just proved that give us a guarantee only in expectation over the calibration set do not on their own give a very strong guarantee. What if, e.g. the variance of the coverage is enormous? Since we only sample the calibration set once, we don't get to "experience" the expectation, but only the realized coverage guarantee conditional on the calibration set. So one way in which we might strengthen Theorem 7 is to give a bound that holds with high probability over the draw of $D \sim \mathcal{D}^n$ rather than only in expectation. To do this, all we need to do is find a high probability estimate for the $1 - \delta$ quantile of the non-conformity score distribution, which is a problem that we already solved in Chapter 3!

Algorithm 4 HighProbabilitySplitConformal(D, s, δ, γ)

Let τ be the smallest value such that:

$$\frac{1}{n} \sum_{i=1}^n \mathbb{1}[s(x_i, y_i) \leq \tau] \geq (1 - \delta) + \sqrt{\frac{\log(2/\gamma)}{2n}}$$

Output the function:

$$\mathcal{T}_D(x) = \{\hat{y} : s(x, \hat{y}) \leq \tau\}$$

Theorem 8 Fix any distribution $\mathcal{D} \in \Delta Z$, any $0 \leq \delta \leq 1$ and any non-conformity score $s : Z \rightarrow \mathbb{R}$. Assume the induced distribution on non-conformity scores $s(x, y)$ for $(x, y) \sim \mathcal{D}$ is continuous. Let $D \sim \mathcal{D}^n$ be a dataset of n points sampled i.i.d. from \mathcal{D} . Then for the function $\mathcal{T}_D(x)$ output by SplitConformal(D, s, δ, γ) (Algorithm 3) we have that with probability $1 - \gamma$ over the draw of $D \sim \mathcal{D}^n$:

$$1 - \delta \leq \Pr_{(x, y) \sim \mathcal{D}} [y \in \mathcal{T}_D(x)] \leq 1 - \delta + 2\sqrt{\frac{\log(2/\gamma)}{2n}} + \frac{1}{n}$$

Proof 12 By construction, τ is an empirical q -quantile for the empirical distribution of scores $s(x, y)$ over D for:

$$(1 - \delta) + \sqrt{\frac{\log(2/\gamma)}{2n}} \leq q \leq (1 - \delta) + \sqrt{\frac{\log(2/\gamma)}{2n}} + \frac{1}{n}$$

From Theorem 2, we have that with probability $1 - \gamma$, τ is therefore a q' quantile for the distribution of scores $s(x, y)$ over \mathcal{D} for:

$$q - \sqrt{\frac{\log(2/\gamma)}{2n}} \leq q' \leq q + \sqrt{\frac{\log(2/\gamma)}{2n}}$$

Combining these two bounds, we have that with probability $1 - \gamma$, τ is a q' quantile for \mathcal{D} such that:

$$(1 - \delta) \leq q' \leq (1 - \delta) + 2\sqrt{\frac{\log(2/\gamma)}{2n}} + \frac{1}{n}$$

Since $y \in \mathcal{T}_D(x)$ exactly when $s(x, y) \leq \tau$, we have that $\Pr[y \in \mathcal{T}_D(x)] = q'$, which completes the proof.

4.4 Sequential Conformal Prediction

So far we have considered the problem of conformal prediction in the batch setting, in which we have a dataset of labelled examples D that we can use to

train a model that defines a prediction set function $\mathcal{T} : \mathcal{X} \rightarrow 2^{\mathcal{Y}}$ that we can later deploy to produce prediction sets $\mathcal{T}(x)$. A major advantage of this kind of approach is that we do not need to observe labels at test time, but a major disadvantage is that we need to make strong assumptions about the test time distribution — generally that it is identical to the training distribution, and that it is distributed independently — or is at least exchangeable.

In this section we apply the techniques we have developed to the sequential conformal prediction problem, which can be described as the following interaction between a learner and an adversary. In rounds $t \in \{1, \dots, T\}$

1. The adversary chooses a feature vector $x_t \in \mathcal{X}$ and a distribution over labels $y_t \in \mathcal{Y}$.
2. The learner produces a prediction set $\mathcal{T}_{\pi^{<t}}(x_t)$.
3. The learner observes the realized label y_t .

This interaction generates a transcript $\pi = \{(x_1, \mathcal{T}_{\pi^{<1}}(x_1), y_1), \dots, (x_T, \mathcal{T}_{\pi^{<T}}(x_T), y_T)\}$. The learner is an algorithm mapping transcript prefixes $\pi^{<t}$ and feature vectors x_t to prediction sets $\mathcal{T}_{\pi^{<t}}(x_t)$, and the adversary is a mapping from transcript prefixes $\pi^{<t}$ to pairs of feature vectors and label distributions $\mathcal{X} \times \Delta\mathcal{Y}$. The adversary may be arbitrary, or we may impose restrictions on the label distributions that she chooses.

The prediction sets we study will continue to be based on non-conformity score functions s — but since we no longer require exchangeability, we will also allow the non-conformity score function s_t to potentially change at every round. So, for example, if our non-conformity score function is based on a model f , we can use a model f_t that is retrained on all of the examples seen so far, at each round — something that breaks the exchangeability of the non-conformity scores of past and future data by introducing a dependence between the past data and the non-conformity score.

4.4.1 Sequential Marginal Coverage Guarantees

We can derive algorithms for adversarial sequential conformal prediction from our algorithms for online sequential quantile prediction. For example, we can use Algorithm 2 (which promises *marginal* quantile consistency against an adversary) to obtain a sequential conformal prediction algorithm with a corresponding marginal coverage guarantee against an adversary. To talk about coverage rates in the sequential setting, we write $\Pr_{(x_t, \mathcal{T}_t(x_t), y_t) \sim \pi}[\cdot]$ to denote the uniformly random selection of a record $(x_t, \mathcal{T}_t(x_t), y_t)$ from a transcript of T records $\pi = \{(x_t, \mathcal{T}_t(x_t), y_t)\}_{t=1}^T$.

Algorithm 5 Adversarial-Marginal-Conformal-Predictor(δ, η, T)

Let $q = 1 - \delta + \frac{1+\eta}{\eta T}$ and $p_1 = 0$ **for** $t = 1$ to T **do** Obtain non-conformity score s_t and observe x_t .

Predict

$$\mathcal{T}_t(x_t) = \{\hat{y} : s_t(x_t, \hat{y}) \leq p_t\}$$

 Observe y_t . Let $p_{t+1} = p_t + \eta(q - \mathbb{1}[s_t(x_t, y_t) \leq p_t])$

Theorem 9 Fix any $\delta, \eta > 0$. Paired with any adversary, Adversarial-Marginal-Conformal-Predictor(δ, η) (Algorithm 5) produces a transcript such that:

$$(1 - \delta) \leq \Pr_{(x_t, \mathcal{T}_t(x_t), y_t) \sim \pi} [y_t \in \mathcal{T}_t(x_t)] \leq 1 - \delta + 2\frac{1 + \eta}{\eta T}$$

Proof 13 This is an application of Algorithm 2. Thus we can apply Theorem 6 to conclude that the sequence of thresholds p_t produced satisfy α -approximate marginal quantile consistency with respect to target quantile q and the sequence of non-conformity scores $s(x_t, y_t)$ for $\alpha \leq \frac{1+\eta}{\eta T}$. This means that:

$$q - \frac{1 + \eta}{\eta T} \leq \frac{1}{T} \sum_{t=1}^T \mathbb{1}[s_t(x_t, y_t) \leq p_t] \leq q + \frac{1 + \eta}{\eta T}$$

Plugging in our definition of q and noting that $y_t \in \mathcal{T}_t(x_t)$ exactly when $s_t(x_t, y_t) \leq p_t$ completes the proof.

4.5 Things to Reflect On

The most attractive features of conformal prediction are its generality, its simplicity, and its lack of needed assumptions. We needed to make no assumptions at all about the distribution (in the batch setting), or the relationship between the nonconformity score or the model and the data. In the sequential setting we didn't even need to assume the existence of a distribution, and still we were able to get worst-case "coverage" guarantees.

These are indeed attractive properties, and we won't want to abandon them. But we should also continue to reflect on the goals of uncertainty quantification. When is "coverage" the thing we want? What do we or should we actually do with prediction sets? And of course, what are the right ways to strengthen the kinds of marginal coverage guarantees we were able to derive

in this chapter? In the sequential setting — in which we change the threshold at each round — we should be especially skeptical of marginal coverage guarantees, since, as we have seen, it is possible to obtain them in entirely data agnostic ways (for example by producing the full/trivial prediction set a $1 - \delta$ fraction of the time and the empty prediction set a δ fraction of the time). Although this is not necessarily what the algorithms we have seen will (always) do, examples like this do tell us that marginal coverage as a theorem promises us very little about “usefulness” in any kind of downstream application. As we move forward throughout this monograph, we’ll want to keep our eyes on what we can *do* with various kinds of uncertainty quantification, and let these downstream use cases guide our evaluations.

References and Further Reading

See Shafer and Vovk [2008] for a classical introduction to conformal prediction and Angelopoulos and Bates [2021] for an excellent recent survey. The batch conformal prediction algorithms we present here are variants of “split conformal prediction” that use a held out calibration set — this general idea dates back to [Papadopoulos et al., 2002] and was studied in detail by Lei et al. [2018]. Romano et al. [2019] introduced quantile regression as the basis of a non-conformity score into the conformal prediction literature, and Angelopoulos et al. [2020] introduced a non-conformity score based on the softmax output of a neural network for classification problems and demonstrated its utility on ImageNet. The kind of “high probability” dataset conditional marginal guarantees we present were studied by Vovk [2012] and Park et al. [2019]. Gibbs and Candès [2021] studied sequential conformal prediction with marginal coverage guarantees, and derived the algorithm we present here.



5

Calibration

CONTENTS

5.1	Introduction to Calibration	41
5.2	Calibrating a Model f	43
5.3	Interlude: What is Calibration Good For?	47
5.3.1	A Simple Model for Decision Making	47
5.4	Quantile Calibration	51
5.5	Sequential Prediction	53
5.5.1	Sequential (Mean) Calibration	54
5.5.2	Sequential Quantile Calibration	60
	References and Further Reading	65

The marginal guarantees we saw in Chapter 3 were easy to obtain, but extremely weak. In this chapter we'll see one way to go beyond marginal guarantees, by making *calibrated* predictions. Calibration on its own is also quite weak, but not as weak as a marginal guarantee, and should be thought of as one step up in terms of a “sanity check” intended to falsify whether we have learned the true conditional label distribution. As we will see it also has foundational connections to decision making, which will be one of its primary attractions. In a formal sense, we will show that *trusting* that calibrated predictions are correct and *acting accordingly* will be the optimal policy for any downstream decision maker amongst all policies that map predictions to actions.

5.1 Introduction to Calibration

In this section we'll focus on one dimensional regression problems in which the label domain is real valued: $\mathcal{Y} \subset [0, 1]$. A natural special case is when we are predicting binary outcomes: $\mathcal{Y} = \{0, 1\}$, but everything we say holds also for the general real valued case. We will also see later on that everything generalizes to higher dimensional regression problems as well, although we will need to think carefully about our goals in higher dimensions to avoid the complexity of the problem growing exponentially with the dimension.

In a setting like this, we often want to find a model $f : \mathcal{X} \rightarrow [0, 1]$ that has the property that for all $x \in \mathcal{X}$, $f^*(x) = \mathbb{E}_{y \sim \mathcal{D}_{y(x)}}[y]$ is the *conditional label expectation given x* . Of course, we don't generally expect to actually find this function (for the same fundamental reason we saw in Chapter 3), but that's going to be our goal.

Suppose we try and solve this problem and come up with some model f . How can we evaluate whether f is any good? If our goal is purely prediction, we might evaluate f via its *squared error* — i.e. the expected (squared) deviation of its prediction from the true label (Recall definition 1). This is the objective we would minimize if we were solving (e.g.) a least squares regression problem over some class of models \mathcal{H} :

Definition 9 (Squared Error Regression) *The solution to a squared error regression function over model class \mathcal{H} and distribution \mathcal{D} is:*

$$f_{\mathcal{H}}^* = \arg \min_{f \in \mathcal{H}} \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x) - y)^2]$$

On the other hand, if we want our predictions $f(x)$ to have the same probabilistic semantics as $f^*(x)$ — namely that they be a prediction about the *expected value of y* , then we might want that $f(x)$ be calibrated. Calibration asks that the predictions of f be correct conditional on its own predictions: Roughly that $\mathbb{E}_{(x,y) \sim \mathcal{D}}[y | f(x) = v] = v$ for all v . So that the conditioning event makes sense, we will restrict attention so functions f that have a range of finite cardinality, and study *average* calibration error. Let $R(f) = \{f(x) : x \in \mathcal{X}\}$ denote the range of f , and let $m = |R(f)|$ denote the cardinality of f 's range. We will assume $m < \infty$.

Definition 10 (Average Calibration Error) *The average calibration error of a predictor f on a distribution \mathcal{D} is:*

$$K_1(f, \mathcal{D}) = \sum_{v \in R(f)} \Pr_{(x,y) \sim \mathcal{D}} [f(x) = v] \left| v - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f(x) = v] \right|$$

The average squared calibration error is:

$$K_2(f, \mathcal{D}) = \sum_{v \in R(f)} \Pr_{(x,y) \sim \mathcal{D}} [f(x) = v] \left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f(x) = v] \right)^2$$

Finally, we can define a notion of maximum calibration error. Just as with our average notions, we weight by the probability mass of the levelsets to avoid needing to measure quantities over sets with tiny mass:

$$K_{\infty}(f, \mathcal{D}) = \max_{v \in R(f)} \Pr_{(x,y) \sim \mathcal{D}} [f(x) = v] \left| v - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f(x) = v] \right|$$

When the distribution \mathcal{D} is clear from context we will sometimes elide it and simply write $K_1(f)$, $K_2(f)$, $K_{\infty}(f)$, etc.

Sometimes it will be more convenient to work with one of these quantities over another, but they are closely related to one another:

Lemma 5.1.1 For any predictor $f : \mathcal{X} \rightarrow [0, 1]$,

$$\begin{aligned} K_2(f) &\leq K_1(f) \leq \sqrt{K_2(f)} \\ K_\infty(f) &\leq K_1(f) \leq mK_\infty(f) \end{aligned}$$

Proof 14 $K_2(f) \leq K_1(f)$ follows from the fact that since v and y are bounded in $[0, 1]$, term by term $(v - \mathbb{E}_{(x,y) \sim \mathcal{D}}[y|f(x) = v])^2 \leq |v - \mathbb{E}_{(x,y) \sim \mathcal{D}}[y|f(x) = v]| \cdot \sqrt{K_2(f)}$ follows from the Cauchy-Schwarz inequality:

$$\mathbb{E}_v \left[1 \cdot \left| v - \mathbb{E}_{(x,y) \sim \mathcal{D}}[y|f(x) = v] \right| \right]^2 \leq \mathbb{E}_v[1^2] \cdot \mathbb{E} \left[\left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}}[y|f(x) = v] \right)^2 \right]$$

$K_\infty(f) \leq K_1(f)$ follows from the fact that a sum of non-negative terms upper bounds a maximum over the terms, and $K_1(f) \leq mK_\infty(f)$ follows from the fact that $K_1(f)$ is a sum of m terms each of which is upper bounded by $K_\infty(f)$.

Unlike squared error, which we may never be able to drive to zero (because of inherent unpredictability), we can in principle drive calibration error to zero: observe that $K_2(f^*) = 0$.

Recall from Lemma 3.1.2 that the *constant* predictor that minimizes squared error over a label distribution is the mean of that label distribution. From this we can establish (it is a good exercise to think about why this is) that if $f^* \in \mathcal{H}$, then $f^* = f_{\mathcal{H}}^*$ is the solution to the squared error regression problem over \mathcal{H} , and that f^* will have 0 calibration error. In general however, we should not expect that $f^* \in \mathcal{H}$ for any class that we are able to optimize over, and so squared error regression will not necessarily result in a calibrated predictor on its own — although we will later see conditions under which it will.

We'll restate a version of Lemma 3.1.2, which will be useful for us shortly.

Lemma 5.1.2 Fix any distribution on labels \mathcal{D}_y . Let $v^* = \mathbb{E}_{\mathcal{D}_y}[y]$ denote the true label expectation, and let \hat{v} be any other value. Then:

$$\mathbb{E}_{y \sim \mathcal{D}_y} [(\hat{v} - y)^2 - (v^* - y)^2] = (v^* - \hat{v})^2$$

It is precisely because the equality in Lemma 5.1.2 is exact that we will find it convenient to work with the “Euclidean” measure of calibration error, K_2 .

5.2 Calibrating a Model f

Suppose we are given a model f with large average calibration error $K_2(f)$. Can we fix it? And will fixing it come at the cost of accuracy (say, as measured by squared error $B(f)$)? The answers are “Yes”, and “No” respectively! :-) There is a simple algorithm that takes as input an arbitrary model f and outputs a modified model \hat{f} such that:

1. \hat{f} has as low average calibration error as we like: For any α , we can produce \hat{f} such that $K_2(\hat{f}) \leq \alpha$
2. \hat{f} has strictly lower squared error than f if f was not already calibrated.
3. The range of \hat{f} is only smaller than the range of f : $|R(\hat{f})| \leq |R(f)|$

The basic idea will be to take some intermediate model f_t and then “patch” it if it is not already calibrated, to produce a better model f_{t+1} . We will focus on the simplest possible “patch”, and form our calibrated model by simply stringing them together.

Definition 11 (Value Patch) *Given a model f and a pair of values $v, v' \in [0, 1]$, we say that the value patch applied to f with pair (v, v') is the function:*

$$h(x, f; v \rightarrow v') = \begin{cases} v' & f(x) = v \\ f(x) & \text{otherwise} \end{cases}$$

Algorithm 6 Calibrate(f, α, \mathcal{D})

Let $f_0 = f$ and $t = 0$.

while $K_2(f_t, \mathcal{D}) \geq \alpha$ **do**

Let:

$$v_t \in \arg \max_{v \in R(f_t)} \Pr_{(x,y) \sim \mathcal{D}} [f_t(x) = v] \left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v] \right)^2$$

$$v'_t = \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v_t]$$

Let $f_{t+1} = h(x; f_t, v_t \rightarrow v'_t)$ and $t = t + 1$.

Output f_t .

We can now analyze the algorithm.

Theorem 10 *After T rounds, where $T \leq \frac{m}{\alpha}$, Algorithm 6 outputs a model f_T such that $K_2(f_T) \leq \alpha$ and $B(f_T) \leq B(f)$.*

Proof 15 Observe that at each round before the algorithm halts, since $K_2(f_t) \geq \alpha$ we must have that:

$$\Delta_t \equiv \Pr_{(x,y) \sim \mathcal{D}}[f_t(x) = v_t] \left(v_t - \mathbb{E}_{(x,y) \sim \mathcal{D}}[y | f_t(x) = v_t] \right)^2 \geq \frac{\alpha}{m}$$

Rearranging, we also have that:

$$(v_t - v'_t)^2 = \frac{\Delta_t}{\Pr_{(x,y) \sim \mathcal{D}}[f_t(x) = v_t]}$$

Let $\mathcal{D}(v_t) = \mathcal{D}|(f_t(x) = v_t)$ be the distribution that results from conditioning on the event that $f_t(x) = v_t$ and let $\mathcal{D}(\bar{v}_t) = \mathcal{D}|(f_t(x) \neq v_t)$ be the distribution that results from conditioning on the event that $f_t(x) \neq v_t$. We have from Lemma 5.1.2 that:

$$\begin{aligned} & B(f_t, \mathcal{D}) - B(f_{t+1}, \mathcal{D}) \\ &= \Pr[f_t(x) = v_t](B(f_t, \mathcal{D}(v_t)) - B(f_{t+1}, \mathcal{D}(v_t))) + \Pr[f_t(x) \neq v_t](B(f_t, \mathcal{D}(\bar{v}_t)) - B(f_{t+1}, \mathcal{D}(\bar{v}_t))) \\ &= \Pr[f_t(x) = v_t](B(f_t, \mathcal{D}(v_t)) - B(f_{t+1}, \mathcal{D}(v_t))) \\ &= \Pr[f_t(x) = v_t](v_t - v'_t)^2 \\ &= \Delta_t \\ &\geq \frac{\alpha}{m} \end{aligned}$$

Here the second to last equality follows from Lemma 5.1.2. Since for any model $f : \mathcal{X} \rightarrow [0, 1]$, $B(f, \mathcal{D}) \leq 1$ and for any model f_T $B(f_T, \mathcal{D}) \geq 0$, the algorithm must halt after at most $T \leq \frac{m}{\alpha}$ many rounds. Since each iteration decreases squared error, it must be that $B(f_T, \mathcal{D}) \leq B(f, \mathcal{D})$.

In fact, this argument is wasteful, although its form will be useful for us later when we investigate stronger forms of calibration. For simple calibration, there is a simple one-shot algorithm that obtains perfect calibration and decreases squared error by exactly the amount of the calibration error of the original model. This is a variant of what is sometimes called ‘‘Histogram Binning’’:

Algorithm 7 Histogram-Binning(f, \mathcal{D})

For each $v \in R(f)$ let $c(v) = \mathbb{E}_{(x,y) \sim \mathcal{D}}[y | f(x) = v]$
 Output the model \hat{f} defined as $\hat{f}(x) = c(f(x))$.

Theorem 11 For any function f , Histogram-Binning(f, \mathcal{D}) (Algorithm 7) outputs a model \hat{f} such that $K_2(\hat{f}, \mathcal{D}) = 0$ and $B(\hat{f}, \mathcal{D}) = B(f, \mathcal{D}) - K_2(f, \mathcal{D})$.

Proof 16 Consider any level set of \hat{f} : $S(v) = \{x : \hat{f}(x) = v\}$. By definition, for all $x \in S(v)$, we must have $f(x) = v'$ such that $c(v') = v$ — i.e. such that

$c(f(x)) = \mathbb{E}_{(x,y) \sim \mathcal{D}}[y|f(x) = v'] = v$. Let $P(v) = \{v' : c(v') = v\}$. We have that

$$\mathbb{E}_{(x,y)} [y|x \in S(v)] = \frac{\sum_{v' \in P(v)} \Pr[f(x) = v'] c(v')}{\sum_{v' \in P(v)} \Pr[f(x) = v']} = v$$

Hence:

$$K_2(\hat{f}, \mathcal{D}) = \sum_{v \in R(\hat{f})} \Pr[\hat{f}(x) = v] (v - \mathbb{E}[y|x \in S(v)])^2 = 0$$

Next, observe that we can decompose the squared error of both f and \hat{f} according to the level sets of f , which form a partition for \mathcal{X} :

$$\begin{aligned} B(f, \mathcal{D}) - B(\hat{f}, \mathcal{D}) &= \mathbb{E}[(f(x) - y)^2] - \mathbb{E}[(\hat{f}(x) - y)^2] \\ &= \sum_{v \in R(f)} \Pr[f(x) = v] \mathbb{E}[(f(x) - y)^2 - (c(f(x)) - y)^2 | f(x) = v] \\ &= \sum_{v \in R(f)} \Pr[f(x) = v] \mathbb{E}[(v - y)^2 - (c(v) - y)^2 | f(x) = v] \\ &= \sum_{v \in R(f)} \Pr[f(x) = v] (v - c(v))^2 \end{aligned}$$

where the last equality follows from Lemma 5.1.2. But:

$$\sum_{v \in R(f)} \Pr[f(x) = v] (v - c(v))^2 = \sum_{v \in R(f)} \Pr[f(x) = v] (v - \mathbb{E}[y|f(x) = v])^2 = K_2(f, \mathcal{D})$$

which completes the proof.

Thus we see that mis-calibrated models can always be improved: they can be efficiently updated to have *no* calibration error, and in performing this simple update, their squared error is improved by an amount equal to their initial calibration error. This also shows that squared error can be decomposed into two terms: calibration error, and the remainder (which is sometimes called *refinement* error), and that the part corresponding to calibration error can always be removed.

Corollary 5.2.1 *For any model f and distribution \mathcal{D} , we can decompose its squared error into two terms: its calibration error, and its so-called “refinement error”:*

$$B(f, \mathcal{D}) = K_2(f, \mathcal{D}) + \underbrace{\mathbb{E}_{(x,y) \sim \mathcal{D}} [(\mathbb{E}[y|f(x)] - y)^2]}_{\text{refinement error}}$$

Proof 17 *From Theorem 11 we have for any model f :*

$$\begin{aligned} B(f, \mathcal{D}) &= K_2(f, \mathcal{D}) + B(\hat{f}, \mathcal{D}) \\ &= K_2(f) + \mathbb{E}_{(x,y) \sim \mathcal{D}} [(\mathbb{E}[y|f(x)] - y)^2] \end{aligned}$$

In words, the “refinement score” gives us a measure of how much information a model f has about the underlying label distribution — how much of its variance the model could explain, if it were properly calibrated. What Theorem 11 tells us is that we can always obtain squared error as low as the refinement score of the model simply by calibrating it. In other words, “calibration error” is error that can be removed simply by relabeling the outputs of the model. The refinement error cannot be improved without having some way to add information to the model by “refining” it — i.e. causing it to make non-trivial predictions in regions in which it was previously constant valued.

We will eventually be interested in calibrating predictors using a finite sample of data from a distribution (rather than giving our algorithm the ability to directly and exactly compute expectations on the distribution), which will require proving *generalization* theorems. But we will defer this to Chapter 6, when we will prove such theorems for more demanding notions of calibration.

5.3 Interlude: What is Calibration Good For?

So far we have introduced calibration as:

1. A set of basic *consistency checks* that would be passed by “real probabilities” that we can also algorithmically enforce, and
2. Something that we can obtain starting from any predictor while only making that predictor more accurate (at least as measured by squared error).

But why *these* consistency checks? Why did we choose to ask that our predictor be unbiased conditional on its own predictions, instead of any of a variety of other conditional bias guarantees we could have asked for, for example? In this section we’ll take a brief pause to see perhaps the most important reason: calibration provides a powerful interface between prediction and downstream *decision making*. We’ll make the connection here in the simple 1-dimensional calibration setting we’ve been studying, but later we’ll see that the connection continues to hold in higher dimensional settings which makes it a powerful way to think about prediction for decision making.

5.3.1 A Simple Model for Decision Making

In this section we’ll imagine that the label space $\mathcal{Y} = \{0, 1\}$ is binary valued (we’ll generalize this later). As usual, we will consider a model $f : \mathcal{X} \rightarrow [0, 1]$ that aims to predict the conditional label mean $f^*(x) = \mathbb{E}_{y \sim \mathcal{D}_y}(x)$, but may of course fall short.

We will add to the model a collection of *downstream decision makers*, who,

given x , need to choose some *action* from some action space \mathcal{A} . Their goal in choosing an action is to maximize their expected *utility*, which depends not only on the action $a \in \mathcal{A}$ that they choose, but on the realized label $y \in \mathcal{Y}$. Formally, a utility function is a mapping:

$$u : \mathcal{A} \times \mathcal{Y} \rightarrow [0, 1]$$

Different downstream decision makers are parameterized by different utility functions.

Given x , how should a decision maker play so as to maximize their expected utility? They should choose the action that maximizes their expected utility given that $y = 1$ with probability $f^*(x)$. More generally, if they believe that $f^*(x) = p$, they should choose their action according to their *best response* function:

Definition 12 *Given a utility function $u : \mathcal{A} \times \mathcal{Y} \rightarrow [0, 1]$, the corresponding best response function $BR_u : [0, 1] \rightarrow \mathcal{A}$ is:*

$$BR_u(p) = \arg \max_{a \in \mathcal{A}} (p \cdot u(a, 1) + (1 - p)u(a, 0))$$

i.e. the function that chooses the expectation maximizing action under the assumption that $y = 1$ with probability p .

Now suppose we have a decision maker whose only knowledge of the distribution of y conditional on x is through our predictive model $f : \mathcal{X} \rightarrow [0, 1]$. How should they choose their actions? They can choose a policy $P : [0, 1] \rightarrow \mathcal{A}$ mapping predictions to actions. The best response function is a policy, but there are others as well. Their expected utility adopting a policy P is: $\mathbb{E}_{(x,y) \sim \mathcal{D}} [u(P(f(x)), y)]$. It isn't hard to see that if our model $f = f^*$, then simultaneously for all utility functions, playing the best response policy is uniformly better than any other policy:

Theorem 12 *Fix any distribution \mathcal{D} and any utility function $u : \mathcal{A} \times \mathcal{Y} \rightarrow [0, 1]$. Let $f^*(x) = \mathbb{E}_{y \sim \mathcal{D}_y(x)} [y]$. Then for any policy $P : [0, 1] \rightarrow \mathcal{A}$:*

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [u(BR_u(f^*(x)), y)] \geq \mathbb{E}_{(x,y) \sim \mathcal{D}} [u(P(f^*(x)), y)]$$

Proof 18

$$\begin{aligned} \mathbb{E}_{(x,y) \sim \mathcal{D}} [u(BR_u(f^*(x)), y)] &= \mathbb{E}_{x \sim \mathcal{D}_x} [\mathbb{E}_{y \sim \mathcal{D}_y(x)} [u(BR_u(f^*(x)), y)]] \\ &= \mathbb{E}_{x \sim \mathcal{D}_x} [f^*(x)u(BR_u(f^*(x)), 1) + (1 - f^*(x))u(BR_u(f^*(x)), 0)] \\ &\geq \mathbb{E}_{x \sim \mathcal{D}_x} [f^*(x)u(P(f^*(x)), 1) + (1 - f^*(x))u(P(f^*(x)), 0)] \\ &= \mathbb{E}_{(x,y) \sim \mathcal{D}} [u(P(f^*(x)), y)] \end{aligned}$$

The inequality follows from the definition of the best response function.

What we have shown is not surprising — if we knew the *true* conditional label expectation given x , $f^*(x)$, then best responding to it would be an optimal policy. But if we have some other predictor f , is it a good idea to best respond to it? If we have two predictors f_1 and f_2 and f_1 has lower squared error than f_2 , is it at least a *better* idea to best respond to f_1 than it is to f_2 ? Not necessarily.

Example 1 Let $\mathcal{X} = \{a, b\}$ and $\mathcal{Y} = \{0, 1\}$. Let $\mathcal{D} \in \Delta(\mathcal{X} \times \mathcal{Y})$ be the distribution such that $\mathcal{D}_{\mathcal{X}}$ is uniform over $\{a, b\}$ and $f^*(a) = 0.2$ and $f^*(b) = 0.8$.

We define two predictors:

1. f_2 is a constant predictor: $f_2(a) = f_2(b) = 1/2$. Observe that f_2 is calibrated but has high squared error: $B(f_2, \mathcal{D}) = 0.25$.
2. f_1 is better fit to the distribution: $f_1(a) = 0$ and $f_1(b) = 1$. f_1 is not calibrated, but has better squared error: $B(f_1, \mathcal{D}) = 0.2$.

Now consider a decision maker with action set $\mathcal{A} = \{L, R\}$ and the following utility function:

$$\begin{aligned} u(L, 1) &= u(L, 0) = 0.8 \\ u(R, 1) &= 0.9 \quad u(R, 0) = 0 \end{aligned}$$

Essentially, L is a safe action that will give the decision maker utility 0.8 for any outcome, whereas R is a riskier gamble: it will give payoff 0.9 if the outcome is $y = 1$, but will give no payoff otherwise. The distribution \mathcal{D} is such that we never have enough certainty about the outcome to justify choosing R , as the most certain we ever are about an outcome of $y = 1$ is when we see $x = b$, but $f^*(b) = 0.8$, and at 80% odds, we still prefer the safe option as $0.8 \cdot 0.9 = 0.72 \leq 0.8$. Thus the optimal policy for this decision maker plays L always. This happens to be exactly the policy that corresponds to best responding to the constant predictor f_2 : $BR_u(0.5) = L$ — and so best responding to f_2 obtains utility 0.8 for the decision maker, and there is no way to do better on this distribution. What about best responding to f_1 ? This causes the decision maker to play the safe option L half of the time (when $x = a$) and to play the risky option half of the time (when $x = b$) — since the forecasts produced by f_1 make it appear as if the high payoff option is a sure thing. Thus best responding to f_1 gets utility $\frac{1}{2} \cdot 0.8 + \frac{1}{2} \cdot 0.72 = 0.76 < 0.8$.

From this example we learn two things. First, squared error is not necessarily aligned with the utility of a downstream decision maker: in this case best responding to the model with lower squared error obtains lower utility. But we also learn that for an uncalibrated model, best response might not be the optimal policy — in the example above, the constant policy P_L defined as $P_L(p) = L$ for all p would be a superior policy if the decision maker only had access to f_1 . In general, for uncalibrated models, a decision maker might have to themselves *learn* what the best policy was. So if all we've done is we've

learned a low error predictive model, there might be another learning problem that we need to solve when it comes time to actually use the predictions to take action. It would be better if we had predictors that we could *trust* in the sense that we could use them straightforwardly by trusting their predictions and behaving accordingly (by best responding to them). Calibrated predictors turn out to have exactly this property: just like we proved for the perfect predictor f^* , if an arbitrary model f is calibrated, then simultaneously for all utility functions that a downstream decision maker might have, best responding to f is the optimal policy.

Theorem 13 *Fix any distribution \mathcal{D} , and let $f : \mathcal{X} \rightarrow [0, 1]$ be a model with bounded calibration error: $K_1(f, \mathcal{D}) \leq \alpha$. Then for any utility function $u : \mathcal{A} \times \{0, 1\} \rightarrow [0, 1]$ and for any policy $P : [0, 1] \rightarrow \mathcal{A}$ we have that:*

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [u(BR_u(f(x)), y)] \geq \mathbb{E}_{(x,y) \sim \mathcal{D}} [u(P(f(x)), y)] - 2\alpha$$

Proof 19 *For each $v \in R(f)$, let $\alpha_v = |v - \mathbb{E}_{\mathcal{D}}[y|f(x) = v]|$ be the component of the calibration error associated with prediction v . By definition of $K_1(f, \mathcal{D})$ and our assumption, we have that:*

$$\sum_{v \in R(f)} \Pr_{\mathcal{D}}[f(x) = v] \cdot \alpha_v \leq \alpha.$$

We can now calculate:

$$\begin{aligned} & \mathbb{E}_{(x,y) \sim \mathcal{D}} [u(BR_u(f(x)), y)] \\ = & \sum_{v \in R(f)} \Pr[f(x) = v] \cdot \mathbb{E}[u(BR_u(v), y) | f(x) = v] \\ = & \sum_{v \in R(f)} \Pr[f(x) = v] \cdot \left(\frac{\mathbb{E}[y|f(x) = v]}{\mathcal{D}} u(BR_u(v), 1) + (1 - \frac{\mathbb{E}[y|f(x) = v]}{\mathcal{D}}) u(BR_u(v), 0) \right) \\ \geq & \sum_{v \in R(f)} \Pr[f(x) = v] \cdot (v \cdot u(BR_u(v), 1) + (1 - v) \cdot u(BR_u(v), 0) - \alpha_v) \\ \geq & \sum_{v \in R(f)} \Pr[f(x) = v] \cdot (v \cdot u(BR_u(v), 1) + (1 - v) \cdot u(BR_u(v), 0)) - \alpha \\ \geq & \sum_{v \in R(f)} \Pr[f(x) = v] \cdot (v \cdot u(P(v), 1) + (1 - v) \cdot u(P(v), 0)) - \alpha \\ \geq & \sum_{v \in R(f)} \Pr[f(x) = v] \cdot \left(\frac{\mathbb{E}[y|f(x) = v]}{\mathcal{D}} \cdot u(P(v), 1) + (1 - \frac{\mathbb{E}[y|f(x) = v]}{\mathcal{D}}) \cdot u(P(v), 0) \right) - 2\alpha \\ = & \mathbb{E}_{(x,y) \sim \mathcal{D}} [u(P(f(x)), y)] - 2\alpha \end{aligned}$$

Here the first inequality follows from the definition of α_v , the second follows from our bound on calibration error, the third follows from the definition of BR_u , and the fourth uses our bound on calibration error to transition back from the predicted probabilities v to the true conditional expectations $\mathbb{E}[y|f(x) = v]$.

So we can think of calibration as providing a simple interface between prediction and decision making: finding the optimal policy mapping predictions of a calibrated predictor to decisions does not require any further learning step: best responding to calibrated predictions is always the optimal policy.

Lets pause a moment to reflect, because the guarantee we have just proven is tempting to interpret as something stronger than it is. Recall that a constant predictor (that always predicts the marginal label mean) is calibrated, and yet clearly is not terribly useful for decision making. How can we square these two facts? The key is that what we have shown is that best responding to a calibrated predictor is the optimal *policy*, and the way we have defined *policies* is as functions that map a prediction to an action. So indeed, if we have a constant predictor (that is calibrated), best responding is the best policy, but because the predictor is not very informative, all of the policies we can implement from its predictions are bad. Calibration does not promise that the decision making might not be able to do better by using other, external information not encoded in a model's forecasts. If we want best responding to our forecasts to be superior to a broader set of policies that also have external information available to them, we will need a stronger set of guarantees than simply (marginal) calibration.

5.4 Quantile Calibration

We can similarly define *quantile calibration* for a target quantile q , which asks that a model f produce quantiles $f(x)$ that satisfy marginal quantile consistency not just overall, but conditional on the value of $f(x)$.

Definition 13 (Average Quantile Calibration Error) *The average quantile calibration error with respect to a target quantile q of a predictor f is:*

$$Q_1(f) = \sum_{v \in R(f)} \Pr_{(x,y) \sim \mathcal{D}} [f(x) = v] \left| q - \Pr_{(x,y) \sim \mathcal{D}} [y \leq v | f(x) = v] \right|$$

The average squared quantile calibration error is:

$$Q_2(f) = \sum_{v \in R(f)} \Pr_{(x,y) \sim \mathcal{D}} [f(x) = v] \left(q - \Pr_{(x,y) \sim \mathcal{D}} [y \leq v | f(x) = v] \right)^2$$

Finally, we can define a notion of maximum quantile calibration error. Just as with our average notions, we weight by the probability mass of the levelsets to avoid needing to measure quantities over sets with tiny mass:

$$Q_\infty(f) = \max_{v \in R(f)} \Pr_{(x,y) \sim \mathcal{D}} [f(x) = v] \left| q - \Pr_{(x,y) \sim \mathcal{D}} [y \leq v | f(x) = v] \right|$$

The relationship between these different measures of quantile calibration error is the same as it is for the corresponding measures of (mean) calibration error: we restate their relationship here without proof, which is identical to the case of mean calibration.

Lemma 5.4.1 *For any predictor $f : \mathcal{X} \rightarrow [0, 1]$,*

$$\begin{aligned} Q_2(f) &\leq Q_1(f) \leq \sqrt{Q_2(f)} \\ Q_\infty(f) &\leq Q_1(f) \leq mQ_\infty(f) \end{aligned}$$

We now give an analogue to our one-shot mean calibrator. There is also an analogous iterative version — that we will build on when we study multigroup guarantees in Chapter 6 — but as with mean calibration, it has no advantages in this setting.

Algorithm 8 One-Shot-Quantile-Calibrate(f, q, \mathcal{D})

For each $v \in R(f)$ let

$$c(v) = \arg \min_{v'} |q - \Pr[y \leq v' | f(x) = v]|$$

Output the model \hat{f} defined as $\hat{f}(x) = c(f(x))$.

Theorem 14 *For any function f , any target quantile value $q \in [0, 1]$, and any ρ -Lipschitz distribution \mathcal{D} , One-Shot-Quantile-Calibrate(f, \mathcal{D}) (Algorithm 8) outputs a model \hat{f} such that $Q_2(\hat{f}) = 0$ and $PB_q(\hat{f}) \leq PB_q(f) - \frac{1}{2\rho}Q_2(f)$.*

Proof 20 *Consider any level set of \hat{f} : $S(v) = \{x : \hat{f}(x) = v\}$. By definition, for all $x \in S(v)$, we must have $f(x) = v'$ such that $c(v') = v$ — i.e. such that $c(f(x))$ satisfies $\Pr_{(x,y) \sim \mathcal{D}}[y \leq c(f(x)) | f(x) = v'] = q$. Let $P(v) = \{v' : c(v') = v\}$. We have that*

$$\Pr_{(x,y)} [y \leq v | x \in S(v)] = \frac{\sum_{v' \in P(v)} \Pr[f(x) = v'] \Pr[y \leq v | f(x) = v']}{\sum_{v' \in P(v)} \Pr[f(x) = v']} = q$$

Hence:

$$Q_2(\hat{f}) = \sum_{v \in R(\hat{f})} \Pr[\hat{f}(x) = v] (q - \Pr[y \leq v | x \in S(v)])^2 = 0$$

Next, observe that we can decompose the pinball loss of both f and \hat{f} ac-

cording to the level sets of f , which form a partition for \mathcal{X} :

$$\begin{aligned}
PB_q(f, \mathcal{D}) - PB_q(\hat{f}, \mathcal{D}) &= \mathbb{E}[L_q(f(x), y)] - \mathbb{E}[L_q(\hat{f}(x), y)] \\
&= \sum_{v \in R(f)} \Pr[f(x) = v] \mathbb{E}[L_q(f(x), y) - L_q(\hat{f}(x), y) | f(x) = v] \\
&\geq \sum_{v \in R(f)} \Pr[f(x) = v] \frac{(\Pr[y \leq f(x) | f(x) = v] - q)^2}{2\rho} \\
&= \frac{1}{2\rho} Q_2(f)
\end{aligned}$$

where the second to last inequality follows from Lemma 3.2.2.

5.5 Sequential Prediction

We now return to the sequential prediction setting, this time to solve a more challenging problem than simply obtaining marginal mean or quantile consistency: our goal will be to obtain empirically calibrated predictions p_t in the worst case over sequences of observations and outcomes. (x_t, y_t) . We will assume that our prediction algorithm makes predictions in the discrete grid $p_t \in [1/m] = \{0, 1/m, 2/m, \dots, 1\}$. We begin by defining empirical analogues of our calibration scores K and Q :

Definition 14 (Average Mean and Quantile Calibration Error) Fix any transcript $\pi = \{(p_1, x_1, y_1), \dots, (p_T, x_T, y_T)\}$ of length T . For each $p \in [1/m]$ let $n(\pi, p) = \sum_{t=1}^T \mathbb{1}[p_t = p]$ be the number of times that the prediction $p_t = p$ is made over the T rounds of the transcript.

The average squared (mean) calibration error on this transcript is:

$$K_2(\pi) = \sum_{p \in [1/m]} \frac{n(\pi, p)}{T} \left(\frac{\sum_{t=1}^T \mathbb{1}[p_t = p](y_t - p_t)}{n(\pi, p)} \right)^2 = \frac{1}{T} \sum_{p \in [1/m]} \left(\frac{\sum_{t=1}^T \mathbb{1}[p_t = p](y_t - p_t)}{\sqrt{n(\pi, p)}} \right)^2$$

It will be convenient for us to be able to refer to the un-normalized calibration error:

$$\hat{K}_2(\pi) = \sum_{p \in [1/m]} \left(\frac{\sum_{t=1}^T \mathbb{1}[p_t = p](y_t - p_t)}{\sqrt{n(\pi, p)}} \right)^2$$

Observe that $K_2(\pi) = \frac{1}{T} \hat{K}_2(\pi)$.

For a target quantile $q \in [0, 1]$, the average squared quantile calibration error on this transcript is:

$$Q_2(\pi) = \sum_{p \in [1/m]} \frac{n(\pi, p)}{T} \left(\sum_{t=1}^T \frac{\mathbb{1}[p_t = p](q - \mathbb{1}[y_t \leq p_t])}{n(\pi, p)} \right)^2 = \frac{1}{T} \sum_{p \in [1/m]} \left(\sum_{t=1}^T \frac{\mathbb{1}[p_t = p](q - \mathbb{1}[y_t \leq p_t])}{\sqrt{n(\pi, p)}} \right)^2$$

Similarly, we define the unnormalized quantile calibration error:

$$\hat{Q}_2(\pi) = \sum_{p \in [1/m]} \left(\sum_{t=1}^T \frac{\mathbb{1}[p_t = p](q - \mathbb{1}[y_t \leq p_t])}{\sqrt{n(\pi, p)}} \right)^2$$

Here any term in the sum in which $n(\pi, p) = 0$ evaluates to 0 by convention.

Note that these measures are identical to their distributional analogues, where the “expectation” is taken over the uniform distribution over the T entries of the transcript. We can similarly define the ℓ_1 and ℓ_∞ variants of calibration error for the sequential setting, and the relationships amongst them will remain unchanged.

We will derive algorithms that (based on their performance on a transcript of length $t - 1$ so far, and possibly on the next context x_t) decide on their prediction p_t at round t . After they make their prediction, they learn the true label y_t , and the transcript extends by one round. We write $\pi^{<t} = \{(p_1, x_1, y_1), \dots, (p_{t-1}, x_{t-1}, y_{t-1})\}$ to denote a transcript corresponding to rounds $1, \dots, t - 1$, and given a record of the prediction, context, and outcome at round t (p_t, x_t, y_t) write the transcript that is extended by one round as $\pi^{\leq t} = \pi^{<t+1} = \pi^{<t} \circ (p_t, x_t, y_t)$. Similarly given a transcript π of length T we will write $\pi^{\leq t}$ to denote the prefix of this transcript of length t .

5.5.1 Sequential (Mean) Calibration

In deriving an algorithm for guaranteeing sequential mean calibration, it will be helpful for us to understand how the average squared calibration score increases from round to round, given the prediction of the algorithm p_t and the outcome y_t . It will be useful for us to develop some notation.

Definition 15 *Fixing a transcript π of length T , for any $s \leq T$ and $p \in [1/m]$ define the quantity:*

$$V_s^p(\pi) = \sum_{t=1}^s \frac{\mathbb{1}[p_t = p](y_t - p_t)}{\sqrt{n(\pi^{\leq s}, p)}}$$

If $n(\pi, p) = 0$ then by convention we define $V_s^p(\pi) = 0$.

We observe that for all p, s, π :

$$|V_s^p(\pi)| \leq \sqrt{n(\pi^{\leq s}, p)}$$

Our goal is to understand how the calibration error increases from round to round as a function of the transcript — and once we understand it, give an algorithm that guarantees that the increase is small. The next lemma bounds the increase in calibration error between rounds s and $s + 1$ as a function of the transcript up through round $s + 1$.

Lemma 5.5.1 Fix any partial transcript $\pi^{\leq s}$ and any triple $(p_{s+1}, x_{s+1}, y_{s+1})$ of potential outcomes for the next round. Let $\pi^{\leq s+1} = \pi^{\leq s} \circ (p_{s+1}, x_{s+1}, y_{s+1})$ be the corresponding continuation of the transcript. Define:

$$\Delta_{s+1}(p_{s+1}, y_{s+1}) = \hat{K}_2(\pi^{\leq s+1}) - \hat{K}_2(\pi^{\leq s})$$

to be the increase in the (unnormalized) squared calibration error that results from the transcript continuation. Then we have that:

$$\Delta_{s+1}(p_{s+1}, y_{s+1}) \leq \left(\frac{2V_s^{p_{s+1}}}{\sqrt{n(\pi^{\leq s}, p_{s+1})}} \cdot (y_{s+1} - p_{s+1}) + \frac{1}{n(\pi^{\leq s}, p_{s+1})} \right)$$

Proof 21 Since the terms in the squared mean calibration error corresponding to predictions $p \neq p_{s+1}$ do not change, We can compute:

$$\begin{aligned} \Delta_{s+1}(p_{s+1}, y_{s+1}) &= \hat{K}_2(\pi^{\leq s+1}) - \hat{K}_2(\pi^{\leq s}) \\ &= \left(\left(\frac{\sum_{t=1}^{s+1} \mathbb{1}[p_t = p_{s+1}](y_t - p_t)}{\sqrt{n(\pi^{\leq s+1}, p_{s+1})}} \right)^2 - \left(\frac{\sum_{t=1}^s \mathbb{1}[p_t = p_{s+1}](y_t - p_t)}{\sqrt{n(\pi^{\leq s}, p_{s+1})}} \right)^2 \right) \\ &\leq \left(\left(\frac{\sum_{t=1}^{s+1} \mathbb{1}[p_t = p_{s+1}](y_t - p_t)}{\sqrt{n(\pi^{\leq s}, p_{s+1})}} \right)^2 - \left(\frac{\sum_{t=1}^s \mathbb{1}[p_t = p_{s+1}](y_t - p_t)}{\sqrt{n(\pi^{\leq s}, p_{s+1})}} \right)^2 \right) \\ &= \left(\left(V_s^{p_{s+1}}(\pi^{\leq s}) + \frac{y_{s+1} - p_{s+1}}{\sqrt{n(\pi^{\leq s}, p_{s+1})}} \right)^2 - V_s^{p_{s+1}}(\pi^{\leq s})^2 \right) \\ &= \left(2V_s^{p_{s+1}}(\pi^{\leq s}) \left(\frac{y_{s+1} - p_{s+1}}{\sqrt{n(\pi^{\leq s}, p_{s+1})}} \right) + \frac{(y_{s+1} - p_{s+1})^2}{n(\pi^{\leq s}, p_{s+1})} \right) \\ &\leq \left(\frac{2V_s^{p_{s+1}}}{\sqrt{n(\pi^{\leq s}, p_{s+1})}} \cdot (y_{s+1} - p_{s+1}) + \frac{1}{n(\pi^{\leq s}, p_{s+1})} \right) \end{aligned}$$

Next, our plan is to show that for *every* transcript $\pi^{\leq s}$ there is a distribution over subsequent predictions p_{s+1} such that for *every* possible realization of y_{s+1} , $\mathbb{E}_{p_{s+1}}[\Delta_{s+1}(p_{s+1}, y_{s+1})]$ is small. If we can show this, then the algorithm that consists of playing this randomized strategy at each round will have small expected calibration loss, which we can conclude simply by summing the terms $\Delta_s(p_s, y_s)$ from $s = 1$ to T .

Towards this end, define:

$$\Delta_{s+1}^1(p_{s+1}, y_{s+1}) = \frac{2V_s^{p_{s+1}}}{\sqrt{n(\pi^{\leq s}, p_{s+1})}} \cdot (y_{s+1} - p_{s+1})$$

With this notation, Lemma 5.5.1 states that:

$$\Delta_{s+1}(p_{s+1}, y_{s+1}) \leq \Delta_{s+1}^1(p_{s+1}, y_{s+1}) + \frac{1}{n(\pi^{\leq s}, p_{s+1})}.$$

Here the term $\frac{1}{n(\pi^{\leq s}, p_{s+1})}$ evaluates to 0 if $n(\pi^{\leq s}, p_{s+1}) = 0$.

We next establish a randomized prediction strategy that makes the first term $\mathbb{E}_{p_{s+1}}[\Delta_{s+1}^1(p_{s+1}, y_{s+1})]$ small in expectation.

Lemma 5.5.2 *Fix any partial transcript $\pi^{\leq s}$. Consider the distribution over p_{s+1} that we can sample from as follows:*

1. If $V_s^1(\pi^{\leq s}) \geq 0$: Predict $p_{s+1} = 1$ with probability 1
2. If $V_s^0(\pi^{\leq s}) \leq 0$: Predict $p_{s+1} = 0$ with probability 1.
3. Otherwise: Find a $p \in \{0, \frac{1}{m}, \dots, \frac{m-1}{m}\}$ such that $V_s^p(\pi^{\leq s}) \geq 0$ and $V_s^{p+1/m}(\pi^{\leq s}) \leq 0$. Compute $q \in [0, 1]$ such that:

$$q \cdot \frac{V_s^p(\pi^{\leq s})}{\sqrt{n(\pi^{\leq s}, p)}} + (1 - q) \cdot \frac{V_s^{p+1/m}(\pi^{\leq s})}{\sqrt{n(\pi^{\leq s}, p + \frac{1}{m})}} = 0$$

Predict $p_{s+1} = p$ with probability q and predict $p_{s+1} = p + \frac{1}{m}$ with probability $1 - q$.

This distribution has the property that for every $y_{s+1} \in [0, 1]$:

$$\mathbb{E}_{p_{s+1}}[\Delta_{s+1}^1(p_{s+1}, y_{s+1})] \leq \frac{2}{m}$$

Proof 22 We bound $\mathbb{E}_{p_{s+1}}[\Delta_{s+1}^1(p_{s+1}, y_{s+1})]$ separately in each of the three cases.

Case 1:

In this case, $V_s^1(\pi^{\leq s}) \geq 0$ and $p_{s+1} = 1$. Note that since $y_{s+1} \in [0, 1]$, we must have that $(y_{s+1} - p_{s+1}) \leq 0$ and so for all $y_{s+1} \in [0, 1]$:

$$\Delta_{s+1}^1(p_{s+1}, y_{s+1}) = \frac{2V_s^{p_{s+1}}}{\sqrt{n(\pi^{\leq s}, p_{s+1})}} \cdot (y_{s+1} - p_{s+1}) \leq 0$$

Case 2:

In this case, $V_s^0(\pi^{\leq s}) \leq 0$ and $p_{s+1} = 0$. Note that since $y_{s+1} \in [0, 1]$, we must have that $(y_{s+1} - p_{s+1}) \geq 0$ and so for all $y_{s+1} \in [0, 1]$:

$$\Delta_{s+1}^1(p_{s+1}, y_{s+1}) = \frac{2V_s^{p_{s+1}}}{\sqrt{n(\pi^{\leq s}, p_{s+1})}} \cdot (y_{s+1} - p_{s+1}) \leq 0$$

Case 3:

First we observe that in this case, $V_s^0(\pi^{\leq s}) \geq 0$ and $V_s^1(\pi^{\leq s}) \leq 0$. Hence there must exist some adjacent pair $p, p + 1/m \in [1/m]$ such that $V_s^p(\pi^{\leq s}) \geq 0$ and

$V_s^{p+1/m}(\pi^{\leq s}) \leq 0$, so the algorithm is well defined. Recall that $q \in [0, 1]$ is such that $q \cdot \frac{V_s^p(\pi^{\leq s})}{\sqrt{n(\pi^{\leq s}, p)}} + (1 - q) \cdot \frac{V_s^{p+1/m}(\pi^{\leq s})}{\sqrt{n(\pi^{\leq s}, p + \frac{1}{m})}} = 0$. We can compute:

$$\begin{aligned}
& \mathbb{E}_{p_{s+1}} [\Delta_{s+1}^1(p_{s+1}, y_{s+1})] \\
& \leq \left(q \cdot \frac{2V_s^p(\pi^{\leq s})}{\sqrt{n(\pi^{\leq s}, p)}} \cdot (y_{s+1} - p) + (1 - q) \frac{2V_s^{p+1/m}(\pi^{\leq s})}{\sqrt{n(\pi^{\leq s}, p + \frac{1}{m})}} \cdot \left(y_{s+1} - p - \frac{1}{m} \right) \right) \\
& = \left(-\frac{1}{m} \cdot (1 - q) \frac{2V_s^{p+1/m}(\pi^{\leq s})}{\sqrt{n(\pi^{\leq s}, p + \frac{1}{m})}} \right) \\
& \leq \frac{2}{m}
\end{aligned}$$

Here the last inequality follows from the fact that for all $p \in [1/m]$, $|V_s^p(\pi^{\leq s})| \leq \sqrt{n(\pi^{\leq s}, p)}$.

Applying the prediction strategy defined in 5.5.2 repeatedly gives us an algorithm (Algorithm 9) for making sequential predictions that are calibrated against arbitrary sequences of outcomes.

Algorithm 9 Online-Calibrated-Predictor(m)

for $t = 1$ to T **do**

Observe x_t (and ignore it!)

if $V_{t-1}^1(\pi^{<t}) \geq 0$ **then**

Predict $p_t = 1$.

else if $V_{t-1}^0(\pi^{<t}) \leq 0$ **then**

Predict $p_t = 0$.

else

Select $p \in \{0, \frac{1}{m}, \dots, \frac{m-1}{m}\}$ such that $V_{t-1}^p(\pi^{<t}) \geq 0$ and

$V_{t-1}^{p+1/m}(\pi^{<t}) \leq 0$.

Compute $q \in [0, 1]$ such that:

$$q \cdot \frac{V_{t-1}^p(\pi^{<t})}{\sqrt{n(\pi^{<t}, p)}} + (1 - q) \cdot \frac{V_{t-1}^{p+1/m}(\pi^{<t})}{\sqrt{n(\pi^{<t}, p + \frac{1}{m})}} = 0$$

Predict $p_t = p$ with probability q and predict $p_t = p + \frac{1}{m}$ with probability $1 - q$.

Observe y_t

Let $\pi^{<t+1} = \pi^{<t} \circ (x_t, p_t, y_t)$

Theorem 15 *Against any adaptive adversary, Online-Calibrated-Predictor (Algorithm 9) invoked with the range $[1/m]$ induces a distribution over length*

T transcripts π such that:

$$\mathbb{E}_{\pi}[K_2(\pi)] \leq \frac{2}{m} + \frac{m+1}{T} \cdot (\log(T/m) + 1)$$

In particular, if we choose discretization parameter $m = \sqrt{\frac{2T}{\log T}}$ then we have:

$$\mathbb{E}_{\pi}[K_2(\pi)] \leq O\left(\sqrt{\frac{\log T}{T}}\right)$$

Proof 23 Fix any length T transcript $\pi = \{(x_1, p_1, y_1), \dots, (x_T, p_T, y_T)\}$. Since $\hat{K}_2(\pi^{\leq 0}) = 0$ we have that the telescoping sum:

$$\sum_{t=1}^T \Delta_t(p_t, y_t) = \sum_{t=1}^T \hat{K}_2(\pi^{\leq t}) - \hat{K}_2(\pi^{\leq t-1}) = \hat{K}_2(\pi)$$

From Lemma 5.5.1 we can write this as:

$$\begin{aligned} \hat{K}_2(\pi) &= \sum_{t=1}^T \Delta_t(p_t, y_t) \\ &\leq \sum_{t=1}^T \left(\Delta_t^1(p_t, y_t) + \frac{1}{n(\pi^{\leq t-1}, p_t)} \right) \\ &\leq \sum_{t=1}^T \Delta_t^1(p_t, y_t) + \max_{\tilde{\pi}} \sum_{t=1}^T \frac{1}{n(\tilde{\pi}^{\leq t-1}, \tilde{p}_t)} \end{aligned}$$

where in the last step, we take the maximum over all length t transcripts $\tilde{\pi} = \{(\tilde{x}_1, \tilde{p}_1, \tilde{y}_1), \dots, (\tilde{x}_T, \tilde{p}_T, \tilde{y}_T)\}$

We now take the expectation of both sides (over the randomness of the algorithm's predictions p_t) and apply Lemma 5.5.2:

$$\begin{aligned} \mathbb{E}[\hat{K}_2(\pi)] &\leq \sum_{t=1}^T \mathbb{E}_{p_t, y_t} [\Delta_t^1(p_t, y_t) | \pi^{\leq t}] + \max_{\tilde{\pi}} \sum_{t=1}^T \frac{1}{n(\tilde{\pi}^{\leq t-1}, \tilde{p}_t)} \\ &\leq \frac{2T}{m} + \max_{\tilde{\pi}} \sum_{t=1}^T \frac{1}{n(\tilde{\pi}^{\leq t-1}, \tilde{p}_t)} \end{aligned}$$

It remains to bound $\max_{\tilde{\pi}} \sum_{t=1}^T \frac{1}{n(\tilde{\pi}^{\leq t-1}, \tilde{p}_t)}$. To do this, we observe that whenever $\tilde{p}_t = p$, then we must have that $n(\tilde{\pi}^{\leq t}, p) = n(\tilde{\pi}^{\leq t-1}, p) + 1$. Hence for

any transcript $\tilde{\pi}$ we can write:

$$\begin{aligned}
\sum_{t=1}^T \frac{1}{n(\tilde{\pi}^{\leq t-1}, \tilde{p}_t)} &= \sum_{p \in [1/m]} \sum_{t: \tilde{p}_t = p} \frac{1}{n(\tilde{\pi}^{\leq t-1}, p)} \\
&= \sum_{p \in [1/m]} \sum_{k=1}^{n(\tilde{\pi}, p)-1} \frac{1}{k} \\
&\leq (m+1) \sum_{k=1}^{T/m} \frac{1}{k} \\
&= (m+1) \cdot H_{T/m} \\
&\leq (m+1) \cdot (\log(T/m) + 1)
\end{aligned}$$

Here H_k denotes the k 'th Harmonic number.

Combining these bounds we find that:

$$\mathbb{E}[K_2(\pi)] = \mathbb{E} \left[\frac{\hat{K}_2(\pi)}{T} \right] \leq \frac{2}{m} + \frac{m+1}{T} \cdot (\log(T/m) + 1)$$

We derived a bound on the *expected* calibration error of this algorithm, but if we prefer we can easily state and prove a high probability bound on its calibration error as well. To do so will be another application of Azuma's inequality (Theorem 69).

Theorem 16 *Against any adaptive adversary, Online-Calibrated-Predictor (Algorithm 9) invoked with the range $[1/m]$ induces a distribution over length T transcripts π such that with probability $1 - \delta$ over the randomness of the algorithm:*

$$K_2(\pi) \leq \frac{2}{m} + \sqrt{\frac{2 \ln(2/\delta)}{T}} + \frac{m+1}{T} \cdot (\log(T/m) + 1)$$

In particular, if we choose discretization parameter $m = \sqrt{\frac{2T}{\log T}}$ then we have that with probability $1 - \delta$:

$$K_2(\pi) \leq O \left(\sqrt{\frac{\log \frac{T}{\delta}}{T}} \right)$$

Proof 24 *We recall several facts. First the (un-normalized) calibration error can be expressed as the sum of T terms:*

$$\hat{K}_2(\pi) \leq \sum_{t=1}^T \Delta_t^1(p_t, y_t) + (m+1)(\log(T/m) + 1)$$

Next, each of these terms is bounded in absolute value in the worst case over predictions, outcomes, and transcripts: $|\Delta_t^1(p_t, y_t)| \leq 1$ Finally, we have bounded the conditional expectation of each term in this sum in Lemma 5.5.2:

$$\mathbb{E}_{p_t, y_t} [\Delta_t^1(p_t, y_t) | \pi^{<t}] \leq \frac{2}{m}$$

These are all of the ingredients we need to apply Azuma's inequality (Theorem 69) to conclude that for all $b > 0$:

$$\Pr \left[\sum_{t=1}^T \Delta_t^1(p_t, y_t) \geq \frac{2T}{m} + b \right] \leq 2 \exp \left(\frac{-b^2}{2T} \right)$$

Setting the right hand side to δ and solving for b yields:

$$\Pr \left[\sum_{t=1}^T \Delta_t^1(p_t, y_t) \geq \frac{2T}{m} + \sqrt{2T \ln(2/\delta)} \right] \leq \delta$$

Hence we have that with probability $1 - \delta$:

$$K_2(\pi) = \frac{1}{T} \hat{K}_2(\pi) \leq \frac{2}{m} + \sqrt{\frac{2 \ln(2/\delta)}{T}} + \frac{m+1}{T} \cdot (\log(T/m) + 1)$$

5.5.2 Sequential Quantile Calibration

We can derive an algorithm for making sequential predictions that are quantile calibrated in an analogous way. The derivation is almost identical to our derivation of sequential mean calibration: so we will sketch it, pointing out those parts that differ. As in our batch quantile algorithm, we will need to now assume that the adversary picks continuous *distributions* of labels at each round rather than allowing her to choose deterministically, since obtaining quantile calibration is not in general possible against point mass distributions. Moreover our quantitative bound will require assuming that the adversary's label distributions are ρ -Lipschitz and will depend on ρ .

Definition 16 Fix a target quantile $q \in [0, 1]$. Fixing a transcript π of length T , for any $s \leq T$ and $p \in [1/m]$ define the quantity:

$$W_s^p(\pi, q) = \sum_{t=1}^s \frac{\mathbb{1}[p_t = p](q - \mathbb{1}[y_t \leq p_t])}{\sqrt{n(\pi^{\leq s}, p)}}$$

If $n(\pi, p) = 0$ then by convention we define $W_s^p(\pi, q) = 0$. When q is clear from context we elide it and just write $W_s^p(\pi)$

We observe that for all p, s, π :

$$|W_s^p(\pi, q)| \leq \sqrt{n(\pi^{\leq s}, p)}$$

Lemma 5.5.3 Fix any $q \in [0, 1]$ and any partial transcript $\pi^{\leq s}$ and any triple $(p_{s+1}, x_{s+1}, y_{s+1})$ of potential outcomes for the next round. Let $\pi^{\leq s+1} = \pi^{\leq s} \circ (p_{s+1}, x_{s+1}, y_{s+1})$ be the corresponding continuation of the transcript. Redefine:

$$\Delta_{s+1}(p_{s+1}, y_{s+1}) = \hat{Q}_2(\pi^{\leq s+1}) - \hat{Q}_2(\pi^{\leq s})$$

to be the increase in the (unnormalized) squared quantile calibration error that results from the transcript continuation. Then we have that:

$$\Delta_{s+1}(p_{s+1}, y_{s+1}) \leq \left(\frac{2W_s^{p_{s+1}}}{\sqrt{n(\pi^{\leq s}, p_{s+1})}} \cdot (q - \mathbb{1}[y_{s+1} \leq p_{s+1}]) + \frac{1}{n(\pi^{\leq s}, p_{s+1})} \right)$$

The proof is essentially identical to that of Lemma 5.5.1 — we include it here for completeness.

Proof 25 Since the terms in the squared quantile calibration error corresponding to predictions $p \neq p_{s+1}$ do not change, We can compute:

$$\begin{aligned} & \Delta_{s+1}(p_{s+1}, y_{s+1}) \\ &= \hat{Q}_2(\pi^{\leq s+1}) - \hat{Q}_2(\pi^{\leq s}) \\ &= \left(\left(\frac{\sum_{t=1}^{s+1} \mathbb{1}[p_t = p_{s+1}](q - \mathbb{1}[y_t \leq p_t])}{\sqrt{n(\pi^{\leq s+1}, p_{s+1})}} \right)^2 - \left(\frac{\sum_{t=1}^s \mathbb{1}[p_t = p_{s+1}](q - \mathbb{1}[y_t \leq p_t])}{\sqrt{n(\pi^{\leq s}, p_{s+1})}} \right)^2 \right) \\ &\leq \left(\left(\frac{\sum_{t=1}^{s+1} \mathbb{1}[p_t = p_{s+1}](q - \mathbb{1}[y_t \leq p_t])}{\sqrt{n(\pi^{\leq s}, p_{s+1})}} \right)^2 - \left(\frac{\sum_{t=1}^s \mathbb{1}[p_t = p_{s+1}](q - \mathbb{1}[y_t \leq p_t])}{\sqrt{n(\pi^{\leq s}, p_{s+1})}} \right)^2 \right) \\ &= \left(\left(W_s^{p_{s+1}}(\pi^{\leq s}) + \frac{(q - \mathbb{1}[y_{s+1} \leq p_{s+1}])}{\sqrt{n(\pi^{\leq s}, p_{s+1})}} \right)^2 - W_s^{p_{s+1}}(\pi^{\leq s})^2 \right) \\ &= \left(2W_s^{p_{s+1}}(\pi^{\leq s}) \left(\frac{(q - \mathbb{1}[y_{s+1} \leq p_{s+1}])}{\sqrt{n(\pi^{\leq s}, p_{s+1})}} \right) + \frac{(q - \mathbb{1}[y_{s+1} \leq p_{s+1}])^2}{n(\pi^{\leq s}, p_{s+1})} \right) \\ &\leq \left(\frac{2W_s^{p_{s+1}}}{\sqrt{n(\pi^{\leq s}, p_{s+1})}} \cdot (q - \mathbb{1}[y_{s+1} \leq p_{s+1}]) + \frac{1}{n(\pi^{\leq s}, p_{s+1})} \right) \end{aligned}$$

Next, our plan is to show that for *every* transcript $\pi^{\leq s}$ there is a distribution over subsequent predictions p_{s+1} such that for *every* possible ρ -Lipschitz distribution over y_{s+1} , $\mathbb{E}_{p_{s+1}, y_{s+1}}[\Delta_{s+1}(p_{s+1}, y_{s+1})]$ is small. Note that here we are deviating from our derivation of mean calibration algorithms, in that we are requiring that the label y_{s+1} be drawn from a ρ -Lipschitz distribution, and we are taking the expectation over y_{s+1} as well as p_{s+1} . If we can show this, then the algorithm that consists of playing this randomized strategy at each round will have small expected quantile calibration loss, which we can conclude simply by summing the terms $\Delta_s(p_s, y_s)$ from $s = 1$ to T .

Towards this end, define:

$$\Delta_{s+1}^1(p_{s+1}, y_{s+1}) = \frac{2W_s^{p_{s+1}}}{\sqrt{n(\pi^{\leq s}, p_{s+1})}} \cdot (q - \mathbb{1}[y_{s+1} \leq p_{s+1}])$$

With this notation, Lemma 5.5.3 states that:

$$\Delta_{s+1}(p_{s+1}, y_{s+1}) \leq \Delta_{s+1}^1(p_{s+1}, y_{s+1}) + \frac{1}{n(\pi^{\leq s}, p_{s+1})}.$$

Here the term $\frac{1}{n(\pi^{\leq s}, p_{s+1})}$ evaluates to 0 if $n(\pi^{\leq s}, p_{s+1}) = 0$.

We next establish a randomized prediction strategy that makes the first term $\mathbb{E}_{p_{s+1}, y_{s+1}}[\Delta_{s+1}^1(p_{s+1}, y_{s+1})]$ small in expectation for any ρ -Lipchitz distribution over y_{s+1} .

Lemma 5.5.4 *Fix any partial transcript $\pi^{\leq s}$. Consider the distribution over p_{s+1} that we can sample from as follows:*

1. If $W_s^1(\pi^{\leq s}) \geq 0$: Predict $p_{s+1} = 1$ with probability 1
2. If $W_s^0(\pi^{\leq s}) \leq 0$: Predict $p_{s+1} = 0$ with probability 1.
3. Otherwise: Find a $p \in \{0, \frac{1}{m}, \dots, \frac{m-1}{m}\}$ such that $W_s^p(\pi^{\leq s}) \geq 0$ and $W_s^{p+1/m}(\pi^{\leq s}) \leq 0$. Compute $b \in [0, 1]$ such that:

$$b \cdot \frac{W_s^p(\pi^{\leq s})}{\sqrt{n(\pi^{\leq s}, p)}} + (1-b) \cdot \frac{W_s^{p+1/m}(\pi^{\leq s})}{\sqrt{n(\pi^{\leq s}, p + \frac{1}{m})}} = 0$$

Predict $p_{s+1} = p$ with probability b and predict $p_{s+1} = p + \frac{1}{m}$ with probability $1-b$.

This distribution has the property that for every ρ -Lipschitz distribution over $y_{s+1} \in [0, 1]$:

$$\mathbb{E}_{p_{s+1}, y_{s+1}} [\Delta_{s+1}^1(p_{s+1}, y_{s+1})] \leq \frac{2\rho}{m}$$

Proof 26 *We bound $\mathbb{E}_{p_{s+1}, y_{s+1}}[\Delta_{s+1}^1(p_{s+1}, y_{s+1})]$ separately in each of the three cases.*

Case 1:

In this case, $W_s^1(\pi^{\leq s}) \geq 0$ and $p_{s+1} = 1$. Note that since $q, y_{s+1} \in [0, 1]$, we must have that $(q - \mathbb{1}[y_{s+1} \leq p_{s+1}]) \leq 0$ and so for all $y_{s+1} \in [0, 1]$:

$$\Delta_{s+1}^1(p_{s+1}, y_{s+1}) = \frac{2W_s^{p_{s+1}}}{\sqrt{n(\pi^{\leq s}, p_{s+1})}} \cdot (q - \mathbb{1}[y_{s+1} \leq p_{s+1}]) \leq 0$$

Case 2:

In this case, $W_s^0(\pi^{\leq s}) \leq 0$ and $p_{s+1} = 0$. Note that since $q, y_{s+1} \in [0, 1]$, we must have that if $y_{s+1} > 0$ (which occurs with probability 1 if it is drawn from a continuous distribution), $(q - \mathbb{1}[y_{s+1} \leq p_{s+1}]) \geq 0$ and so for all $q, y_{s+1} \in (0, 1]$:

$$\Delta_{s+1}^1(p_{s+1}, y_{s+1}) = \frac{2W_s^{p_{s+1}}}{\sqrt{n(\pi^{\leq s}, p_{s+1})}} \cdot (q - \mathbb{1}[y_{s+1} \leq p_{s+1}]) \leq 0$$

Case 3:

First we observe that in this case, $W_s^0(\pi^{\leq s}) \geq 0$ and $W_s^1(\pi^{\leq s}) \leq 0$. Hence there must exist some adjacent pair $p, p+1/m \in [1/m]$ such that $W_s^p(\pi^{\leq s}) \geq 0$ and $W_s^{p+1/m}(\pi^{\leq s}) \leq 0$, so the algorithm is well defined. Recall that $b \in [0, 1]$ is such that $b \cdot \frac{W_s^p(\pi^{\leq s})}{\sqrt{n(\pi^{\leq s}, p)}} + (1-b) \cdot \frac{W_s^{p+1/m}(\pi^{\leq s})}{\sqrt{n(\pi^{\leq s}, p+1/m)}} = 0$. We can compute:

$$\begin{aligned} & \mathbb{E}_{p_{s+1}, y_{s+1}} [\Delta_{s+1}^1(p_{s+1}, y_{s+1})] \\ & \leq \left(b \cdot \frac{2W_s^p(\pi^{\leq s})}{\sqrt{n(\pi^{\leq s}, p)}} \cdot (q - \Pr[y_{s+1} \leq p]) + (1-b) \frac{2W_s^{p+1/m}(\pi^{\leq s})}{\sqrt{n(\pi^{\leq s}, p+1/m)}} \cdot \left(q - \Pr[y_{s+1} \leq p + \frac{1}{m}] \right) \right) \\ & \leq \left(b \cdot \frac{2W_s^p(\pi^{\leq s})}{\sqrt{n(\pi^{\leq s}, p)}} \cdot (q - \Pr[y_{s+1} \leq p]) + (1-b) \frac{2W_s^{p+1/m}(\pi^{\leq s})}{\sqrt{n(\pi^{\leq s}, p+1/m)}} \cdot \left(q - \Pr[y_{s+1} \leq p] - \frac{\rho}{m} \right) \right) \\ & = \left(-\frac{\rho}{m} \cdot (1-b) \frac{2W_s^{p+1/m}(\pi^{\leq s})}{\sqrt{n(\pi^{\leq s}, p+1/m)}} \right) \\ & \leq \frac{2\rho}{m} \end{aligned}$$

Here the second inequality follows from the fact that the distribution over y_{s+1} is assumed to be ρ -Lipschitz, and hence for all p :

$$\Pr_{y_{s+1}} \left[y_{s+1} \leq p + \frac{1}{m} \right] \leq \Pr_{y_{s+1}} [y_{s+1} \leq p] + \frac{\rho}{m}$$

The last inequality follows from the fact that for all $p \in [1/m]$, $|W_s^p(\pi^{\leq s})| \leq \sqrt{n(\pi^{\leq s}, p)}$.

Applying the prediction strategy defined in 5.5.4 repeatedly gives us an algorithm (Algorithm 10) for making sequential predictions that are calibrated against arbitrary sequences of outcomes.

Algorithm 10 Online-Quantile-Calibrated-Predictor(q, m)

for $t = 1$ to T **do**
 Observe x_t (and ignore it!)
if $W_s^1(\pi^{<t}, q) \geq 0$ **then**
 Predict $p_t = 1$.
else if $W_s^0(\pi^{<t}, q) \leq 0$ **then**
 Predict $p_t = 0$.
else
 Select $p \in \{0, \frac{1}{m}, \dots, \frac{m-1}{m}\}$ such that such that $W_s^p(\pi^{\leq s}, q) \geq 0$ and
 $W_s^{p+1/m}(\pi^{\leq s}, q) \leq 0$.
 Compute $b \in [0, 1]$ such that:

$$b \cdot \frac{W_s^p(\pi^{\leq s}, q)}{\sqrt{n(\pi^{\leq s}, p)}} + (1 - b) \cdot \frac{W_s^{p+1/m}(\pi^{\leq s}, q)}{\sqrt{n(\pi^{\leq s}, p + \frac{1}{m})}} = 0$$

 Predict $p_{s+1} = p$ with probability b and predict $p_{s+1} = p + \frac{1}{m}$ with
 probability $1 - b$.
 Observe y_t
 Let $\pi^{<t+1} = \pi^{<t} \circ (x_t, p_t, y_t)$

Theorem 17 *Against any adaptive adversary that chooses a ρ -Lipschitz distribution over y_t at each round t , Online-Quantile-Calibrated-Predictor (Algorithm 10) invoked with quantile $q \in [0, 1]$ and the range $[1/m]$ induces a distribution over length T transcripts π such that:*

$$\mathbb{E}_\pi[Q_2(\pi)] \leq \frac{2\rho}{m} + \frac{m+1}{T} \cdot (\log(T/m) + 1)$$

In particular, if we choose discretization parameter $m = \sqrt{\frac{2\rho T}{\log T}}$ then we have:

$$\mathbb{E}_\pi[Q_2(\pi)] \leq O\left(\sqrt{\frac{\rho \log T}{T}}\right)$$

Proof 27 Fix any length T transcript $\pi = \{(x_1, p_1, y_1), \dots, (x_T, p_T, y_T)\}$. Since $\hat{Q}_2(\pi^{\leq 0}) = 0$ we have that the telescoping sum:

$$\sum_{t=1}^T \Delta_t(p_t, y_t) = \sum_{t=1}^T \hat{Q}_2(\pi^{\leq t}) - \hat{Q}_2(\pi^{\leq t-1}) = \hat{Q}_2(\pi)$$

From Lemma 5.5.3 we can write this as:

$$\begin{aligned}\hat{Q}_2(\pi) &= \sum_{t=1}^T \Delta_t(p_t, y_t) \\ &\leq \sum_{t=1}^T \left(\Delta_t^1(p_t, y_t) + \frac{1}{n(\pi^{\leq t-1}, p_t)} \right) \\ &\leq \sum_{t=1}^T \Delta_t^1(p_t, y_t) + \max_{\tilde{\pi}} \sum_{t=1}^T \frac{1}{n(\tilde{\pi}^{\leq t-1}, \tilde{p}_t)}\end{aligned}$$

where in the last step, we take the maximum over all length t transcripts $\tilde{p}_i = \{(\tilde{x}_1, \tilde{p}_1, \tilde{y}_1), \dots, (\tilde{x}_T, \tilde{p}_T, \tilde{y}_T)\}$

We now take the expectation of both sides (over the randomness of the algorithm's predictions p_t) and apply Lemma 5.5.4:

$$\begin{aligned}\mathbb{E}[\hat{Q}_2(\pi)] &\leq \sum_{t=1}^T \mathbb{E}_{p_t, y_t} [\Delta_t^1(p_t, y_t) | \pi^{\leq t}] + \max_{\tilde{\pi}} \sum_{t=1}^T \frac{1}{n(\tilde{\pi}^{\leq t-1}, \tilde{p}_t)} \\ &\leq \frac{2\rho T}{m} + \max_{\tilde{\pi}} \sum_{t=1}^T \frac{1}{n(\tilde{\pi}^{\leq t-1}, \tilde{p}_t)}\end{aligned}$$

It remains to bound $\max_{\tilde{\pi}} \sum_{t=1}^T \frac{1}{n(\tilde{\pi}^{\leq t-1}, \tilde{p}_t)}$. To do this, we observe that whenever $\tilde{p}_t = p$, then we must have that $n(\tilde{\pi}^{\leq t}, p) = n(\tilde{\pi}^{\leq t-1}, p) + 1$. Hence for any transcript \tilde{p}_i we can write:

$$\begin{aligned}\sum_{t=1}^T \frac{1}{n(\tilde{\pi}^{\leq t-1}, \tilde{p}_t)} &= \sum_{p \in [1/m]} \sum_{t: \tilde{p}_t = p} \frac{1}{n(\tilde{\pi}^{\leq t-1}, p)} \\ &= \sum_{p \in [1/m]} \sum_{k=1}^{n(\tilde{\pi}, p)-1} \frac{1}{k} \\ &\leq (m+1) \sum_{k=1}^{T/m} \frac{1}{k} \\ &= (m+1) \cdot H_{T/m} \\ &\leq (m+1) \cdot (\log(T/m) + 1)\end{aligned}$$

Here H_k denotes the k 'th Harmonic number.

Combining these bounds we find that:

$$\mathbb{E}[Q_2(\pi)] = \mathbb{E} \left[\frac{\hat{Q}_2(\pi)}{T} \right] \leq \frac{2\rho}{m} + \frac{m+1}{T} \cdot (\log(T/m) + 1)$$

Add high probability bound

References and Further Reading

See Zadrozny and Elkan [2001] for an evaluation of methods like histogram binning for simple calibration. The iterative calibration algorithm (Algorithm 6) follows the design of the *multicalibration* algorithms given in Hébert-Johnson et al. [2018] (which bounds $K_\infty(f)$) and Gopalan et al. [2022b] (which bounds $K_1(f)$) (See Chapter 6 for more on multicalibration).

Calibration in the sequential setting has a long history dating back to Dawid [1982] and Dawid [1985]. The first algorithm that guaranteed worst-case calibration in the sequential setting was given by Foster and Vohra [1998] and alternative derivations were given by Foster [1999], Fudenberg and Levine [1999], Hart [2020], and others. Algorithm 9 is a variant of the algorithm given by Foster and Hart [2021] and its generalization to multicalibration given in Gupta et al. [2022]. Algorithm 10 is a variant of the online quantile multicalibration algorithm given by Bastani et al. [2022].

6

Multigroup Guarantees

CONTENTS

6.1	Group Conditional Mean Consistency	68
6.2	Group Conditional Quantile Consistency	70
6.2.1	A More Direct Approach to Group Conditional Guarantees	72
6.2.1.1	Generalization	74
6.3	Multicalibration: Group Conditional Calibration	83
6.3.1	What is Multicalibration Good For?	87
6.4	Quantile Multicalibration	89
6.5	Out of Sample Generalization	92
6.5.1	Mean Multicalibration	92
6.5.2	Quantile Multicalibration	100
6.6	Loss Minimization with Group Conditional Accuracy Guarantees	101
6.6.1	Reducing Multi-Group Optimal Learning to (Marginal) PAC Learning	103
6.7	Sequential Prediction	106
6.7.1	Achieving Bucketed Calibration	106
6.7.2	Obtaining Bucketed Quantile Multicalibration	113
	References and Further Reading	117

Marginal guarantees are easy to obtain, but very weak. We saw one way of strengthening those guarantees: calibration. But on its own calibration is also quite weak. Obtaining it in the adversarial sequential prediction setting was non-trivial, but we could obtain it in the batch setting with a simple constant predictor $\hat{f}(x) = \mathbb{E}_{(x,y) \sim \mathcal{D}}[y]$ that just predicts the mean of the marginal label distribution. Moreover, all of the techniques we've seen so far *entirely ignore the features x and depend only on the labels y !* We'll now consider a different way to strengthen marginal guarantees, first on its own, and then together with calibration. We will call these *multi-group* guarantees, and they ask for guarantees that hold conditional on the features x in various ways.

Let $\mathcal{G} \in 2^{\mathcal{X}}$ denote a collection of *groups* or *subsets* of the data domain \mathcal{X} . We will represent groups using their indicator functions: so $g \in \mathcal{G}$ is represented as a function $g : \mathcal{X} \rightarrow \{0, 1\}$, where $g(x) = 1$ denotes that $x \in \mathcal{X}$ is a member of group g , and $g(x) = 0$ denotes that x is not a member of \mathcal{G} .

Given an example $x \in \mathcal{X}$, we will write $\mathcal{G}(x) = \{g \in \mathcal{G} : g(x) = 1\}$ to denote the set of groups that x is a member of. At a high level, our aim will be to obtain guarantees like mean consistency (and eventually calibration) not just marginally, but *conditionally* on $g(x) = 1$ for every $g \in \mathcal{G}$ for some large set \mathcal{G} .

6.1 Group Conditional Mean Consistency

With only a finite number of samples from the distribution, we will not in general be able to provide group conditional guarantees conditional on groups that have tiny probability under our distribution, simply because we won't have seen very many points from this part of the probability space. So, the probability mass of a group will be a key parameter for us:

Definition 17 *Under a distribution \mathcal{D} , a group $g : \mathcal{X} \rightarrow \{0, 1\}$ has probability mass $\mu(g)$ defined as:*

$$\mu(g) = \Pr_{x \sim \mathcal{D}_x} [g(x) = 1]$$

Definition 18 *A model $f : \mathcal{X} \rightarrow [0, 1]$ satisfies α -approximate group conditional mean consistency with respect to a set of groups $\mathcal{G} \in 2^{\mathcal{X}}$ if for every $g \in \mathcal{G}$:*

$$\left(\mathbb{E}_{(x,y) \sim \mathcal{D}} [f(x)|g(x) = 1] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y|g(x) = 1] \right)^2 \leq \frac{\alpha}{\mu(g)}$$

Notice that our requirement smoothly becomes less demanding as the measure of the group g grows smaller, allowing us to ask for stronger guarantees for groups for which we will have more data. We have parameterized things so that the scaling is at the right rate: the error within a sub-group g increases at a rate of $1/\sqrt{\mu(g)}$, which is the same rate at which the error of our best estimate of $\mathbb{E}_{(x,y) \sim \mathcal{D}} [y|g(x) = 1]$ from the data will necessarily increase.

We will now show how to update a model f that does not satisfy group conditional mean consistency to one that does, using a sequence of “patches” that are similar to how we obtained calibration. Just as in the examples we have seen thus far, these patches will be accuracy improving, and so we will quickly converge to a group conditional mean consistent model.

Definition 19 (Group Shift Patch) *Given a model f , a shift $\Delta \in \mathbb{R}$, and a group $g : \mathcal{X} \rightarrow \{0, 1\}$ we say that the group patch applied to f with shift Δ and group g is the function:*

$$h(x, f; g, \Delta) = \begin{cases} f(x) + \Delta & g(x) = 1 \\ f(x) & \text{otherwise} \end{cases}$$

Algorithm 11 GroupShift(f, α, \mathcal{G})

Let $f_0 = f$ and $t = 0$.

while f_t does not satisfy α -approximate group conditional mean consistency w.r.t. \mathcal{G} : **do**

Let:

$$g_t \in \arg \max_{g \in \mathcal{G}} \mu(g) \left(\mathbb{E}_{(x,y) \sim \mathcal{D}} [f_t(x)|g(x) = 1] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y|g(x) = 1] \right)^2$$

$$\Delta_t = \mathbb{E}_{(x,y) \sim \mathcal{D}} [y|g_t(x) = 1] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [f_t(x)|g_t(x) = 1]$$

Let $f_{t+1} = h(x, f_t; g_t, \Delta_t)$ and $t = t + 1$.

Output f_t .

Lemma 6.1.1 Fix any model $f_t : \mathcal{X} \rightarrow [0, 1]$ and group $g : \mathcal{X} \rightarrow \{0, 1\}$. Let

$$\Delta_t = \mathbb{E}_{(x,y) \sim \mathcal{D}} [y|g_t(x) = 1] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [f_t(x)|g_t(x) = 1]$$

and

$$f_{t+1} = h(x, f_t; g_t, \Delta_t)$$

(i.e. the update performed at Round t of Algorithm 11). Then:

$$B(f_t) - B(f_{t+1}) = \mu(g_t) \cdot \Delta_t^2$$

Proof 28 By the definition of the patch $h(x, f_t; g_t, \Delta_t)$, models f_t and f_{t+1} differ in their predictions only for x such that $g_t(x) = 1$. Therefore we can calculate:

$$\begin{aligned} B(f_t) - B(f_{t+1}) &= \Pr[g_t(x) = 0] \cdot \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f_t(x) - y)^2 - (f_{t+1}(x) - y)^2 | g_t(x) = 0] \\ &\quad + \Pr[g_t(x) = 1] \cdot \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f_t(x) - y)^2 - (f_{t+1}(x) - y)^2 | g_t(x) = 1] \\ &= \mu(g_t) \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f_t(x) - y)^2 - (f_t(x) + \Delta_t - y)^2 | g_t(x) = 1] \\ &= \mu(g_t) \left(2\Delta_t \mathbb{E}_{(x,y) \sim \mathcal{D}} [y - f_t(x) | g_t(x) = 1] - \Delta_t^2 \right) \\ &= \mu(g_t) (2\Delta_t^2 - \Delta_t^2) \\ &= \mu(g_t) \Delta_t^2 \end{aligned}$$

Theorem 18 Given any model f , any collection of groups \mathcal{G} , and any $\alpha > 0$ Algorithm 11 (GroupShift) halts after $T \leq 1/\alpha$ many rounds and outputs a model f_T that satisfies α -approximate group conditional mean consistency. Moreover, if the algorithm runs for T rounds, then $B(f_T) \leq B(f) - T\alpha$.

Proof 29 At any round T at which the algorithm halts, by the stopping condition of the algorithm it must be that f_T satisfies α -approximate group conditional mean consistency. It remains to bound T and $B(f_T)$.

Consider any intermediate round $t < T$ of the algorithm. We know since the algorithm has not halted that:

$$\max_{g \in \mathcal{G}} \mu(g) \cdot \left(\mathbb{E}_{(x,y) \sim \mathcal{D}} [f(x)|g(x) = 1] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y|g(x) = 1] \right)^2 \geq \alpha$$

g_t realizes this maximum, so we must have:

$$\mu(g_t) \cdot \Delta_t^2 \geq \alpha$$

Thus by Lemma 6.1.1, $B(f_{t+1}) \leq B(f_t) - \alpha$. Inductively applying this claim gives $B(f_T) \leq B(f) - T\alpha$ as desired.

Since (by assumption) $y, f(x) \in [0, 1]$, we have that $B(f_T) \geq 0$ and $B(f) \leq 1$. Thus we must have that $T \leq 1/\alpha$.

Unlike marginal mean consistency, group conditioned mean consistency is clearly a non-trivial promise: if $\mathcal{G} = 2^{\mathcal{X}}$, the set of all subsets, and $\alpha = 0$, then the only model satisfying α -approximate group conditional mean consistency with respect to \mathcal{G} is the model encoding true conditional label distributions f^* . For smaller collections of groups \mathcal{G} and larger values of α we have a necessarily weaker guarantee, but at least we have a parametric family of guarantees that allows us to interpolate between one satisfied by a (trivial) constant function, and one only satisfied by a perfect model.

6.2 Group Conditional Quantile Consistency

Our algorithm and analysis for group conditional mean consistency directly translates to quantiles when we replace the role of the Brier score in our analysis with Pinball loss. First, we can define an analogous notion for group conditional quantile consistency:

Definition 20 A model $f : \mathcal{X} \rightarrow [0, 1]$ satisfies α -approximate group conditional quantile consistency with respect to a target quantile q and set of groups $\mathcal{G} \in 2^{\mathcal{X}}$ if for every $g \in \mathcal{G}$:

$$\left(\Pr_{(x,y) \sim \mathcal{D}} [y \leq f(x)|g(x) = 1] - q \right)^2 \leq \frac{\alpha}{\mu(g)}$$

Our algorithm proceeds by applying the same kind of group-shift patches we used in the case of group conditional mean consistency.

Algorithm 12 QuantileGroupShift($f, \alpha, \mathcal{G}, q$)

Let $f_0 = f$ and $t = 0$.

while f_t does not satisfy α -approximate group conditional quantile consistency w.r.t. target quantile q and \mathcal{G} : **do**

Let:

$$g_t \in \arg \max_{g \in \mathcal{G}} \mu(g_t) \left(\Pr_{(x,y) \sim \mathcal{D}} [y \leq f_t(x) | g(x) = 1] - q \right)^2$$

$$\Delta_t = \arg \min_{\Delta} \left(\Pr_{(x,y) \sim \mathcal{D}} [y \leq f_t(x) + \Delta | g_t(x) = 1] - q \right)^2$$

Let $f_{t+1} = h(x, f_t; g_t, \Delta_t)$ and $t = t + 1$.

Output f_t .

Theorem 19 Assume that \mathcal{D} is a ρ -Lipschitz continuous probability distribution. Given any model f , any collection of groups \mathcal{G} , any target quantile $q \in [0, 1]$ and any $\alpha > 0$ Algorithm 12 (QuantileGroupShift) halts after T rounds where:

$$T \leq \frac{2\rho PB_q(f)}{\alpha} \leq \frac{2\rho}{\alpha}.$$

It outputs a model f_T that satisfies α -approximate group conditional quantile consistency. Moreover, if the algorithm runs for T rounds, then $PB_q(f_T) \leq PB_q(f) - T \cdot \frac{\alpha}{2\rho}$.

Proof 30 If the algorithm halts at round T , then by definition of the halting condition it must be that f_T satisfies α -approximate group conditional quantile consistency with respect to q and \mathcal{G} , so it remains to bound T .

If the algorithm has not halted at round t , then by definition it must be that g_t satisfies:

$$\mu(g_t) \cdot \left(\Pr_{(x,y) \sim \mathcal{D}} [y \leq f(x) | g(x) = 1] - q \right)^2 \geq \alpha$$

Since \mathcal{D} is continuous, it must be that Δ_t is such that:

$$\Pr_{(x,y) \sim \mathcal{D}} [y \leq f(x) + \Delta_t | g_t(x) = 1] = q$$

Finally by the separability of Pinball loss, we have that:

$$\begin{aligned} & PB_q(f_t) - PB_q(f_{t+1}) \\ &= \Pr[g_t(x) = 1] \cdot \left(\mathbb{E}_{(x,y) \sim \mathcal{D}} [L_q(f_t(x), y) - L_q(f_{t+1}(x), y) | g_t(x) = 1] \right) \\ &\geq \mu(g_t) \cdot \frac{\alpha}{2\rho\mu(g_t)} \\ &= \frac{\alpha}{2\rho} \end{aligned}$$

where the inequality follows from Lemma 3.2.2 applied to the conditional distribution $\mathcal{D}|(g_t(x) = 1)$, which must also be ρ -smooth.

Applying this bound iteratively, we have that for every T , $PB_q(f_T) \leq PB_q(f) - T \cdot \frac{\alpha}{2\rho}$. Since when $f(x)$ and y are bounded in $[0, 1]$, $PB_q(f) \leq 1$ and $PB_q(f_T) \geq 0$ it must be that the total number of iterations that the algorithm runs for is bounded by:

$$T \leq \frac{2\rho PB_q(f)}{\alpha} \leq \frac{2\rho}{\alpha}$$

6.2.1 A More Direct Approach to Group Conditional Guarantees

Algorithms 11 and 12 gave us a relatively simple method for obtaining approximate group conditional mean and quantile consistency respectively. These algorithms will be a useful template for our algorithms for mean and quantile multicalibration in the next section — but it turns out that for group conditional consistency (without calibration) there is an even simpler algorithm that gives an even better guarantee. Observe that the “group shift” patches $h(x, f_t; g_t, \Delta_t)$ that Algorithms 11 and 12 apply have an extremely simple form: They add Δ_t to the output of $f_t(x)$ if $x \in \mathcal{G}$ and do nothing otherwise. Since addition is commutative, we can observe that these patches are actually order invariant! Consider any run of Algorithm 11 or 12 for T rounds, and for each group $g \in \mathcal{G}$ define the quantities:

$$\lambda_g = \sum_{t: g_t = g} \Delta_t$$

Then the final model f_T that is output can be seen to have the form:

$$f_T(x) = \hat{f}(x; \lambda) \equiv f(x) + \sum_{g \in \mathcal{G}} \lambda_g \cdot g(x)$$

So to compute a model satisfying group conditional mean consistency, we can just directly optimize over functions that have this form, which is a $|\mathcal{G}|$ dimensional convex optimization described in Algorithm 13 for group conditional mean consistency and Algorithm 14 for group conditional quantile consistency. The only difference between the two algorithms is that we minimize the Brier score for mean consistency and the pinball loss for quantile consistency.

Algorithm 13 Simple-Group-Conditional(f, \mathcal{G})

Let λ^* be a solution to the optimization problem:

$$\text{Minimize}_{\lambda} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\left(\hat{f}(x; \lambda) - y \right)^2 \right]$$

Such that:

$$\hat{f}(x; \lambda) \equiv f(x) + \sum_{g \in \mathcal{G}} \lambda_g \cdot g(x)$$

Output $\hat{f}(x; \lambda^*)$

Algorithm 14 Simple-Quantile-Group-Conditional(f, \mathcal{G}, q)

Let λ^* be a solution to the optimization problem:

$$\text{Minimize}_{\lambda} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[L_q \left(\hat{f}(x; \lambda), y \right) \right]$$

Such that:

$$\hat{f}(x; \lambda) \equiv f(x) + \sum_{g \in \mathcal{G}} \lambda_g \cdot g(x)$$

Output $\hat{f}(x; \lambda^*)$

Theorem 20 Fix any model $f : \mathcal{X} \rightarrow [0, 1]$ and class of groups \mathcal{G} . The model $\hat{f}(x; \lambda^*)$ output by Algorithm 13 satisfies perfect (i.e. 0-approximate) group conditional mean consistency. Moreover, if f_T is the model output by Algorithm 11, then $B(\hat{f}(\cdot; \lambda^*)) \leq B(f_T)$.

Proof 31 Suppose $\hat{f}(x; \lambda^*)$ does not satisfy group conditional mean consistency. Then there must be a group $g \in \mathcal{G}$ such that:

$$\left(\mathbb{E}_{(x,y) \sim \mathcal{D}} [f(x)|g(x) = 1] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y|g(x) = 1] \right)^2 > 0$$

Let $\Delta = \mathbb{E}_{(x,y) \sim \mathcal{D}} [y|g(x) = 1] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [\hat{f}(x; \lambda^*)|g(x) = 1]$ and note that $\Delta \neq 0$. In this case, by Lemma 6.1.1 the model obtained by applying the same patch as in the update rule in Algorithm 11 — i.e. $f'(x) = h(x, \hat{f}(x; \lambda^*), g, \Delta)$ is such that $B(f') < B(\hat{f}(x; \lambda^*))$. But this is a contradiction to the optimality of λ^* . Let $\hat{\lambda}$ be the vector such that for all $g' \neq g$, $\hat{\lambda}_{g'} = \lambda_{g'}^*$, and such that $\hat{\lambda}_g = \lambda_g^* + \Delta$. We can write f' as $f'(x) = \hat{f}(x; \hat{\lambda})$. Since $\hat{\lambda}$ is a feasible solution to the optimization problem in Algorithm 13 — by the optimality of λ^* we must have $B(f(x; \hat{\lambda})) \geq B(f(x; \lambda^*))$.

Similarly, since f_T can be represented as $\hat{f}(x; \lambda)$ for some λ , we have $B(f_T) \geq B(f(x; \lambda^*))$.

Theorem 21 Fix any model $f : \mathcal{X} \rightarrow [0, 1]$, target quantile q , and class of groups \mathcal{G} . The model $\hat{f}(x; \lambda^*)$ output by Algorithm 14 satisfies perfect (i.e. 0-approximate) group conditional quantile consistency with respect to q and \mathcal{G} . Moreover, if f_T is the model output by Algorithm 12, then $PB_q(\hat{f}(\cdot; \lambda^*)) \leq PB_q(f_T)$.

Proof 32 Suppose $\hat{f}(x; \lambda^*)$ does not satisfy group conditional quantile consistency. Then there must be a group $g \in \mathcal{G}$ such that:

$$\Delta = \arg \min_{\Delta} \left(\Pr_{(x,y) \sim \mathcal{D}} [y \leq f_t(x) + \Delta | g(x) = 1] - q \right)^2 > 0$$

In this case, by Lemma 3.2.1 applied to the distribution $\mathcal{D} | (g(x) = 1)$, the model obtained by applying the same patch as in the update rule in Algorithm 12 — i.e. $f'(x) = h(x, \hat{f}(x; \lambda^*), g, \Delta)$ is such that $PB_q(f') < PB_q(\hat{f}(x; \lambda^*))$. But this is a contradiction to the optimality of λ^* . Let $\hat{\lambda}$ be the vector such that for all $g' \neq g$, $\hat{\lambda}_{g'} = \lambda_{g'}^*$ and such that $\hat{\lambda}_g = \lambda_g^* + \Delta$. We can write f' as $f'(x) = \hat{f}(x; \hat{\lambda})$. Since $\hat{\lambda}$ is a feasible solution to the optimization problem in Algorithm 14 — by the optimality of λ^* we must have $PB_q(f(x; \hat{\lambda})) \geq PB_q(f(x; \lambda^*))$.

Similarly, since f_T can be represented as $\hat{f}(x; \lambda)$ for some λ , we have $PB_q(f_T) \geq PB_q(f(x; \lambda^*))$.

6.2.1.1 Generalization

What about out of sample guarantees — i.e. what if we run algorithms 13 and 14 on the empirical distributions on datasets $D \sim \mathcal{D}^n$?

Our generalization theorem will depend on the norm of the solution λ^* output by our algorithms, so it will be helpful for us to study a *regularized* version of these simple algorithms that is guaranteed to output a solution of small norm.

Definition 21 For any vector $v \in \mathbb{R}^d$, the ℓ_1 norm is defined as:

$$\|v\|_1 = \sum_{i=1}^d |v_i|$$

Algorithm 15 Simple-Group-Conditional-Regularized($f, \mathcal{G}, \mathcal{D}, \eta$)

Let λ^* be a solution to the optimization problem:

$$\text{Minimize}_{\lambda} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\left(\hat{f}(x; \lambda) - y \right)^2 \right] + \eta \|\lambda\|_1$$

Such that:

$$\hat{f}(x; \lambda) \equiv f(x) + \sum_{g \in \mathcal{G}} \lambda_g \cdot g(x)$$

Output $\hat{f}(x; \lambda^*)$

Algorithm 15 is identical to Algorithm 13, except that its objective function has been augmented with the regularization term $\eta \|\lambda\|_1$, where η is a parameter of the algorithm. The reason to add this regularization term is to guarantee that the output parameters λ^* will have small norm:

Lemma 6.2.1 *Let $f : \mathcal{X} \rightarrow [0, 1]$ be any model with range $[0, 1]$, let \mathcal{G} be any set of groups, let \mathcal{D} be any distribution over labelled example, and let $\eta > 0$. Then Simple-Group-Conditional-Regularized($f, \mathcal{G}, \mathcal{D}, \eta$) (Algorithm 15) outputs a model $\hat{f}(x, \lambda^*)$ with:*

$$\|\lambda^*\|_1 \leq \frac{1}{\eta}$$

Proof 33 *Suppose otherwise, and we have that $\|\lambda^*\|_1 > \frac{1}{\eta}$. Then since squared error is non-negative, we must have that:*

$$\begin{aligned} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\left(\hat{f}(x; \lambda^*) - y \right)^2 \right] + \eta \|\lambda^*\|_1 &\geq \eta \|\lambda^*\|_1 \\ &> 1 \end{aligned}$$

On the other hand, consider the candidate solution λ_0 , where $\lambda_0 = 0^{|\mathcal{G}|}$ is the all 0's vector. Since the squared error of f is bounded by 1 (since f has range in $[0, 1]$), we have that:

$$\begin{aligned} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\left(\hat{f}(x; \lambda_0) - y \right)^2 \right] + \eta \|\lambda_0\|_1 &= \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[(f(x) - y)^2 \right] + \eta \|\lambda_0\|_1 \\ &\leq 1 + \eta \|\lambda_0\|_1 \\ &= 1 \end{aligned}$$

Thus $\hat{f}(x; \lambda_0)$ has lower objective value than $\hat{f}(x; \lambda^)$, contradicting the optimality of λ^* .*

Ok — so Algorithm 15 produces solutions of small norm. How many small norm solutions are there anyhow? Obviously there are continuously many, so we need a more refined way to ask this question. To do this lets define an ϵ -net.

Definition 22 Let $B(C, d) = \{x \in \mathbb{R}^d : \|x\|_1 \leq C\}$ denote the d dimensional ℓ_1 ball of radius C . Let $N_\epsilon(C, d) \subset B(C, d)$ be some finite subset of the ball. We say that $N_\epsilon(C, d)$ is an ℓ_1 ϵ -net for $B(C, d)$ if for every $x \in B(C, d)$ there is an $x' \in N_\epsilon(C, d)$ such that $\|x - x'\|_1 \leq \epsilon$.

Theorem 22 There is a finite ℓ_1 ϵ -net for $B(C, d)$ that has cardinality:

$$|N_\epsilon(C, d)| \leq \left(1 + \frac{2C}{\epsilon}\right)^d$$

Proof 34 Let $N_\epsilon \subset B(C, d)$ be a maximal subset of points that are ϵ -separated — i.e. such that for all $\lambda, \lambda' \in N_\epsilon$, $\|\lambda - \lambda'\|_1 \geq \epsilon$, and such that no other point from $B(C, d)$ can be added to N_ϵ while maintaining this property. Observe that N_ϵ must be an ϵ -net for $B(C, d)$, since if there were any point $\lambda^* \in B(C, d)$ such that for all $\lambda \in N_\epsilon$, $\|\lambda - \lambda^*\|_1 > \epsilon$, then λ^* could be added to N_ϵ while preserving its ϵ -separation property, which would contradict its maximality.

Consider the union of ℓ_1 balls of radius $\epsilon/2$ centered at each point $\lambda \in N_\epsilon$. Because of the ϵ -separation property of N_ϵ , these balls are disjoint, and so their total volume is the sum of their individual volumes: $|N_\epsilon| \cdot V_{\epsilon/2}^d$, where $V_{\epsilon/2}^d$ is the volume of a d -dimensional ℓ_1 ball of radius $\epsilon/2$. On the other hand, the union of these balls are all contained within a ball of radius $C + \epsilon/2$. Hence:

$$|N_\epsilon| \cdot V_{\epsilon/2}^d \leq V_{C+\epsilon/2}^d$$

and in particular:

$$\begin{aligned} |N_\epsilon| &\leq \frac{V_{C+\epsilon/2}^d}{V_{\epsilon/2}^d} \\ &\leq \left(\frac{C + \frac{\epsilon}{2}}{\frac{\epsilon}{2}}\right)^d \\ &= \left(\frac{2C}{\epsilon} + 1\right)^d \end{aligned}$$

What good is an ϵ -net? It is useful for two reasons. Since it is finite, we can use Hoeffding's inequality together with a union bound to argue that for every parameter vector λ' in the net, our in and out-of-sample objective values are close. And what about for parameter vectors λ that aren't in the net? We argue that they take objective value close to the objective value of the closest parameter vector λ' in the net.

Lemma 6.2.2 Let $\lambda, \lambda' \in B(C, |\mathcal{G}|)$ be such that $\|\lambda - \lambda'\|_1 \leq \epsilon$. Then for all x, y , we have that:

$$(f(x; \lambda) - y)^2 - (f(x; \lambda') - y)^2 \leq 2\epsilon C$$

Proof 35 *We can write:*

$$\begin{aligned}
(f(x; \lambda) - y)^2 - (f(x; \lambda') - y)^2 &= f(x; \lambda)^2 - f(x; \lambda')^2 + 2y(f(x; \lambda') - f(x; \lambda)) \\
&= (f(x; \lambda) - f(x; \lambda'))(f(x; \lambda) + f(x; \lambda') - 2y) \\
&\leq \|\lambda - \lambda'\|_1 (\|\lambda\|_1 + \|\lambda'\|_1) \\
&\leq 2\epsilon C
\end{aligned}$$

We're almost done. First we argue that if $D \sim \mathcal{D}^n$, then if n is sufficiently large, the squared error of any model $f(x; \lambda)$ for $\lambda \in N_\epsilon(C, |\mathcal{G}|)$ is close under both D and \mathcal{D} . In fact this is true for any finite set $S \subseteq B(C, |\mathcal{G}|)$ so long as n scales with the log of $|S|$:

Lemma 6.2.3 *Fix any finite subset $S \subset B(C, |\mathcal{G}|)$ and any $\delta > 0$. Let $D \sim \mathcal{D}^n$ consist of n samples (x, y) . Then with probability $1 - \delta$, for every $\lambda \in S$:*

$$\left| \mathbb{E}_{(x,y) \sim D} [(f(x; \lambda) - y)^2] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x; \lambda) - y)^2] \right| \leq (C + 1)^2 \sqrt{\frac{\ln\left(\frac{2|S|}{\delta}\right)}{2n}}$$

Proof 36 *First observe that since $\lambda \in B(C, |\mathcal{G}|)$, we have that for all x , $-C \leq f(x; \lambda) \leq C$. Thus for all x, y :*

$$(f(x; \lambda) - y)^2 \leq (C + 1)^2$$

We can therefore apply Hoeffding's inequality to conclude that for any fixed $\lambda \in B(C, d)$:

$$\Pr_{D \sim \mathcal{D}^n} \left[\left| \mathbb{E}_{(x,y) \sim D} [(f(x; \lambda) - y)^2] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x; \lambda) - y)^2] \right| \geq t \right] \leq 2 \exp\left(\frac{-2nt^2}{(C + 1)^4}\right)$$

The right hand side evaluates to δ if we take:

$$t = (C + 1)^2 \sqrt{\frac{\ln(2/\delta)}{2n}}$$

Replacing δ with $\delta/|S|$ and union bounding over all $\lambda \in S$ we have that with probability $1 - \delta$, simultaneously for every $\lambda \in S$:

$$\left| \mathbb{E}_{(x,y) \sim D} [(f(x; \lambda) - y)^2] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x; \lambda) - y)^2] \right| \leq (C + 1)^2 \sqrt{\frac{\ln\left(\frac{2|S|}{\delta}\right)}{2n}}$$

We can combine Lemma 6.2.3 (uniform convergence of squared error over points in a finite set) with Lemma 6.2.2 (squared error is Lipschitz) together with the existence of a finite ϵ -net for the ℓ_1 ball (Theorem 22) to obtain a similar claim for all of the (continuously many) vectors $\lambda \in B(C, |\mathcal{G}|)$:

Theorem 23 Fix any $C, \delta, \epsilon > 0$. Let $D \sim \mathcal{D}^n$ consist of n samples (x, y) . Then with probability $1 - \delta$, for every $\lambda \in B(C, |\mathcal{G}|)$:

$$\left| \mathbb{E}_{(x,y) \sim D} [(f(x; \lambda) - y)^2] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x; \lambda) - y)^2] \right| \leq (C+1)^2 \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + \frac{2C}{\epsilon})}{2n}} + 4\epsilon C$$

In particular, choosing $\epsilon = \frac{C}{\sqrt{n}}$ gives:

$$\left| \mathbb{E}_{(x,y) \sim D} [(f(x; \lambda) - y)^2] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x; \lambda) - y)^2] \right| \leq 2(C+1)^2 \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}}$$

Proof 37 Let $N_\epsilon = N_\epsilon(C, |\mathcal{G}|)$ be an ℓ_1 ϵ -net for $B(C, |\mathcal{G}|)$ of size $|N_\epsilon| \leq (1 + \frac{2C}{\epsilon})^{|\mathcal{G}|}$ — which we know exists from Theorem 22. Applying Lemma 6.2.3 with $S = N_\epsilon$, we get that with probability $1 - \delta$, for every $\lambda \in N_\epsilon$:

$$\left| \mathbb{E}_{(x,y) \sim D} [(f(x; \lambda) - y)^2] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x; \lambda) - y)^2] \right| \leq (C+1)^2 \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + \frac{2C}{\epsilon})}{2n}}$$

Now fix any $\lambda \in B(C, |\mathcal{G}|)$ and let $\lambda' = \arg \min_{\hat{\lambda} \in N_\epsilon} \|\hat{\lambda} - \lambda\|_1$. We know from the ϵ -net property that $\|\lambda - \lambda'\|_1 \leq \epsilon$. Applying Lemma 6.2.2 twice we can conclude:

$$\begin{aligned} & \left| \mathbb{E}_{(x,y) \sim D} [(f(x; \lambda) - y)^2] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x; \lambda) - y)^2] \right| \\ & \leq \left| \mathbb{E}_{(x,y) \sim D} [(f(x; \lambda') - y)^2] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x; \lambda') - y)^2] \right| + 4\epsilon C \\ & \leq (C+1)^2 \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + \frac{2C}{\epsilon})}{2n}} + 4\epsilon C \end{aligned}$$

We're now ready to prove our generalization bound for Algorithm 15.

Theorem 24 Fix any $\delta > 0$ and model $f : X \rightarrow [0, 1]$. Let \mathcal{G} be any collection of groups. Let $D \sim \mathcal{D}^n$ consist of n samples (x, y) . Then with probability $1 - \delta$, the model $\hat{f}(x, \lambda^*)$ output by Simple-Group-Conditional-Regularized(f, \mathcal{G}, D, η) (Algorithm 15) satisfies α -approximate group conditional mean consistency on \mathcal{D} whenever $\min_{g \in \mathcal{G}} \mu(g) \geq \alpha$ for:

$$\alpha \leq \eta + 4 \left(\frac{1}{\eta} + 1 \right)^2 \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}}$$

Choosing η to minimize this expression gives:

$$\alpha \leq O \left(\left(\frac{\ln(\frac{1}{\delta}) + |\mathcal{G}| \ln(n)}{n} \right)^{1/6} \right)$$

Proof 38 *Let*

$$\hat{\lambda} = \arg \min_{\lambda} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\left(\hat{f}(x; \lambda) - y \right)^2 \right] + \eta \|\lambda\|_1$$

i.e. the true minimizer of regularized objective function over \mathcal{D} . We know from Lemma 6.2.1 that $\lambda^, \hat{\lambda} \in B(1/\eta, |\mathcal{G}|)$. Hence from Theorem 23 and the fact that λ^* minimizes the objective function on \mathcal{D} , we have that with probability $1 - \delta$:*

$$\begin{aligned} & \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x; \lambda^*) - y)^2] + \eta \|\lambda^*\|_1 \\ \leq & \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x; \lambda^*) - y)^2] + \eta \|\lambda^*\|_1 + 2 \left(\frac{1}{\eta} + 1 \right)^2 \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}} \\ \leq & \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x; \hat{\lambda}) - y)^2] + \eta \|\hat{\lambda}\|_1 + 2 \left(\frac{1}{\eta} + 1 \right)^2 \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}} \\ \leq & \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x; \hat{\lambda}) - y)^2] + \eta \|\hat{\lambda}\|_1 + 4 \left(\frac{1}{\eta} + 1 \right)^2 \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}} \end{aligned}$$

Let α be the minimum value such that $f(x; \lambda^)$ satisfies α -approximate group conditional mean consistency on \mathcal{D} . In other words, there exists a group g such that:*

$$\mu(g) \cdot \left(\mathbb{E}_{\mathcal{D}} [f(x; \lambda^*) - y | g(x) = 1] \right)^2 = \alpha$$

Let $\Delta = \mathbb{E}_{\mathcal{D}} [f(x; \lambda^) - y | g(x) = 1]$, and let $h(x, f(x; \lambda^*), g, \Delta) = f(x, \lambda')$ be the result of applying a patch operation, where $\lambda'_{g'} = \lambda^*_{g'}$ for all $g' \neq g$ and $\lambda'_g = \lambda^*_g + \Delta$. By Lemma 6.1.1, we have that $B(f(x, \lambda^*), \mathcal{D}) - B(f(x, \lambda'), \mathcal{D}) > \alpha$. This will contradict the optimality of $\hat{\lambda}$ above if we have that:*

$$\alpha > \eta (\|\lambda'\|_1 - \|\lambda^*\|_1) + 4 \left(\frac{1}{\eta} + 1 \right)^2 \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}}$$

To avoid the contradiction we must have that:

$$\begin{aligned} \alpha & \leq \eta (\|\lambda'\|_1 - \|\lambda^*\|_1) + 4 \left(\frac{1}{\eta} + 1 \right)^2 \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}} \\ & \leq \eta |\Delta| + 4 \left(\frac{1}{\eta} + 1 \right)^2 \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}} \\ & \leq \eta \sqrt{\frac{\alpha}{\mu(g)}} + 4 \left(\frac{1}{\eta} + 1 \right)^2 \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}} \\ & \leq \eta + 4 \left(\frac{1}{\eta} + 1 \right)^2 \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}} \end{aligned}$$

where the second to last inequality follows from the fact that $|\Delta| = \sqrt{\frac{\alpha}{\mu(g)}}$ and the last inequality follows from the assumption that $\mu(g) \geq \alpha$.

We can carry out a similar analysis of a regularized variant of our algorithm for group conditional quantile consistency:

Algorithm 16 Simple-Quantile-Group-Conditional-Regularized(f, \mathcal{G}, q, η)

Let λ^* be a solution to the optimization problem:

$$\text{Minimize}_{\lambda} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[L_q(\hat{f}(x; \lambda), y) \right] + \eta \|\lambda\|_1$$

Such that:

$$\hat{f}(x; \lambda) \equiv f(x) + \sum_{g \in \mathcal{G}} \lambda_g \cdot g(x)$$

Output $\hat{f}(x; \lambda^*)$

The basic strategy is the same, and so we highlight only the differences. Since Pinball loss is also bounded within $[0, 1]$ when $f(x), y \in [0, 1]$ we continue to have that solutions output by Algorithm 15 are norm bounded:

Lemma 6.2.4 *Let $f : \mathcal{X} \rightarrow [0, 1]$ be any model with range $[0, 1]$, let \mathcal{G} be any set of groups, let \mathcal{D} be any distribution over labelled example, and let $\eta > 0$. Then Simple-Quantile-Group-Conditional-Regularized($f, \mathcal{G}, \mathcal{D}, \eta$) (Algorithm 16) outputs a model $\hat{f}(x, \lambda^*)$ with:*

$$\|\lambda^*\|_1 \leq \frac{1}{\eta}$$

We get an even better Lipschitz bound on the loss function:

Lemma 6.2.5 *Let $\lambda, \lambda' \in B(C, |\mathcal{G}|)$ be such that $\|\lambda - \lambda'\|_1 \leq \epsilon$. Then for all x, y , and for all $q \in [0, 1]$ we have that:*

$$|L_q(f(x; \lambda), y) - L_q(f(x; \lambda'), y)| \leq \epsilon$$

Similarly, since $|L_q(f(x; \lambda), y)| \leq C + 1$ (rather than $(C + 1)^2$) for $\lambda \in B(C, |\mathcal{G}|)$, we get a uniform convergence bound that is improved over our version for squared loss by a factor of $(C + 1)$:

Lemma 6.2.6 *Fix any $q \in [0, 1]$, any finite subset $S \subset B(C, |\mathcal{G}|)$ and any $\delta > 0$. Let $D \sim \mathcal{D}^n$ consist of n samples (x, y) . Then with probability $1 - \delta$, for every $\lambda \in S$:*

$$\left| \mathbb{E}_{(x,y) \sim D} [L_q(f(x; \lambda), y)] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [L_q(f(x; \lambda), y)] \right| \leq (C + 1) \sqrt{\frac{\ln\left(\frac{2|S|}{\delta}\right)}{2n}}$$

Combining these two improved lemmas gives a correspondingly improved uniform convergence theorem over all of $B(C, |\mathcal{G}|)$:

Theorem 25 *Fix any $q \in [0, 1]$ and $C, \delta, \epsilon > 0$. Let $D \sim \mathcal{D}^n$ consist of n samples (x, y) . Then with probability $1 - \delta$, for every $\lambda \in B(C, |\mathcal{G}|)$:*

$$\left| \mathbb{E}_{(x,y) \sim D} [L_q(f(x; \lambda), y)] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [L_q(f(x; \lambda), y)] \right| \leq (C+1) \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + \frac{2C}{\epsilon})}{2n}} + 2\epsilon$$

In particular, choosing $\epsilon = \frac{C}{\sqrt{n}}$ gives:

$$\left| \mathbb{E}_{(x,y) \sim D} [L_q(f(x; \lambda), y)] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [L_q(f(x; \lambda), y)] \right| \leq 2(C+1) \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}}$$

We can now obtain our generalization theorem for quantiles — but we'll need one more assumption. Recall that we have already been assuming that our label distributions have CDFs that are ρ -Lipschitz, which means that they have CDFs F such that $F(\tau) - F(\tau') \leq \rho(\tau - \tau')$. To prove our next generalization theorem, we'll also have to assume that the label distributions are not too flat — that is, that they are σ -anti-Lipschitz:

Definition 23 *A CDF F is σ anti-Lipschitz if for all $\tau \geq \tau'$, we have that: $F(\tau) - F(\tau') \geq \sigma(\tau - \tau')$. We say that a distribution $\mathcal{D} \in \Delta\mathcal{X} \times \mathcal{Y}$ is σ -anti-Lipschitz if all of its conditional label distributions $\mathcal{D}_Y(x)$ have σ -anti-Lipschitz CDFs.*

Theorem 26 *Fix any $\delta > 0$ and model $f : X \rightarrow [0, 1]$. Let \mathcal{G} be any collection of groups. Let $D \sim \mathcal{D}^n$ consist of n samples (x, y) from a distribution \mathcal{D} that is ρ -Lipschitz and σ -anti-Lipschitz. Then with probability $1 - \delta$, the model $\hat{f}(x, \lambda^*)$ output by Simple-Quantile-Group-Conditional-Regularized(f, \mathcal{G}, D, η) (Algorithm 16) satisfies α -approximate group conditional quantile consistency on \mathcal{D} whenever $\min_{g \in \mathcal{G}} \mu(g) \geq \alpha$ for:*

$$\alpha \leq \frac{2\eta\rho}{\sigma} + 8\rho \left(\frac{1}{\eta} + 1 \right) \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}}$$

Choosing η to minimize this expression gives:

$$\alpha \leq O \left(\frac{\rho}{\sqrt{\sigma}} \cdot \left(\frac{\ln(\frac{1}{\delta}) + |\mathcal{G}| \ln(n)}{n} \right)^{1/4} \right)$$

Proof 39 *Let*

$$\hat{\lambda} = \arg \min_{\lambda} \mathbb{E}_{(x,y) \sim \mathcal{D}} [L_q(\hat{f}(x; \lambda), y)] + \eta \|\lambda\|_1$$

i.e. the true minimizer of regularized objective function over \mathcal{D} . We know from Lemma 6.2.4 that $\lambda^, \hat{\lambda} \in B(1/\eta, |\mathcal{G}|)$. Hence from Theorem 25 and the fact that λ^* minimizes the objective function on \mathcal{D} , we have that with probability $1 - \delta$:*

$$\begin{aligned} & \mathbb{E}_{(x,y) \sim \mathcal{D}} [L_q(\hat{f}(x; \lambda^*), y)] + \eta \|\lambda^*\|_1 \\ \leq & \mathbb{E}_{(x,y) \sim \mathcal{D}} [L_q(\hat{f}(x; \lambda^*), y)] + \eta \|\lambda^*\|_1 + 2 \left(\frac{1}{\eta} + 1 \right) \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}} \\ \leq & \mathbb{E}_{(x,y) \sim \mathcal{D}} [L_q(\hat{f}(x; \hat{\lambda}), y)] + \eta \|\hat{\lambda}\|_1 + 2 \left(\frac{1}{\eta} + 1 \right) \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}} \\ \leq & \mathbb{E}_{(x,y) \sim \mathcal{D}} [L_q(\hat{f}(x; \hat{\lambda}), y)] + \eta \|\hat{\lambda}\|_1 + 4 \left(\frac{1}{\eta} + 1 \right) \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}} \end{aligned}$$

Let α be the minimum value such that $f(x; \lambda^)$ satisfies α -approximate group conditional quantile consistency on \mathcal{D} . In other words, there exists a group g such that:*

$$\mu(g) \cdot \left(\Pr_{\mathcal{D}}[y \leq f(x; \lambda^*) - q | g(x) = 1] \right)^2 = \alpha$$

Let Δ be such that $\Pr_{\mathcal{D}}[y \leq f(x; \lambda^) + \Delta | g(x) = 1] = q$, and let $h(x, f(x; \lambda^*), g, \Delta) = f(x, \lambda')$ be the result of applying a patch operation, where $\lambda'_{g'} = \lambda^*_{g'}$ for all $g' \neq g$ and $\lambda'_g = \lambda^*_g + \Delta$. We have that $PB_q(f(x, \lambda^*), \mathcal{D}) - PB_q(f(x, \lambda'), \mathcal{D}) > \frac{\alpha}{2\rho}$. This will contradict the optimality of $\hat{\lambda}$ above if we have that:*

$$\frac{\alpha}{2\rho} > \eta (\|\lambda'\|_1 - \|\lambda^*\|_1) + 4 \left(\frac{1}{\eta} + 1 \right) \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}}$$

To avoid the contradiction we must have that:

$$\begin{aligned} \frac{\alpha}{2\rho} & \leq \eta (\|\lambda'\|_1 - \|\lambda^*\|_1) + 4 \left(\frac{1}{\eta} + 1 \right) \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}} \\ & \leq \eta |\Delta| + 4 \left(\frac{1}{\eta} + 1 \right) \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}} \\ & \leq \frac{\eta}{\sigma} \sqrt{\frac{\alpha}{\mu(g)}} + 4 \left(\frac{1}{\eta} + 1 \right) \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}} \\ & \leq \frac{\eta}{\sigma} + 4 \left(\frac{1}{\eta} + 1 \right) \sqrt{\frac{\ln(\frac{2}{\delta}) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}} \end{aligned}$$

where the second to last inequality follows from the fact that $|\Delta| \leq \frac{1}{\sigma} \sqrt{\frac{\alpha}{\mu(g)}}$ by the anti-Lipschitzness property, and the last inequality follows from the assumption that $\mu(g) \geq \alpha$.

Solving we get:

$$\alpha \leq \frac{2\eta\rho}{\sigma} + 8\rho \left(\frac{1}{\eta} + 1 \right) \sqrt{\frac{\ln\left(\frac{2}{\delta}\right) + |\mathcal{G}| \ln(1 + 2\sqrt{n})}{2n}}$$

6.3 Multicalibration: Group Conditional Calibration

We can go further and simultaneously ask for group conditional mean consistency and calibration. Combined, these two constraints are called multicalibration:

Definition 24 Fix any model $f : \mathcal{X} \rightarrow [0, 1]$ and group $g : \mathcal{X} \rightarrow \{0, 1\}$. The average squared calibration error of f on g is:

$$K_2(f, g, \mathcal{D}) = \sum_{v \in R(f)} \Pr_{(x,y) \sim \mathcal{D}} [f(x) = v | g(x) = 1] \left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f(x) = v, g(x) = 1] \right)^2$$

We say that a model f is α -approximately multicalibrated with respect to a collection of groups \mathcal{G} and a distribution \mathcal{D} if for every group $g \in \mathcal{G}$:

$$K_2(f, g, \mathcal{D}) \leq \frac{\alpha}{\mu(g)}.$$

When \mathcal{D} is clear from context we just write $K_2(f, g)$.

Remark 6.3.1 We can similarly define an ℓ_1 notion of multicalibration. The average calibration error of f on g is:

$$K_1(f, g, \mathcal{D}) = \sum_{v \in R(f)} \Pr_{(x,y) \sim \mathcal{D}} [f(x) = v | g(x) = 1] \left| v - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f(x) = v, g(x) = 1] \right|$$

Similarly we can say that a model f is α -approximately multicalibrated (in the ℓ_1 metric) with respect to a collection of groups \mathcal{G} and a distribution \mathcal{D} if for every group $g \in \mathcal{G}$:

$$K_1(f, g, \mathcal{D}) \leq \frac{\alpha}{\mu(g)}.$$

We can similarly define an ℓ_∞ notion of multicalibration. The relationships between these metrics are the same as they are for marginal calibration.

Just as in our previous cases, we will proceed by starting with an initial model which we will patch:

Definition 25 (Group Value Patch) *Given a model $f : \mathcal{X} \rightarrow [0, 1]$, a group $g : \mathcal{X} \rightarrow \{0, 1\}$ and a pair of values $v, v' \in [0, 1]$, we say that the group value patch applied to f with pair (v, v') and group g is the function:*

$$h(x, f; v \rightarrow v', g) = \begin{cases} v' & f(x) = v \text{ and } g(x) = 1 \\ f(x) & \text{otherwise} \end{cases}$$

Algorithm 17 Multicalibrate($f, \alpha, \mathcal{G}, \mathcal{D}$) (First Attempt)

Let $f_0 = f$ and $t = 0$.

while f_t is not α -approximately multicalibrated with respect to \mathcal{G} : do

Let:

$$(v_t, g_t) \in \arg \max_{(v, g) \in R(f_t) \times \mathcal{G}} \Pr_{(x, y) \sim \mathcal{D}} [f_t(x) = v, g(x) = 1] \left(v - \mathbb{E}_{(x, y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1] \right)^2$$

$$v'_t = \mathbb{E}_{(x, y) \sim \mathcal{D}} [y | f_t(x) = v_t, g_t(x) = 1]$$

Let $f_{t+1} = h(x; f_t, v_t \rightarrow v'_t, g_t)$ and $t = t + 1$.

Output f_t .

Definition 26 *Fix a model f_t and a group $g : \mathcal{X} \rightarrow \{0, 1\}$. For a value $v \in R(f_t)$, we write:*

$$\mu_t(v, g, \mathcal{D}) = \Pr_{(x, y) \sim \mathcal{D}} [f_t(x) = v, g(x) = 1]$$

When \mathcal{D} is clear from context we just write $\mu_t(v, g)$.

Lemma 6.3.1 *Fix any intermediate round t of Algorithm 17 (Multicalibrate).*

We have that:

$$B(f_t) - B(f_{t+1}) = \mu_t(v_t, g_t) \cdot (v_t - v'_t)^2$$

Proof 40 *Since by construction $f_{t+1}(x) = f_t(x)$ for every x such that either $g(x) = 0$ or $f_t(x) \neq v_t$, we have that:*

$$\begin{aligned} B(f_t) - B(f_{t+1}) &= \mu_t(v_t, g_t) \cdot \mathbb{E}_{(x, y) \sim \mathcal{D}} [(f_t(x) - y)^2 - (f_{t+1}(x) - y)^2 | g_t(x) = 1, f_t(x) = v_t] \\ &= \mu_t(v_t, g_t) \cdot \mathbb{E}_{(x, y) \sim \mathcal{D}} [(v_t - y)^2 - (v'_t - y)^2 | g_t(x) = 1, f_t(x) = v_t] \\ &= \mu_t(v_t, g_t) \cdot (v_t - v'_t)^2 \end{aligned}$$

Where the final equality follows from Lemma 5.1.2 and the fact that by definition $v'_t = \mathbb{E}_{(x, y) \sim \mathcal{D}} [y | f_t(x) = v_t, g_t(x) = 1]$.

So far everything is mirroring our past derivations — but there is an important difference here that will complicate things (just a little!) in comparison to our analysis of Algorithm 6 (our calibration algorithm — without groups). The issue is that the updates for Multicalibrate can *increase* the cardinality of the range of our model: i.e. it might be that $|R(f_{t+1})| = |R(f_t)| + 1$. This is problematic for us since our updates change the function at the granularity of a group intersected with an element of $R(f_t)$ — and so as $R(f_t)$ grows, the rate of progress that we make slows down. We will still eventually get to multicalibration (since $\sum_{t=0}^{\infty} 1/(m+t)$ is a divergent series for any m), but we might need a lot of updates.

The fix is to realize that we don't need arbitrary precision to achieve α -approximate multicalibration — we only need some finite precision that depends on α . If we restrict our updates to an appropriately discretized set of m finite values, then the range of f_{t+1} can never grow above m and our problem is solved. This will also be useful to us when it comes time to argue that solving the empirical multi-calibration problem on a modestly sized dataset suffices to solve it out of sample, on the distribution from which the data was drawn.

Definition 27 Let $[1/m]$ denote the set of $m+1$ grid points:

$$\left[\frac{1}{m} \right] = \left\{ 0, \frac{1}{m}, \frac{2}{m}, \dots, \frac{m-1}{m}, 1 \right\}$$

For any value $v \in [0, 1]$ let $\text{Round}(v; m) = \arg \min_{v' \in [1/m]} |v - v'|$ denote the closest grid point to v in $[1/m]$. For a model $f : \mathcal{X} \rightarrow [0, 1]$, let $\text{Round}(f; m)$ denote the function $f'(x) = \text{Round}(f(x); m)$ that simply rounds the output of f to the nearest grid point of $[1/m]$.

Observe that for $v' = \text{Round}(v; m)$ we always have that $|v - v'| \leq \frac{1}{2m}$.

Algorithm 18 Multicalibrate($f, \alpha, \mathcal{G}, \mathcal{D}$)

Let $m = \frac{1}{\alpha}$.

Let $f_0 = \text{Round}(f; m)$ and $t = 0$.

while f_t is not α -approximately multicalibrated with respect to \mathcal{G} : **do**

Let:

$$(v_t, g_t) \in \arg \max_{(v, g) \in R(f_t) \times \mathcal{G}} \Pr_{(x, y) \sim \mathcal{D}} [f_t(x) = v, g(x) = 1] \left(v - \mathbb{E}_{(x, y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1] \right)^2$$

$$\tilde{v}_t = \mathbb{E}_{(x, y) \sim \mathcal{D}} [y | f_t(x) = v_t, g_t(x) = 1] \text{ and } v'_t = \text{Round}(\tilde{v}_t; m)$$

Let $f_{t+1} = h(x; f_t, v_t \rightarrow v'_t, g_t)$ and $t = t + 1$.

Output f_t .

Lets start by proving an approximate variant of Lemma 6.3.1

Lemma 6.3.2 Fix any intermediate round t of Algorithm 18 (Multicalibrate).

We have that:

$$B(f_t) - B(f_{t+1}) \geq \mu_t(v_t, g_t) \cdot \left((v_t - \tilde{v}_t)^2 - \frac{1}{4m^2} \right)$$

Proof 41 Let $\tilde{f}_{t+1} = h(x; f_t, v_t \rightarrow \tilde{v}_t, g_t)$ be the hypothetical update that would have resulted had we not rounded \tilde{v}_t in step t of the algorithm. This is the update that would have resulted from a step of Algorithm 17, and so we can apply Lemma 6.3.1 to conclude that:

$$B(f_t) - B(\tilde{f}_{t+1}) \geq \mu_t(v_t, g_t) \cdot (v_t - \tilde{v}_t)^2$$

We also have that:

$$\begin{aligned} B(f_t) - B(f_{t+1}) &= (B(f_t) - B(\tilde{f}_{t+1})) - (B(f_{t+1}) - B(\tilde{f}_{t+1})) \\ &= \mu_t(v_t, g_t) \cdot (v_t - \tilde{v}_t)^2 - (B(f_{t+1}) - B(\tilde{f}_{t+1})) \end{aligned}$$

And so it remains to upper bound $(B(f_{t+1}) - B(\tilde{f}_{t+1}))$. Let $\Delta = \tilde{v}_t - v'_t$ and note that since $v'_t = \text{Round}(\tilde{v}_t; m)$ we have that $|\Delta| \leq \frac{1}{2m}$. We can calculate:

$$\begin{aligned} B(f_{t+1}) - B(\tilde{f}_{t+1}) &= \mu_t(v_t, g_t) \cdot \mathbb{E}_{(x,y) \sim \mathcal{D}} [(v'_t - y)^2 - (\tilde{v}_t - y)^2 | g_t(x) = 1, f_t(x) = v_t] \\ &= \mu_t(v_t, g_t) \Delta^2 \\ &\leq \frac{\mu_t(v_t, g_t)}{4m^2} \end{aligned}$$

Here the 2nd equality follows from the fact that $\tilde{v}_t = \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v_t, g_t(x) = 1]$. Combining with the above we have:

$$\begin{aligned} B(f_t) - B(f_{t+1}) &= \mu_t(v_t, g_t) \cdot (v_t - \tilde{v}_t)^2 - (B(f_{t+1}) - B(\tilde{f}_{t+1})) \\ &\geq \mu_t(v_t, g_t) \cdot (v_t - \tilde{v}_t)^2 - \frac{\mu_t(v_t, g_t)}{4m^2} \\ &= \mu_t(v_t, g_t) \cdot \left((v_t - \tilde{v}_t)^2 - \frac{1}{4m^2} \right) \end{aligned}$$

Theorem 27 Given any model f , any collection of groups \mathcal{G} , and any $1 \geq \alpha > 0$, Algorithm 18 (Multicalibrate) halts after $T < \frac{4}{\alpha^2}$ many rounds and outputs a model f_T that satisfies α -approximate multicalibration. Moreover if the algorithm runs for T rounds then $B(f_T) < B(f_0) - T \frac{\alpha^2}{4}$.

Proof 42 Consider any intermediate round t of the algorithm. If the algorithm has not halted, it must be because f_t is not α -approximately multicalibrated, and so we know that there exists a group $g \in \mathcal{G}$ such that $K_2(f_t, g) > \frac{\alpha}{\mu(g)}$. In other words:

$$\sum_{v \in R(f_t)} \mu_t(g, v) \left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1] \right)^2 \geq \alpha$$

We know by construction that $|R(f_t)| \leq m+1$ and so by averaging there must exist some $v \in R(f_t)$ such that:

$$\mu_t(g, v) \left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1] \right)^2 \geq \frac{\alpha}{m+1}$$

In particular, since in the algorithm (g_t, v_t) are chosen in the algorithm to jointly maximize the left hand side of this quantity, we know that this inequality holds for the pair (g_t, v_t) :

$$\mu_t(v_t, g_t) \cdot (v_t - \tilde{v}_t)^2 \geq \frac{\alpha}{m+1}$$

By Lemma 6.1.1 we have that:

$$\begin{aligned} B(f_t) - B(f_{t+1}) &\geq \mu_t(v_t, g_t) \cdot \left((v_t - \tilde{v}_t)^2 - \frac{1}{4m^2} \right) \\ &\geq \mu_t(v_t, g_t) \cdot (v_t - \tilde{v}_t)^2 - \frac{1}{4m^2} \\ &\geq \frac{\alpha}{m+1} - \frac{1}{4m^2} \\ &= \frac{\alpha^2}{\alpha+1} - \frac{\alpha^2}{4} \\ &\geq \frac{\alpha^2}{2} - \frac{\alpha^2}{4} \\ &= \frac{\alpha^2}{4} \end{aligned}$$

Iterating we therefore have that $B(f_T) \leq B(f_0) - T \frac{\alpha^2}{4}$ and since $B(f_T)$ and $B(f_0)$ are bounded in $[0, 1]$, we must have that: $T \leq \frac{4}{\alpha^2}$.

Remark 6.3.2 Theorem 27 bounds $B(f_t) - B(f_0)$. But recall that f_0 results from rounding the outputs of f to the nearest multiple of $1/m$, which might increase f 's squared error by as much as $\frac{1}{m} + \frac{1}{4m^2} = \alpha + \frac{\alpha^2}{4}$ if f was very poorly calibrated at the outset. Taking this into account we can also conclude that:

$$B(f_T) < B(f) - T \frac{\alpha^2}{4} + \alpha + \frac{\alpha^2}{4}.$$

6.3.1 What is Multicalibration Good For?

We recall from Section 5.3 that (marginal) calibration is useful in part because it provided a “trustworthy” interface between predictions and decisions. In particular, studied binary outcome spaces $\mathcal{Y} = \{0, 1\}$ and decision makers with action spaces \mathcal{A} and utility functions $u : \mathcal{A} \times \mathcal{Y} \rightarrow [0, 1]$ mapping actions

and outcomes to utilities that they would like to maximize. We showed in Theorem 13 that if forecasts v_t were calibrated, then amongst all policies mapping *forecasts* to *actions*, the policy that simply trusted the forecasts and acted accordingly by playing the best response action (as if the forecast was correct) — $\text{BR}_u(v) = \arg \max_{a \in \mathcal{A}} v \cdot u(a, 1) + (1-v) \cdot u(a, 0)$ — was optimal amongst all policies mapping forecasts to actions. However we noted the weakness of this guarantee — it refers only to policies which map forecasts to actions, independently of any external signal — such as the features x . For decision-makers that choose their action as a function of both the predictions and external context, (marginal) calibration no longer offers any guarantees. Multicalibration on the other hand offers a context-dependent notion of calibration. A multicalibrated predictor f remains calibrated conditional on $g(x) = 1$ for each $g \in \mathcal{G}$. This is a way in which we can get context-conditional guarantees that relate prediction to decision making. This is perhaps most useful if we define groups g as a function of some benchmark policy $P : \mathcal{X} \rightarrow \mathcal{A}$ that the decision maker might consider following. In particular, fixing a policy P , for each action $a \in \mathcal{A}$, let the group $g_{P,a}(x) = \{x : P(x) = a\}$ be the group of features that cause policy P to play action \mathcal{A} . Let $\mathcal{G}_P = \{g_{P,a} : a \in \mathcal{A}\}$ be the set of such groups defined by each action $a \in \mathcal{A}$. What we can show is that for any policy P , if $\mathcal{G}_P \subseteq \mathcal{G}$, and a predictor f is multicalibrated with respect to \mathcal{G} , the not only is best responding to f the best policy mapping predictions to actions, but also that this policy dominates playing the context-dependent policy P .

Theorem 28 *Fix any distribution \mathcal{D} , and let $f : \mathcal{X} \rightarrow [0, 1]$ be a model with bounded calibration error for every group $g \in \mathcal{G}$: $K_1(f, g, \mathcal{D}) \leq \alpha$. Then for any utility function $u : \mathcal{A} \times \{0, 1\} \rightarrow [0, 1]$ and for any policy $P : \mathcal{X} \rightarrow \mathcal{A}$ such that $\mathcal{G}_P \subseteq \mathcal{G}$, we have that:*

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [u(\text{BR}_u(f(x)), y)] \geq \mathbb{E}_{(x,y) \sim \mathcal{D}} [u(P(x), y)] - 2\alpha|\mathcal{A}|$$

In fact, this is a corollary of a stronger claim. For any mapping $\phi : [0, 1] \times \mathcal{A} \rightarrow \mathcal{A}$ we have that:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [u(\text{BR}_u(f(x)), y)] \geq \mathbb{E}_{(x,y) \sim \mathcal{D}} [u(\phi(f(x), P(x)), y)] - 2\alpha|\mathcal{A}|$$

The weaker claim follows from applying the stronger claim with the mapping $\phi(p, a) = a$ that is simply the identity function in its second argument.

Proof 43 *Our multicalibration assumption implies that f is $\alpha/\mu(g)$ -approximately calibrated conditional on membership in each group $g \in \mathcal{G}$. i.e. if we let $\mathcal{D}_g = \mathcal{D}_{|g(x)=1}$ denote the conditional distribution conditional on $g(x) = 1$, then we have that $K_1(f, \mathcal{D}_g) \leq \alpha/\mu(g, \mathcal{D})$.*

We apply Theorem 13 to the conditional distributions $\mathcal{D}_{g_{P,a}}$ for each $a \in \mathcal{A}$. This tells us that for any mapping $\phi_a : [0, 1] \rightarrow \mathcal{A}$:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [u(BR_u(f(x)), y) | P(x) = a] \geq \mathbb{E}_{(x,y) \sim \mathcal{D}} [u(\phi_a(f(x)), y) | P(x) = a] - \frac{2\alpha}{\Pr[P(x) = a]}$$

Summing over all actions $a \in \mathcal{A}$ we can conclude that for all functions $\phi : [0, 1] \times \mathcal{A} \rightarrow \mathcal{A}$

$$\begin{aligned} \mathbb{E}_{(x,y) \sim \mathcal{D}} [u(BR_u(f(x)), y)] &= \sum_{a \in \mathcal{A}} \Pr[P(x) = a] \cdot \mathbb{E}_{(x,y) \sim \mathcal{D}} [u(BR_u(f(x)), y) | P(x) = a] \\ &\geq \left(\sum_{a \in \mathcal{A}} \Pr[P(x) = a] \cdot \mathbb{E}_{(x,y) \sim \mathcal{D}} [u(\phi(f(x), P(a)), y) | P(x) = a] \right) - 2\alpha |\mathcal{A}| \end{aligned}$$

Here, in order to apply Theorem 13, we have noted that on each distribution $\mathcal{D}_{g_{P,a}}$, the mapping $\phi(f(x), P(a))$ is independent of its second argument, since $P(a) = a$ uniformly on this distribution.

So multicalibration with respect to the “level sets” of a class of benchmark policies implies that trusting the forecasts and acting accordingly dominates any other function mapping the forecasts *and the actions of any benchmark policy in the class* to actions, in terms of downstream utility, simultaneously for every downstream utility function.

6.4 Quantile Multicalibration

Similarly, if we let Pinball loss play the role of the Brier score in our analysis, we can derive algorithms for quantile multicalibration:

Definition 28 Fix any model $f : \mathcal{X} \rightarrow [0, 1]$, target quantile q , and group $g : \mathcal{X} \rightarrow \{0, 1\}$. The average squared quantile calibration error of f on g is:

$$Q_2(f, g) = \sum_{v \in R(f)} \Pr_{(x,y) \sim \mathcal{D}} [f(x) = v | g(x) = 1] \left(q - \Pr_{(x,y) \sim \mathcal{D}} [y \leq v | f(x) = v, g(x) = 1] \right)^2$$

We say that a model f is α -approximately quantile multicalibrated with respect to a collection of groups \mathcal{G} and q if for every group $g \in \mathcal{G}$:

$$Q_2(f, g) \leq \frac{\alpha}{\mu(g)}.$$

We will use the same kind of group value patches that we used for mean multicalibration, as well as the same rounding procedure. We get the following algorithm:

Algorithm 19 QuantileMulticalibrate($f, \alpha, q, \mathcal{G}, \rho$)

Let $m = \frac{\rho^2}{2\alpha}$.

Let $f_0 = \text{Round}(f; m)$ and $t = 0$.

while f_t is not α -approximately quantile multicalibrated with respect to \mathcal{G} and q : **do**

Let:

$$(v_t, g_t) \in \arg \max_{(v, g) \in R(f_t) \times \mathcal{G}} \Pr_{(x, y) \sim \mathcal{D}} [f_t(x) = v, g(x) = 1] \left(q - \Pr_{(x, y) \sim \mathcal{D}} [y \leq v | f_t(x) = v, g(x) = 1] \right)^2$$

$$\tilde{v}_t = \arg \min_v \left| \Pr_{(x, y) \sim \mathcal{D}} [y \leq v | f_t(x) = v_t, g_t(x) = 1] - q \right| \text{ and } v'_t = \text{Round}(\tilde{v}_t; m)$$

Let $f_{t+1} = h(x; f_t, v_t \rightarrow v'_t, g_t)$ and $t = t + 1$.

Output f_t .

Lemma 6.4.1 Fix any intermediate round t of Algorithm 19 (QuantileMulticalibrate) run with parameters α, q , and ρ . If \mathcal{D} is ρ -Lipschitz, then We have that:

$$PB_q(f_t) - PB_q(f_{t+1}) \geq \frac{\alpha^2}{2\rho^3}$$

Proof 44 Since the algorithm has not halted at around t , it must be that f_t is not α -approximately quantile multicalibrated, and hence we know that:

$$\Pr_{(x, y) \sim \mathcal{D}} [f_t(x) = v_t, g_t(x) = 1] \left(q - \Pr_{(x, y) \sim \mathcal{D}} [y \leq v_t | f_t(x) = v_t, g_t(x) = 1] \right)^2 \geq \frac{\alpha}{m}$$

Let $\tilde{f}_{t+1} = h(x; f_t, v_t \rightarrow \tilde{v}_t, g_t)$ be the hypothetical update that would have resulted had we not rounded \tilde{v}_t in step t of the algorithm. Since $\Pr_{(x, y) \sim \mathcal{D}} [y \leq v | f_t(x) = v_t, g_t(x) = 1] = q$ we can apply Lemma 3.2.2 to the distribution $\mathcal{D} | (f_t(x) = v_t, g_t(x) = 1)$ (which must also be ρ -Lipschitz) to conclude that:

$$\begin{aligned} & PB_q(f_t) - PB_q(\tilde{f}_{t+1}) \\ &= \Pr[g_t(x) = 1, f_t(x) = v_t] \cdot \left(\mathbb{E}_{(x, y) \sim \mathcal{D}} \left[L_q(f_t(x), y) - L_q(\tilde{f}_{t+1}(x), y) | g_t(x) = 1, f_t(x) = v_t \right] \right) \\ &\geq \mu(g_t, v_t) \cdot \frac{\alpha}{2\rho m \mu(g_t, v_t)} \\ &= \frac{\alpha}{2m\rho} \end{aligned}$$

We also have that:

$$\begin{aligned} PB_q(f_t) - PB_q(f_{t+1}) &= (PB_q(f_t) - PB_q(\tilde{f}_{t+1})) - (PB_q(f_{t+1}) - PB_q(\tilde{f}_{t+1})) \\ &\geq \frac{\alpha}{2m\rho} - (PB_q(f_{t+1}) - PB_q(\tilde{f}_{t+1})) \end{aligned}$$

And so it remains to upper bound $(PB_q(f_{t+1}) - PB_q(\tilde{f}_{t+1}))$. Let $\Delta = \tilde{v}_t - v'_t$ and note that since $v'_t = \text{Round}(\tilde{v}_t; m)$ we have that $|\Delta| \leq \frac{1}{2m}$. From Lemma 3.2.2 we have that:

$$\begin{aligned} PB_q(f_{t+1}) - PB_q(\tilde{f}_{t+1}) &= \mu(g_t, v_t) \cdot \mathbb{E}_{(x,y) \sim \mathcal{D}} [L_q(v'_t, y) - L_q(\tilde{v}_t, y) | g_t(x) = 1, f_t(x) = v_t] \\ &\leq |Q(v'_t) - Q(\tilde{v}_t)| \cdot |\Delta| - \frac{(Q(v'_t) - Q(\tilde{v}_t))^2}{2\rho} \\ &\leq |Q(v'_t) - Q(\tilde{v}_t)| \cdot \frac{1}{2m} - \frac{(Q(v'_t) - Q(\tilde{v}_t))^2}{2\rho} \\ &\leq \frac{\rho}{4m^2} - \frac{\rho}{8m^2} \\ &= \frac{\rho}{8m^2} \end{aligned}$$

where the first inequality follows from Lemma 3.2.2, the second follows from the fact that $\Delta \leq 1/2m$, and the third follows from the fact that by ρ -Lipschitzness, we must have that $|Q(v'_t) - Q(\tilde{v}_t)| \leq \frac{\rho}{2m}$.

Putting it all together we get that:

$$PB_q(f_t) - PB_q(f_{t+1}) \geq \frac{\alpha}{2m\rho} - \frac{\rho}{8m^2} = \frac{\alpha^2}{2\rho^3}$$

Here we use the fact that $m = \frac{\rho^2}{2\alpha}$.

With this progress lemma, we can state the final guarantee for Algorithm 19.

Theorem 29 Fix any model $f : \mathcal{X} \rightarrow [0, 1]$, $\alpha > 0$, $q \in [0, 1]$, \mathcal{G} , and ρ . If the distribution \mathcal{D} is ρ -Lipschitz, then Algorithm 19 (QuantileMulticalibrate) runs for T rounds and outputs a model f_T that is α -approximately quantile multicalibrated with respect to \mathcal{G} and q . Moreover:

$$T \leq \frac{2\rho^3}{\alpha^2}$$

and $PB_q(f_T) \leq PB_q(f_0) - T \frac{\alpha^2}{2\rho^3}$.

Proof 45 Lemma 6.4.1 tells us that at any intermediate round $t < T$ of the algorithm, we have that:

$$PB_q(f_t) - PB_q(f_{t+1}) \geq \frac{\alpha^2}{2\rho^3}$$

Applying this repeatedly we have that:

$$PB_q(f_T) \leq PB_q(f_0) - T \frac{\alpha^2}{2\rho^3}$$

For labels in $[0, 1]$, we have that $PB_q(f_0) \leq 1$ and $PB_q(f_T) \geq 0$. Hence we must have that $T \leq \frac{2\rho^3}{\alpha^2}$.

6.5 Out of Sample Generalization

Thus far we have presented our algorithms for multicalibration as if they have direct access to the distribution \mathcal{D} . In practice, they will not: We will run our algorithms on a finite sample of n points $D \in \mathcal{Z}^n$ to obtain multicalibration on the empirical distribution on the sample — but we will want our multicalibration guarantees to carry over to some other distribution. In this section, we will show that if the n points in D were sampled i.i.d. from any distribution \mathcal{D} , then so long as n is sufficiently large, the guarantees of multicalibration will indeed carry over to \mathcal{D} .

6.5.1 Mean Multicalibration

Imagine that we have a distribution $\mathcal{D} \in \Delta\mathcal{Z}$ and that we have sampled n points i.i.d. from \mathcal{D} to form a dataset $D: D \sim \mathcal{D}^n$. We

Our generalization bounds follow a simple formula: We first argue that for any particular function f_t , if it is multicalibrated on D it is very likely multicalibrated on \mathcal{D} as well. We then argue that for any fixed input model, Algorithm 18 can only output a model from a finite (and boundedly large set), and so we can union bound over all possible output models.

Theorem 30 *Fix any model $f_t : \mathcal{X} \rightarrow [0, 1]$, any $v \in R(f_t)$, and any group $g \in \mathcal{G}$. Let $D \sim \mathcal{D}^n$ consist of n points drawn i.i.d. from \mathcal{D} . Then with probability $1 - \delta$.*

$$\begin{aligned} & \left| \mu_t(g, v, D) \left(v - \mathbb{E}_{(x,y) \sim D} [y | f_t(x) = v, g(x) = 1] \right)^2 - \mu_t(g, v, D) \left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1] \right)^2 \right| \\ & \leq 46 \sqrt{\frac{3\mu_t(g, v, D) \ln(8/\delta)}{n}} + \frac{135 \ln(8/\delta)}{n} \\ & \in O \left(\sqrt{\frac{\mu_t(g, v, D) \ln(1/\delta)}{n}} + \frac{\ln(1/\delta)}{n} \right) \end{aligned}$$

Proof 46 This will be a long slog. We will beat each term into submission using Chernoff bounds in sequence, and then combine the resulting bounds.

First we argue that with high probability, $\mu_t(g, v, D)$ and $\mu_t(g, v, \mathcal{D})$ must be close.

Lemma 6.5.1 Fix any model $f_t : \mathcal{X} \rightarrow [0, 1]$, any $v \in R(f_t)$, and any group $g \in \mathcal{G}$. Let $D \sim \mathcal{D}^n$ consist of n points drawn i.i.d. from \mathcal{D} . Then with probability $1 - \delta$:

$$|\mu_t(g, v, D) - \mu_t(g, v, \mathcal{D})| \leq \sqrt{\frac{3 \ln(2/\delta) \mu_t(g, v, \mathcal{D})}{n}}$$

Proof 47 We can write

$$\mu_t(g, v, D) = \frac{1}{n} \sum_{(x,y) \in D} \mathbb{1}[g(x) = 1, f_t(x) = v]$$

We have both that $0 \leq \mathbb{1}[g(x) = 1, f_t(x) = v] \leq 1$ and that $\mathbb{E}_{D \sim \mathcal{D}^n} [\mu_t(g, v, D)] = \mu_t(g, v, \mathcal{D})$, and so we can apply the Chernoff bound (Theorem 68) to conclude:

$$\Pr_{D \sim \mathcal{D}^n} [|n\mu_t(g, v, D) - n\mu_t(g, v, \mathcal{D})| \geq \eta n \mu_t(g, v, \mathcal{D})] \leq 2 \exp\left(-\frac{\mu_t(g, v, \mathcal{D}) \eta^2}{3}\right)$$

Plugging in $\eta = \sqrt{\frac{3 \ln(2/\delta)}{n \mu_t(g, v, \mathcal{D})}}$ yields:

$$\Pr_{D \sim \mathcal{D}^n} \left[|n\mu_t(g, v, D) - n\mu_t(g, v, \mathcal{D})| \geq \sqrt{3 \ln(2/\delta) n \mu_t(g, v, \mathcal{D})} \right] \leq \delta$$

Dividing by n yields the theorem.

We next consider the term: $\mathbb{E}_{(x,y) \sim D} [y | f_t(x) = v, g(x) = 1]$.

Lemma 6.5.2 Fix any model $f_t : \mathcal{X} \rightarrow [0, 1]$, any $v \in R(f_t)$, and any group $g \in \mathcal{G}$. Let $D \sim \mathcal{D}^n$ consist of n points drawn i.i.d. from \mathcal{D} . Then with probability $1 - \delta$, for any $v \in R(f_t)$ such that $\mu_t(g, v, \mathcal{D}) \geq \frac{12 \ln(4/\delta)}{n}$:

$$\left| \mathbb{E}_{(x,y) \sim D} [y | f_t(x) = v, g(x) = 1] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1] \right| \leq 5 \sqrt{\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})}}$$

Proof 48 We have that:

$$\mathbb{E}_{(x,y) \sim D} [y | f_t(x) = v, g(x) = 1] = \frac{\sum_{(x,y) \in D} y \cdot \mathbb{1}[f_t(x) = v] \mathbb{1}[g(x) = 1]}{n \mu_t(g, v, D)}$$

Both the numerator and the denominator are i.i.d. sums of random variables bounded in $[0, 1]$. So, we can apply the Chernoff bound (Theorem 68) to conclude that with probability $1 - \delta$ we have simultaneously:

$$\left| \sum_{(x,y) \in D} y \cdot \mathbb{1}[f_t(x) = v] \mathbb{1}[g(x) = 1] - n \cdot \mathbb{E}_{(x,y) \sim \mathcal{D}} [y \cdot \mathbb{1}[f_t(x) = v] \mathbb{1}[g(x) = 1]] \right| \leq$$

$$\sqrt{3 \ln(4/\delta) n} \mathbb{E}_{(x,y) \sim \mathcal{D}} [y \cdot \mathbb{1}[f_t(x) = v] \mathbb{1}[g(x) = 1]] \leq \sqrt{3 \ln(4/\delta) n \mu_t(g, v, \mathcal{D})}$$

and

$$|n \mu_t(g, v, D) - n \mu_t(g, v, \mathcal{D})| \leq \sqrt{3 \ln(4/\delta) n \mu_t(g, v, \mathcal{D})}$$

Therefore we have that with probability $1 - \delta$:

$$\begin{aligned} & \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1] \\ = & \frac{\sum_{(x,y) \in D} y \cdot \mathbb{1}[f_t(x) = v] \mathbb{1}[g(x) = 1]}{n \mu_t(g, v, D)} \\ \leq & \frac{n \mathbb{E}_{(x,y) \sim \mathcal{D}} [y \cdot \mathbb{1}[f_t(x) = v] \mathbb{1}[g(x) = 1]] + \sqrt{3 \ln(4/\delta) n \mu_t(g, v, \mathcal{D})}}{n \mu_t(g, v, D) - \sqrt{3 \ln(4/\delta) n \mu_t(g, v, \mathcal{D})}} \\ = & \frac{n \mathbb{E}_{(x,y) \sim \mathcal{D}} [y \cdot \mathbb{1}[f_t(x) = v] \mathbb{1}[g(x) = 1]] + \sqrt{3 \ln(4/\delta) n \mu_t(g, v, \mathcal{D})}}{n \mu_t(g, v, \mathcal{D}) \left(1 - \sqrt{\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})}}\right)} \\ = & \left(\frac{1}{\left(1 - \sqrt{\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})}}\right)} \right) \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1] + \frac{\sqrt{3 \ln(4/\delta)}}{\sqrt{n \mu_t(g, v, \mathcal{D})} \left(1 - \sqrt{\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})}}\right)} \\ \leq & \left(1 + 2 \sqrt{\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})}}\right) \left(\mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1] + \frac{\sqrt{3 \ln(4/\delta)}}{\sqrt{n \mu_t(g, v, \mathcal{D})}} \right) \\ \leq & \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1] + 3 \sqrt{\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})}} + 2 \frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})} \\ \leq & \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1] + 5 \sqrt{\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})}} \end{aligned}$$

Here we have applied Lemma 6.5.1 to move between $\mu_t(g, v, D)$ and $\mu_t(g, v, \mathcal{D})$, and have relied on our assumption that $\sqrt{\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})}} \leq \frac{1}{2}$ to apply the inequality $1/(1-x) \leq 1+2x$ for $0 \leq x \leq 1/2$. The last inequality follows because by our assumption on $\mu_t(g, v, \mathcal{D})$, $\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})} \leq 1$ and hence $\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})} \leq \sqrt{\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})}}$.

We can similarly derive the inequality in the reverse direction to conclude that with probability $1 - \delta$:

$$\left| \mathbb{E}_{(x,y) \sim D} [y | f_t(x) = v, g(x) = 1] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1] \right| \leq 5 \sqrt{\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})}}$$

Onwards! We now propagate our error bounds outwards:

Lemma 6.5.3 Fix any model $f_t : \mathcal{X} \rightarrow [0, 1]$, any $v \in R(f_t)$, and any group

$g \in \mathcal{G}$. Let $D \sim \mathcal{D}^n$ consist of n points drawn i.i.d. from \mathcal{D} . Then with probability $1 - \delta$, for any $v \in R(f_t)$ such that $\mu_t(g, v, \mathcal{D}) \geq \frac{12 \ln(4/\delta)}{n}$:

$$\left| \left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1] \right)^2 - \left(v - \mathbb{E}_{(x,y) \sim D} [y | f_t(x) = v, g(x) = 1] \right)^2 \right| \leq 45 \sqrt{\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})}}$$

Proof 49 We compute:

$$\begin{aligned} & \left| \left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1] \right)^2 - \left(v - \mathbb{E}_{(x,y) \sim D} [y | f_t(x) = v, g(x) = 1] \right)^2 \right| \\ &= \left| 2v \left(\mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1] - \mathbb{E}_{(x,y) \sim D} [y | f_t(x) = v, g(x) = 1] \right) + \right. \\ & \quad \left. \left(\mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1]^2 - \mathbb{E}_{(x,y) \sim D} [y | f_t(x) = v, g(x) = 1]^2 \right) \right| \\ &\leq 2v \left| \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1] - \mathbb{E}_{(x,y) \sim D} [y | f_t(x) = v, g(x) = 1] \right| + \\ & \quad \left| \left(\mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1]^2 - \mathbb{E}_{(x,y) \sim D} [y | f_t(x) = v, g(x) = 1]^2 \right) \right| \\ &\leq 10v \sqrt{\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})}} + \left| \left(\mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1]^2 - \mathbb{E}_{(x,y) \sim D} [y | f_t(x) = v, g(x) = 1]^2 \right) \right| \\ &\leq 10 \sqrt{\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})}} + 10 \sqrt{\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})}} + \left(5 \sqrt{\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})}} \right)^2 \\ &\leq 45 \sqrt{\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})}} \end{aligned}$$

Here we have applied Lemma 6.5.2 twice. The last inequality follows because by our assumption on $\mu_t(g, v, \mathcal{D})$, $\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})} \leq 1$ and hence $\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})} \leq \sqrt{\frac{3 \ln(4/\delta)}{n \mu_t(g, v, \mathcal{D})}}$.

Phew. Lets finish this. Applying Lemma 6.5.1, we have that with probability $1 - \delta/2$:

$$\begin{aligned} & \mu_t(g, v, D) \left(v - \mathbb{E}_{(x,y) \sim D} [y | f_t(x) = v, g(x) = 1] \right)^2 \\ &\leq \left(\mu_t(g, v, \mathcal{D}) + \sqrt{\frac{3 \ln(2/\delta) \mu_t(g, v, \mathcal{D})}{n}} \right) \left(v - \mathbb{E}_{(x,y) \sim D} [y | f_t(x) = v, g(x) = 1] \right)^2 \end{aligned}$$

There are two cases to consider. The first case is when $\mu_t(g, v, \mathcal{D}) < \frac{12 \ln(8/\delta)}{n}$. In this case, since $(v - \mathbb{E}_{(x,y) \sim \mathcal{D}}[y | f_t(x) = v, g(x) = 1])^2 \leq 1$ we have:

$$\mu_t(g, v, \mathcal{D}) \left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}}[y | f_t(x) = v, g(x) = 1] \right)^2 \leq \frac{12 \ln(8/\delta)}{n} + \sqrt{\frac{3 \ln(2/\delta) \mu_t(g, v, \mathcal{D})}{n}}$$

In the remaining case, we can apply Lemma 6.5.3 to continue and conclude that with probability $1 - \delta$:

$$\begin{aligned} & \mu_t(g, v, \mathcal{D}) \left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}}[y | f_t(x) = v, g(x) = 1] \right)^2 \\ & \leq \left(\mu_t(g, v, \mathcal{D}) + \sqrt{\frac{3 \ln(2/\delta) \mu_t(g, v, \mathcal{D})}{n}} \right) \left(\left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}}[y | f_t(x) = v, g(x) = 1] \right)^2 + 45 \sqrt{\frac{3 \ln(8/\delta)}{n \mu_t(g, v, \mathcal{D})}} \right) \\ & \leq \mu_t(g, v, \mathcal{D}) \left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}}[y | f_t(x) = v, g(x) = 1] \right)^2 \\ & \quad + \sqrt{\frac{3 \ln(2/\delta) \mu_t(g, v, \mathcal{D})}{n}} + 45 \sqrt{\frac{3 \mu_t(g, v, \mathcal{D}) \ln(8/\delta)}{n}} + 45 \sqrt{\frac{3 \ln(2/\delta) \mu_t(g, v, \mathcal{D})}{n}} \sqrt{\frac{3 \ln(8/\delta)}{n \mu_t(g, v, \mathcal{D})}} \\ & \leq \mu_t(g, v, \mathcal{D}) \left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}}[y | f_t(x) = v, g(x) = 1] \right)^2 + 46 \sqrt{\frac{3 \mu_t(g, v, \mathcal{D}) \ln(8/\delta)}{n}} + \frac{135 \ln(8/\delta)}{n} \end{aligned}$$

The reverse direction follows the same way:

$$\begin{aligned} \mu_t(g, v, \mathcal{D}) \left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}}[y | f_t(x) = v, g(x) = 1] \right)^2 & \geq \mu_t(g, v, \mathcal{D}) \left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}}[y | f_t(x) = v, g(x) = 1] \right)^2 \\ & \quad - 46 \sqrt{\frac{3 \mu_t(g, v, \mathcal{D}) \ln(8/\delta)}{n}} - \frac{135 \ln(8/\delta)}{n} \end{aligned}$$

which finally gives us our theorem.

Recapping where we are, we have shown that for a *single* model f_t , group g , and value v , the quantities $\mu_t(g, v, \mathcal{D}) (v - \mathbb{E}_{(x,y) \sim \mathcal{D}}[y | f_t(x) = v, g(x) = 1])^2$ evaluated in-sample are close to the corresponding quantities out of sample. But we need a corresponding statement for *every* group $g \in \mathcal{G}$, every $v \in [1/m]$ and every model f that might be output by Algorithm 18. Our solution to this will simply be to *count* all possible combinations of g, v , and f , but in order to do this, we need to understand how many different distinct models might be output by Algorithm 18.

Lemma 6.5.4 Fix any model $f : \mathcal{X} \rightarrow [0, 1]$, any finite collection of groups \mathcal{G} , and any $\alpha > 0$. Then there is a set of models \mathcal{C} such that for every distribution

\mathcal{D} (which might be the empirical distribution over an arbitrary dataset), the model f_t output by $\text{Multicalibrate}(f, \alpha, \mathcal{G}, \mathcal{D})$ is such that $f_t \in C$, and:

$$|C| \leq \left(\frac{|\mathcal{G}|}{\alpha^2} \right)^{\frac{4}{\alpha^2} + 1}$$

Proof 50 Given a run of $\text{Multicalibrate}(f, \alpha, \mathcal{G}, \mathcal{D})$ (Algorithm 18) for T rounds, let $\pi = \{(v_t, v'_t, g_t)\}_{t=1}^T$ denote the record of the quantities (v_t, v'_t, g_t) selected by the algorithm at each round t . Let $\pi^{<t} = \{(v_{t'}, v'_{t'}, g_{t'})\}_{t'=1}^{t-1}$ denote the prefix of this transcript up through round $t-1$. Observe that once we fix $\pi^{<t}$ we have also fixed the model f_t that is defined at the start of round t (independently of the distribution \mathcal{D}). Thus to count models that might be output by $\text{Multicalibrate}(f, \alpha, \mathcal{G}, \mathcal{D})$, it suffices to count transcripts.

We let C denote the set of all models defined by transcripts $\pi^{<T}$ for all $T \leq \frac{4}{\alpha^2}$. Since we know from Theorem 27 that Algorithm 18 halts after at most $T \leq \frac{4}{\alpha^2}$ many rounds, the models output by Algorithm 18 must be contained in C as claimed. It remains to count the set of transcripts of length $T \leq \frac{4}{\alpha^2}$. At each round t , there are $m = 1/\alpha$ possible choices for v_t , $m = 1/\alpha$ possible choices for v'_t , and $|\mathcal{G}|$ possible choices for g_t . Hence the number of transcripts of length T is $\left(\frac{|\mathcal{G}|}{\alpha^2} \right)^T$. Thus we have:

$$|C| \leq \sum_{T=0}^{\frac{4}{\alpha^2}} \left(\frac{|\mathcal{G}|}{\alpha^2} \right)^T \leq \left(\frac{|\mathcal{G}|}{\alpha^2} \right)^{\frac{4}{\alpha^2} + 1}$$

Having counted the number of models that multicalibrate might output, we can apply our union bound:

Theorem 31 Fix any model $f : \mathcal{X} \rightarrow [0, 1]$, any finite collection of groups \mathcal{G} , any $\alpha > 0$ and any $\delta > 0$. Let $D \sim \mathcal{D}^n$ consist of n points drawn i.i.d. from \mathcal{D} . Then with probability $1 - \delta$, simultaneously for every model $f_t : \mathcal{X} \rightarrow [0, 1]$ that can be output by $\text{Multicalibrate}(f, \alpha, \mathcal{G}, D)$ (Algorithm 18), any group $g \in \mathcal{G}$, and any $v \in R(f_t)$:

$$\begin{aligned} & \left| \mu_t(g, v, \mathcal{D}) \left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1] \right)^2 - \mu_t(g, v, D) \left(v - \mathbb{E}_{(x,y) \sim D} [y | f_t(x) = v, g(x) = 1] \right)^2 \right| \\ & \leq 46 \sqrt{\frac{3\mu_t(g, v, \mathcal{D}) \left(\frac{4}{\alpha^2} + 2 \right) \ln \left(\frac{8|\mathcal{G}|}{\alpha^2 \delta} \right)}{n} + \frac{135 \left(\frac{4}{\alpha^2} + 2 \right) \ln \left(\frac{8|\mathcal{G}|}{\alpha^2 \delta} \right)}{n}} \\ & \in O \left(\frac{1}{\alpha} \sqrt{\frac{\mu_t(g, v, \mathcal{D}) \ln \left(\frac{|\mathcal{G}|}{\alpha \delta} \right)}{n}} \right) \end{aligned}$$

Proof 51 From Theorem 30 we have that for any $\delta' > 0$ and any single triple (f_t, g, v) we have that:

$$\left| \mu_t(g, v, \mathcal{D}) \left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1] \right)^2 - \mu_t(g, v, D) \left(v - \mathbb{E}_{(x,y) \sim D} [y | f_t(x) = v, g(x) = 1] \right)^2 \right| \leq 46 \sqrt{\frac{3\mu_t(g, v, \mathcal{D}) \ln(8/\delta)}{n} + \frac{135 \ln(8/\delta)}{n}}$$

We now count the number of triples quantified over in our theorem. Lemma 6.5.4 tells us that the number of models f_t that might be output is at most $\left(\frac{|\mathcal{G}|}{\alpha^2}\right)^{\frac{4}{\alpha^2}+1}$. The number of groups $g \in \mathcal{G}$ is $|\mathcal{G}|$, and the number of values $v \in R(f_t)$ is by construction $m = \frac{1}{\alpha}$. Hence the number of triples is at most:

$$\left(\frac{|\mathcal{G}|}{\alpha^2}\right)^{\frac{4}{\alpha^2}+1} \cdot |\mathcal{G}| \cdot \frac{1}{\alpha} \leq \left(\frac{|\mathcal{G}|}{\alpha^2}\right)^{\frac{4}{\alpha^2}+2}$$

The theorem then follows from invoking Theorem 30 with $\delta' = \frac{\delta}{\left(\frac{|\mathcal{G}|}{\alpha^2}\right)^{\frac{4}{\alpha^2}+2}}$ and then summing the failure probability δ' over all enumerated triples.

We're now ready to state our final generalization theorem:

Theorem 32 Fix any model $f : \mathcal{X} \rightarrow [0, 1]$, any finite collection of groups \mathcal{G} , any $\alpha > 0$ and any $\delta > 0$. Let $D \sim \mathcal{D}^n$ consist of n points drawn i.i.d. from \mathcal{D} . Then with probability $1 - \delta$, the model $f_t : \mathcal{X} \rightarrow [0, 1]$ that is output by $\text{Multicalibrate}(f, \alpha, \mathcal{G}, D)$ (Algorithm 18) is α' approximately multicalibrated with respect to \mathcal{G} and \mathcal{D} for:

$$\alpha' \leq \alpha + \frac{1}{\alpha} \left(\frac{135 \left(\frac{4}{\alpha^2} + 2\right) \ln \left(\frac{8|\mathcal{G}|}{\alpha^2 \delta}\right)}{n} \right) + 46 \sqrt{\frac{3 \left(\frac{4}{\alpha^2} + 2\right) \ln \left(\frac{8|\mathcal{G}|}{\alpha^2 \delta}\right)}{\alpha n}}$$

$$\in O \left(\alpha + \frac{\ln \left(\frac{|\mathcal{G}|}{\alpha^2 \delta}\right)}{\alpha^3 n} + \sqrt{\frac{\ln \left(\frac{|\mathcal{G}|}{\alpha^2 \delta}\right)}{\alpha^3 n}} \right)$$

Remark 6.5.1 Choosing α to optimize the bound from Theorem 32, we get a model f_t that is α' approximately multicalibrated with respect to \mathcal{G} and \mathcal{D} for:

$$\alpha' = \tilde{O} \left(\left(\frac{\ln \left(\frac{|\mathcal{G}|}{\delta}\right)}{n} \right)^{1/5} \right)$$

Proof 52 We need to prove that with probability $1 - \delta$, for every group $g \in \mathcal{G}$:
 $K_2(f_t, g, \mathcal{D}) \leq \frac{\alpha'}{\mu(g, \mathcal{D})}$.

Expanding out the definition of $K_2(f_t, g, \mathcal{D})$ this is equivalent to proving that for every $g \in \mathcal{G}$:

$$\mu(g, \mathcal{D}) K_2(f_t, g, \mathcal{D}) = \sum_{v \in R(f_t)} \mu_t(g, v, \mathcal{D}) \left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f(x) = v, g(x) = 1] \right)^2 \leq \alpha'$$

From Theorem 31 we know that with probability $1 - \delta$ we have that for every $v \in R(f_t)$ and $g \in \mathcal{G}$ we have that:

$$\left| \mu_t(g, v, \mathcal{D}) \left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f_t(x) = v, g(x) = 1] \right)^2 - \mu_t(g, v, D) \left(v - \mathbb{E}_{(x,y) \sim D} [y | f_t(x) = v, g(x) = 1] \right)^2 \right|$$

$$\leq 46 \sqrt{\frac{3\mu_t(g, v, \mathcal{D}) \left(\frac{4}{\alpha^2} + 2 \right) \ln \left(\frac{8|\mathcal{G}|}{\alpha^2 \delta} \right)}{n}} + \frac{135 \left(\frac{4}{\alpha^2} + 2 \right) \ln \left(\frac{8|\mathcal{G}|}{\alpha^2 \delta} \right)}{n}$$

From Theorem 27 we know that (with probability 1), $\mu(g, D) \cdot K_2(f_t, g, D) \leq \alpha$ for every $g \in \mathcal{G}$.

Combining these bounds we have:

$$\begin{aligned}
& \mu(g, \mathcal{D}) K_2(f_t, g, \mathcal{D}) \\
= & \sum_{v \in R(f_t)} \mu_t(g, v, \mathcal{D}) \left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f(x) = v, g(x) = 1] \right)^2 \\
\leq & \sum_{v \in R(f_t)} \mu_t(g, v, \mathcal{D}) \left(v - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f(x) = v, g(x) = 1] \right)^2 + \\
& \sum_{v \in R(f_t)} \left(46 \sqrt{\frac{3\mu_t(g, v, \mathcal{D}) \left(\frac{4}{\alpha^2} + 2\right) \ln\left(\frac{8|\mathcal{G}|}{\alpha^2\delta}\right)}{n}} + \frac{135 \left(\frac{4}{\alpha^2} + 2\right) \ln\left(\frac{8|\mathcal{G}|}{\alpha^2\delta}\right)}{n} \right) \\
\leq & \alpha + \sum_{v \in R(f_t)} \left(46 \sqrt{\frac{3\mu_t(g, v, \mathcal{D}) \left(\frac{4}{\alpha^2} + 2\right) \ln\left(\frac{8|\mathcal{G}|}{\alpha^2\delta}\right)}{n}} + \frac{135 \left(\frac{4}{\alpha^2} + 2\right) \ln\left(\frac{8|\mathcal{G}|}{\alpha^2\delta}\right)}{n} \right) \\
\leq & \alpha + \frac{1}{\alpha} \left(\frac{135 \left(\frac{4}{\alpha^2} + 2\right) \ln\left(\frac{8|\mathcal{G}|}{\alpha^2\delta}\right)}{n} \right) + \sum_{v \in R(f_t)} 46 \sqrt{\frac{3\mu_t(g, v, \mathcal{D}) \left(\frac{4}{\alpha^2} + 2\right) \ln\left(\frac{8|\mathcal{G}|}{\alpha^2\delta}\right)}{n}} \\
\leq & \alpha + \frac{1}{\alpha} \left(\frac{135 \left(\frac{4}{\alpha^2} + 2\right) \ln\left(\frac{8|\mathcal{G}|}{\alpha^2\delta}\right)}{n} \right) + 46 \sqrt{\frac{3\mu_t(g, \mathcal{D}) \left(\frac{4}{\alpha^2} + 2\right) \ln\left(\frac{8|\mathcal{G}|}{\alpha^2\delta}\right)}{\alpha n}} \\
\leq & \alpha + \frac{1}{\alpha} \left(\frac{135 \left(\frac{4}{\alpha^2} + 2\right) \ln\left(\frac{8|\mathcal{G}|}{\alpha^2\delta}\right)}{n} \right) + 46 \sqrt{\frac{3 \left(\frac{4}{\alpha^2} + 2\right) \ln\left(\frac{8|\mathcal{G}|}{\alpha^2\delta}\right)}{\alpha n}}
\end{aligned}$$

Where here we have used the fact that $|R(f_t)| = \frac{1}{\alpha}$, and that because $\sqrt{\cdot}$ is a concave function, the final sum is maximized when $\mu_t(g, v, \mathcal{D}) = \alpha\mu_t(g)$ for each v .

6.5.2 Quantile Multicalibration

We can prove a very similar generalization bound for quantile multicalibration. We elide the details that are for the most part similar to the case of mean multicalibration, and state the final theorem:

Theorem 33 Fix any model $f : \mathcal{X} \rightarrow [0, 1]$, any finite collection of groups \mathcal{G} , any $\alpha > 0$ and any $\delta > 0$. Let $D \sim \mathcal{D}^n$ consist of n points drawn i.i.d. from a ρ -Lipschitz distribution \mathcal{D} . Then with probability $1 - \delta$, the model $f_t : \mathcal{X} \rightarrow [0, 1]$ that is output by `QuantileMulticalibrate`($f, \alpha, q, \mathcal{G}, D$) (Algorithm 19) is

α' approximately quantile multicalibrated with respect to target quantile q and \mathcal{G} and \mathcal{D} for:

$$\alpha' = \alpha + 42 \sqrt{\frac{3\rho^2 \left(\ln\left(\frac{4\pi^2 T^2}{3\delta}\right) + T \ln\left(\frac{\rho^4 |\mathcal{G}|}{\alpha^2}\right) \right)}{2\alpha n}}$$

Remark 6.5.2 Choosing α to optimize the bound from Theorem 32, we get a model f_t that is α' approximately multicalibrated with respect to \mathcal{G} and \mathcal{D} for:

$$\alpha' = \tilde{O} \left(\left(\frac{\rho^3 \ln\left(\frac{\rho^4 |\mathcal{G}|}{\delta}\right)}{n} \right)^{1/5} \right)$$

6.6 Loss Minimization with Group Conditional Accuracy Guarantees

The kinds of bias and calibration measures we have been studying are a bit unusual in machine learning. For example, it's possible to drive them to zero, because the true conditional label distributional quantities (i.e. conditional label means and conditional label quantiles) have no calibration error. This is generally not true for other measures of loss in machine learning, like classification error. Even a “Bayes Optimal” predictor might have to frequently make prediction mistakes in a noisy environment. We've seen that reducing calibration error is compatible with reducing squared loss, but what about if we care about other loss functions?

In this section, we will study standard loss minimization problems and show how to reduce arbitrary group-conditional loss minimization problems to traditional marginal loss minimization problems.

Definition 29 Fixing a feature space \mathcal{X} and a label space \mathcal{Y} , a hypothesis is a function $h : \mathcal{X} \rightarrow \mathcal{Y}$. If \mathcal{H} is a collection of such functions, we call \mathcal{H} a **concept class**.

Example 2 Concept classes might include:

- The set of all linear threshold functions.
- The set of all depth- k decision trees.
- The set of all neural networks of a certain architecture.

Definition 30 A **loss function** is a function $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow [0, 1]$. Given a distribution \mathcal{D} , the loss of a hypothesis h on \mathcal{D} is defined to be:

$$\ell_{\mathcal{D}}(h) = \mathbb{E}_{(x,y) \sim \mathcal{D}}[\ell(h(x), y)].$$

Given a group indicator function $g : \mathcal{X} \rightarrow \{0, 1\}$ we can write the group conditional loss of h as:

$$\ell_{\mathcal{D}}(h, g) = \mathbb{E}_{(x,y) \sim \mathcal{D}}[\ell(h(x), y) | g(x) = 1]$$

Example 3 Common loss functions include:

- **Misclassification error:** $\ell(\hat{y}, y) = \mathbf{1}(\hat{y} \neq y)$.
- **Squared error:** $\ell(\hat{y}, y) = (y - \hat{y})^2$.

We will first observe that it does not take very many samples to estimate the loss of a fixed hypothesis.

Claim 6.6.1 Fix any loss function $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow [0, 1]$ and any distribution $\mathcal{D} \in \Delta \mathcal{Z}$. Fix any hypothesis $h : \mathcal{X} \rightarrow \mathcal{Y}$, and let $D \sim \mathcal{D}^m$ consist of m i.i.d. samples from \mathcal{D} . Then for any $\epsilon, \delta > 0$, if:

$$m \geq \frac{\ln(2/\delta)}{\epsilon^2}$$

with probability $1 - \delta$ over the choice of \mathcal{D} :

$$|\ell_D(h) - \ell_{\mathcal{D}}(h)| \leq \epsilon$$

This follows directly from Hoeffding's inequality (Theorem 67).

Definition 31 An algorithm $A : \mathcal{Z}^n \rightarrow \mathcal{H}$ is a **PAC-learner** for \mathcal{H} and ℓ with sample complexity $m_{\mathcal{H}}(\epsilon, \delta)$ if for every distribution $\mathcal{D} \in \Delta(\mathcal{Z})$, if $S \in \mathcal{Z}^n$ consists of n examples drawn i.i.d. from \mathcal{D} , if $n \geq m_{\mathcal{H}}(\epsilon, \delta)$, then with probability at least $1 - \delta$, $A(S) = h$ such that:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}}[\ell(h(x), y)] \leq \min_{h' \in \mathcal{H}} \mathbb{E}_{(x,y) \sim \mathcal{D}}[\ell(h'(x), y)] + \epsilon.$$

Remark 6.6.1 If \mathcal{H} is a finite hypothesis class, then Claim 6.6.1 implies that finding the hypothesis $h \in \mathcal{H}$ that minimizes empirical loss on a sample D will be a PAC learner for \mathcal{H} with sample complexity:

$$m_{\mathcal{H}}(\epsilon, \delta) = O\left(\frac{\log(|\mathcal{H}|/\delta)}{\epsilon^2}\right)$$

This follows from taking $\delta = \delta/|\mathcal{H}|$ and union bounding over all $h \in \mathcal{H}$. More generally, sample complexity functions will typically have the form

$$m_{\mathcal{H}}(\epsilon, \delta) = \tilde{O}\left(\frac{d(\mathcal{H}) + \log(1/\delta)}{\epsilon^2}\right)$$

where $d(\mathcal{H})$ is some measure of complexity of \mathcal{H} that replaces the $\log |\mathcal{H}|$ term above when \mathcal{H} has infinite cardinality. This is the case for all binary hypothesis classes for example, where in this case \mathcal{H} is the VC-dimension of \mathcal{H} . We'll assume in the following that the sample complexity functions have this form.

Definition 32 A hypothesis h is ϵ -groupwise-optimal with respect to loss ℓ , hypothesis class \mathcal{H} collection of groups \mathcal{G} if for every $g \in \mathcal{G}$:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}}[\ell(h(x), y) | g(x) = 1] \leq \min_{h_g \in \mathcal{H}} \mathbb{E}_{(x,y) \sim \mathcal{D}}[\ell(h_g(x), y) | g(x) = 1] + \epsilon.$$

Groupwise optimality asks for us to have a single hypothesis h that is simultaneously competitive with the best model $h_g \in \mathcal{H}$ for each group $g \in \mathcal{G}$, when evaluated only over that group — i.e. it is responsible for competing against independent benchmarks for each group, even though it itself is only a single hypothesis. A result of this requirement is that typically we will not be able to achieve this with any $h \in \mathcal{H}$ — i.e. to find a hypothesis h that is groupwise optimal with respect to \mathcal{H} will have to be from some richer hypothesis class.

6.6.1 Reducing Multi-Group Optimal Learning to (Marginal) PAC Learning

Our goal will be to reduce the problem of finding a groupwise-optimal hypothesis h to the problem of solving standard PAC-learning problems over \mathcal{H} with just marginal guarantees. If the groups were disjoint — i.e. if for every $x \in \mathcal{X}$, $\sum_{g \in \mathcal{G}} g(x) = 1$, this would be easy. We would simply learn the optimal hypothesis h_g separately for each group, and when a new example x came in, we would classify it according to h_g where g was the unique group such that $g(x) = 1$.

Algorithm 20 Disjoint-Multigroup-Learner(D, \mathcal{G}, A)

Given: A dataset $D \sim \mathcal{D}^m$, a collection of disjoint groups \mathcal{G} and a PAC learning algorithm A for \mathcal{H} .

For each group $g \in \mathcal{G}$, let $D_g = \{(x, y) \in D : g(x) = 1\}$ and let $h_g = A(D_g)$.

Output $h(x) = h_g(x)$ for the unique g such that $g(x) = 1$.

Theorem 34 Fix any distribution \mathcal{D} , loss function ℓ , hypothesis class \mathcal{H} , collection of disjoint groups \mathcal{G} , and PAC learner (for \mathcal{H} and ℓ) A with sample complexity $m_{\mathcal{H}}(\epsilon, \delta) = \tilde{O}\left(\frac{d(\mathcal{H}) + \log(1/\delta)}{\epsilon^2}\right)$. Let $D \sim \mathcal{D}^m$ be a dataset consisting of m i.i.d. samples from \mathcal{D} with:

$$m \geq \tilde{\Omega}\left(\frac{d(\mathcal{H}) + \log(|\mathcal{G}|/\delta)}{\epsilon^2 \cdot \min_{g \in \mathcal{G}} \mu(g, \mathcal{D})}\right)$$

Then with probability $1 - \delta$, the hypothesis h output by Disjoint-Multigroup-Learner(D, \mathcal{G}, A) is ϵ -groupwise-optimal with respect to ℓ , \mathcal{H} , and \mathcal{G} .

Proof 53 Each D_g consists of samples drawn i.i.d. from $\mathcal{D}|_{g(x)=1}$. With probability $1 - \delta/2$, we have that for all $g \in \mathcal{G}$ $|D_g| \geq m_{\mathcal{H}}(\epsilon, \delta/2|\mathcal{G}|)$. So with probability $1 - \delta/2$, for every $g \in \mathcal{G}$ we have $\ell_{\mathcal{D}}(h_g, g) \leq \min_{h \in \mathcal{H}} \ell_{\mathcal{D}}(h, g) + \epsilon$. By construction, $\ell_{\mathcal{D}}(h, g) = \ell_{\mathcal{D}}(h_g, g)$ which completes the proof.

But what if the groups intersect? To solve this problem we study another simple way to “patch” a model, similarly to how we used patches to remove mean and quantile bias from models.

Definition 33 Fixing a base model $f : \mathcal{X} \rightarrow \mathcal{Y}$, a group $g : \mathcal{X} \rightarrow \{0, 1\}$, and a model $h : \mathcal{X} \rightarrow \mathcal{Y}$ to patch f with. Let $\text{Prepend}(f; g, h) = f'$ be the model defined as:

$$f'(x) = \begin{cases} h(x) & \text{if } g(x) = 1 \\ f(x) & \text{otherwise.} \end{cases}$$

The prepend operation is simple. It simply routes an example *either* to a new model h if x is a member of g , or else passes it through to the old model f . The idea is that if our old model f is not yet ϵ -groupwise optimal, there is some group g that witnesses this sub-optimality. Whenever we discover such a group, we will apply the prepend operation to route members of this group to the optimal model h_g for g . This will clearly reduce the loss on g . The difficulty is of course that (because the groups intersect), it might increase the loss on other groups in \mathcal{G} . However, as the following lemma notes, it reduces overall error.

Lemma 6.6.1 Suppose f fails to satisfy ϵ -groupwise optimality with respect to a collection of groups \mathcal{G} , as witnessed by some $g \in \mathcal{G}$:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}}[\ell(h(x), y)|g(x) = 1] \geq \min_{h_g \in \mathcal{H}} \mathbb{E}_{(x,y) \sim \mathcal{D}}[\ell(h_g(x), y)|g(x) = 1] + \epsilon.$$

Let $h_g = \arg \min_{h \in \mathcal{H}} \ell_{\mathcal{D}}(h, g)$ be the hypothesis in \mathcal{H} that is optimal for \mathcal{G} . Let $f' = \text{Prepend}(f; g, h_g)$. Then:

$$\ell_{\mathcal{D}}(f') \leq \ell_{\mathcal{D}}(f) - \mu(g, \mathcal{D}) \cdot \epsilon$$

Proof 54 We can write:

$$\begin{aligned} \ell_{\mathcal{D}}(f') &= \mathbb{E}[\ell(f'(x), y)] \\ &= \Pr[g(x) = 1] \cdot \mathbb{E}[\ell(f'(x), y)|g(x) = 1] + \Pr[g(x) = 0] \cdot \mathbb{E}[\ell(f'(x), y)|g(x) = 0] \\ &= \Pr[g(x) = 1] \cdot \mathbb{E}[\ell(h_g(x), y)|g(x) = 1] + \Pr[g(x) = 0] \cdot \mathbb{E}[\ell(f(x), y)|g(x) = 0] \\ &\geq \Pr[g(x) = 1] \cdot (\mathbb{E}[\ell(f(x), y)|g(x) = 1] - \epsilon) + \Pr[g(x) = 0] \cdot \mathbb{E}[\ell(f(x), y)|g(x) = 0] \\ &= \mathbb{E}[\ell(f(x), y)] - \mu(g, \mathcal{D})\epsilon \end{aligned}$$

Now we can give our reduction from multigroup optimal learning with respect to any class of groups \mathcal{G} and hypothesis class \mathcal{H} to a standard (marginal) PAC learning algorithm over any class \mathcal{H} .

Algorithm 21 Multigroup-Learner($f, D, \mathcal{G}, A, \epsilon, \delta$)

Given: An initial model f , a dataset $D \sim \mathcal{D}^m$, a collection of disjoint groups \mathcal{G} , a PAC learning algorithm A for \mathcal{H} , and multigroup accuracy parameters ϵ, δ .

Let $\mu = \min_{g \in \mathcal{G}} \mu(g, \mathcal{D})$.

Partition D into datasets D_0 and D_1, \dots, D_k for $k = \frac{4}{\epsilon\mu}$ and:

$$|D_0| \geq \tilde{\Omega} \left(\frac{d(\mathcal{H}) + \log(k \cdot |\mathcal{G}|/\delta)}{\epsilon^2 \cdot \mu} \right) \quad |D_i| \geq \tilde{\Omega} \left(\frac{\log(k \cdot |\mathcal{G}|/\delta)}{\epsilon^2 \cdot \mu} \right)$$

For each group $g \in \mathcal{G}$ and $i \leq k$, let $D_i^g = \{(x, y) \in D_i : g(x) = 1\}$ and let $h_g = A(D_0^g)$.

Let $\ell_g^* = \ell_{D_0^g}(h_g)$

Let $f_0 = f$, and $t = 1$.

while TRUE **do**

if there is a $g \in \mathcal{G}$ such that $\ell_{D_i}(f_{t-1}, g) \geq \ell_g^* + \epsilon/2$ **then**

 Let $f_t = \text{Prepend}(f; g, h_g)$, $t = t + 1$.

else

 Break and output f_{t-1} .

Theorem 35 Fix any distribution \mathcal{D} , loss function ℓ , hypothesis class \mathcal{H} , collection of groups \mathcal{G} , parameters $\epsilon, \delta > 0$, and PAC learner (for \mathcal{H} and ℓ) A with sample complexity $m_{\mathcal{H}}(\epsilon, \delta) = \tilde{O} \left(\frac{d(\mathcal{H}) + \log(1/\delta)}{\epsilon^2} \right)$. Let $D \sim \mathcal{D}^m$ be a dataset consisting of m i.i.d. samples from \mathcal{D} with:

$$m \geq \tilde{\Omega} \left(\frac{d(\mathcal{H}) + \frac{1}{\epsilon\mu} \log(|\mathcal{G}|/(\epsilon\mu\delta))}{\epsilon^2 \cdot \min_{g \in \mathcal{G}} \mu(g, \mathcal{D})} \right)$$

Then with probability $1 - \delta$, the hypothesis h output by Multigroup-Learner(D, \mathcal{G}, A) is ϵ -groupwise-optimal with respect to ℓ, \mathcal{H} , and \mathcal{G} .

Proof 55 This sample complexity is large enough so that with probability $1 - \delta/2$, we can partition the dataset D as required in the algorithm, which is sufficient to solve the PAC-learning problem on each D_i^g with parameters $\epsilon/8$ and $\delta/k|\mathcal{G}|$ — i.e. that with probability $1 - \delta/2k$ for each $g \in \mathcal{G}$:

$$\ell_{\mathcal{D}}(h_g, g) \leq \min_{h \in \mathcal{H}} \ell_{\mathcal{D}}(h, g) + \epsilon/8$$

and sufficient so that all of the estimates that we make of hypothesis error are correct up to error $\epsilon/4$ with probability $1 - \delta/2k|\mathcal{G}|$ — i.e. with probability $1 - \delta/2$ for all $g \in \mathcal{G}$ and $t \leq k$:

$$|\ell_g^* - \ell_{\mathcal{D}}(h_g, g)| \leq \epsilon/8 \quad |\ell_{D_i}(f_{t-1}, g) - \ell_{\mathcal{D}}(f_{t-1}, g)| \leq \epsilon/8$$

We will assume for the rest of the argument that these conditions hold.

Under these conditions, if the algorithm halts and outputs a hypothesis, it is ϵ -multigroup-optimal with respect to \mathcal{G} , so it only remains to show that the algorithm halts before k rounds.

If it hasn't halted yet, it is because there exists a $g \in \mathcal{G}$ such that:

$$\ell_{\mathcal{D}_t}(f_{t-1}, g) \geq \ell_g^* + \epsilon/2$$

By our accuracy conditions, we know that:

$$\ell_{\mathcal{D}}(f_{t-1}, g) \geq \min_{h \in \mathcal{H}} \ell_{\mathcal{D}}(h, g) + \epsilon/4$$

So, by Lemma 6.6.1 we have that:

$$\ell_{\mathcal{D}}(f_t) \leq \ell_{\mathcal{D}}(f_{t-1}) - \frac{\mu(g, \mathcal{D})\epsilon}{4} \leq \ell_{\mathcal{D}}(f_{t-1}) - \frac{\mu\epsilon}{4}$$

Since loss is bounded between 0 and 1, this can happen for at most $k = \frac{4}{\epsilon\mu}$ many rounds.

6.7 Sequential Prediction

“nobreak

6.7.1 Achieving Bucketed Calibration

Our goal is to design a sequential prediction algorithm that guarantees (α, m) multicalibration, with α tending to 0 with T against any possible sequence of observations. To this end, we define a surrogate loss function that replaces the max in the definition of bucketed multicalibration with a “softmax” function based on the sums of exponentials, that is analytically better behaved but that is nevertheless a good approximation to the max function.

Definition 34 (Surrogate Loss) For a round $s \leq T$ and a transcript π , recall that $\pi^{\leq s}$ denotes the length s prefix of π . For a group $g \in \mathcal{G}$, and a bucket $i \in [m]$ let:

$$V_s^{g,i} = \sum_{t \in S(\pi^{\leq s}, g, i)} (y_t - p_t)$$

denote the average difference between the predictions p_t and the outcomes y_t on the subsequence of $\pi^{\leq s}$ corresponding to examples from group g and predictions in bucket i .

Fixing a parameter $\eta \in [0, \frac{1}{2}]$, define a surrogate calibration loss function at round s as:

$$L_s(\pi^{\leq s}) = \sum_{\substack{g \in \mathcal{G}, \\ i \in [m]}} (\exp(\eta V_s^{g,i}) + \exp(-\eta V_s^{g,i})).$$

When the transcript $\pi^{\leq s}$ is clear from context, we will simply write L_s .

We will leave η unspecified for now, and choose it later to optimize our bounds. Recall that what we really want to do is upper bound $\max_{G \in \mathcal{G}, i \in [m]} |V_T^{G,i}|$, which corresponds to our calibration loss. Observe that this “soft-max style” function allows us to tightly upper bound our calibration loss:

Observation 6.7.1 For any transcript π_T , and any $\eta \in [0, \frac{1}{2}]$, we have that:

$$\max_{g \in \mathcal{G}, i \in [m]} |V_T^{g,i}| \leq \frac{1}{\eta} \ln(L_T) \leq \max_{g \in \mathcal{G}, i \in [m]} |V_T^{g,i}| + \frac{\ln(2|\mathcal{G}|m)}{\eta}.$$

Proof 56 For the first inequality, note that:

$$\begin{aligned} \max_{g \in \mathcal{G}, i \in [m]} |\eta V_T^{g,i}| &= \ln \left(\exp \left(\max_{g \in \mathcal{G}, i \in [m]} \eta |V_T^{g,i}| \right) \right) \\ &= \ln \left(\max_{g \in \mathcal{G}, i \in [m]} \exp \left(\eta |V_T^{g,i}| \right) \right) \\ &\leq \ln \left(\max_{g \in \mathcal{G}, i \in [m]} \exp \left(\eta V_T^{g,i} \right) + \exp \left(-\eta V_T^{g,i} \right) \right) \\ &\leq \ln \left(\sum_{g \in \mathcal{G}, i \in [m]} \exp \left(\eta V_T^{g,i} \right) + \exp \left(-\eta V_T^{g,i} \right) \right) \\ &= \ln(L_T) \end{aligned}$$

Dividing by η gives the inequality. In the other direction we have that:

$$\begin{aligned} \frac{1}{\eta} \ln(L_T) &= \frac{1}{\eta} \ln \left(\sum_{g \in \mathcal{G}, i \in [m]} \exp \left(\eta V_T^{g,i} \right) + \exp \left(-\eta V_T^{g,i} \right) \right) \\ &\leq \frac{1}{\eta} \ln \left(2|\mathcal{G}|m \cdot \max_{g \in \mathcal{G}, i \in [m]} \exp \left(\eta |V_T^{g,i}| \right) \right) \\ &= \frac{\ln(2|\mathcal{G}|m)}{\eta} + \max_{g \in \mathcal{G}, i \in [m]} |V_T^{g,i}| \end{aligned}$$

So we now feel freed to study the analytically nicer surrogate loss function.

Just as in our derivation of algorithms promising (regular) calibration guarantees against an adversary, we will be interested in bounding the *increase* in our surrogate loss function from round to round.

Definition 35 Fix any partial transcript $\pi^{\leq s+1} = \pi^{\leq s} \circ (x_{s+1}, p_{s+1}, y_{s+1})$. Define:

$$\Delta_{s+1}(\pi^{\leq s+1}) \equiv \Delta_{s+1}(\pi^{\leq s}, (x_{s+1}, p_{s+1}, y_{s+1})) = L_{s+1}(\pi^{\leq s+1}) - L_s(\pi^{\leq s})$$

Our first step is to bound $\Delta_{s+1}(\pi^{\leq s+1})$ in terms of a quantity that is *linear* in p_{s+1} and y_{s+1} .

Lemma 6.7.1 Fix any partial transcript $\pi^{\leq s+1} = \pi^{\leq s} \circ (x_{s+1}, p_{s+1}, y_{s+1})$ such that $p_{s+1} \in B(i)$. Then for any $\eta \leq 1$, we have that:

$$\Delta_{s+1}(\pi^{\leq s+1}) \leq \eta(y_{s+1} - p_{s+1}) \cdot C_s^i(x_{s+1}) + 2\eta^2 L_s(\pi^{\leq s})$$

where:

$$C_s^i(x_{s+1}) = \sum_{g \in \mathcal{G}(x_{s+1})} \exp(\eta V_s^{g,i}) - \exp(-\eta V_s^{g,i})$$

is a constant depending only on $\pi^{\leq s}$ and x_{s+1} .

Proof 57 Observe that our surrogate loss function is a sum of terms each defined by a group $g \in \mathcal{G}$ and a bucket $i \in [m]$, and that at round $s+1$, the change in surrogate loss can be written as a sum over only those groups in $\mathcal{G}(x_{s+1})$ over the bucket i such that $p_{s+1} \in B(i)$, since all other terms in the sum cancel out. Therefore we can write:

$$\begin{aligned} & \Delta_{s+1}(\pi^{\leq s+1}) \\ &= L_{s+1} - L_s \\ &= \sum_{g \in \mathcal{G}(x_{s+1})} \left(\exp(\eta V_{s+1}^{g,i}) - \exp(\eta V_s^{g,i}) + \exp(-\eta V_{s+1}^{g,i}) - \exp(-\eta V_s^{g,i}) \right) \\ &= \sum_{g \in \mathcal{G}(x_{s+1})} \exp(\eta V_s^{g,i}) (\exp(\eta(y_{s+1} - p_{s+1})) - 1) + \exp(-\eta V_s^{g,i}) (\exp(-\eta(y_{s+1} - p_{s+1})) - 1) \\ &\leq \sum_{g \in \mathcal{G}(x_{s+1})} \exp(\eta V_s^{g,i}) (\eta(y_{s+1} - p_{s+1}) + 2\eta^2) + \exp(-\eta V_s^{g,i}) (-\eta(y_{s+1} - p_{s+1}) + 2\eta^2) \\ &= \eta(y_{s+1} - p_{s+1}) \left(\sum_{g \in \mathcal{G}(x_{s+1})} \exp(\eta V_s^{g,i}) - \sum_{g \in \mathcal{G}(x_{s+1})} \exp(-\eta V_s^{g,i}) \right) + \\ & \quad 2\eta^2 \left(\sum_{g \in \mathcal{G}(x_{s+1})} \exp(\eta V_s^{g,i}) + \exp(-\eta V_s^{g,i}) \right) \\ &\leq \eta(y_{s+1} - p_{s+1}) \cdot C_s^i(x_{s+1}) + 2\eta^2 L_s(\pi^{\leq s}) \end{aligned}$$

Here the first inequality follows from the fact $\eta(y_{s+1} - p_{s+1}) \leq \eta$ and that for $|x| \leq 1$, $\exp(x) \leq 1 + x + x^2$.

Our goal is to find a strategy for the learner's choice of p_{s+1} , as a function of both $\pi^{\leq s}$ and x^{s+1} , that will guarantee that $\mathbb{E}_{p_{s+1}} [\Delta_{s+1}(\pi^{\leq s}, (x_{s+1}, p_{s+1}, y_{s+1}))]$ is small for every possible realization of y_{s+1} . We measure calibration over m buckets, but we will allow our learner to play from a larger strategy space $[\frac{1}{rm}] = \{0, \frac{1}{rm}, \dots, \frac{rm-1}{rm}, 1\}$ for some integer $r > 1$. In the end we will see that our bounds get better with larger r , but that the algorithm we design has no dependence on r at all in its running time, so we can imagine r to be an arbitrarily large number.

Lemma 6.7.2 *Fix any transcript $\pi^{\leq s}$ and any x_{s+1} . There is a distribution on predictions $p_{s+1} \in [\frac{1}{rm}]$ such that for every $y_{s+1} \in [0, 1]$:*

$$\mathbb{E}_{p_{s+1}} [\Delta_{s+1}(\pi^{\leq s}, (x_{s+1}, p_{s+1}, y_{s+1}))] \leq L_s(\pi^{\leq s}) \cdot \left(\frac{\eta}{rm} + 2\eta^2 \right)$$

The distribution can be sampled from as follows:

1. If $C_s^i(x_{s+1}) > 0$ for all i then predict $p_{s+1} = 1$
2. If $C_s^i(x_{s+1}) < 0$ for all i then predict $p_{s+1} = 0$
3. Otherwise, find $i^* \in [m-1]$ such that $C_s^{i^*}(x_{s+1})C_s^{i^*+1}(x_{s+1}) \leq 0$ and let $q \in [0, 1]$ be such that

$$q \cdot C_s^{i^*}(x_{s+1}) + (1 - q)C_s^{i^*+1}(x_{s+1}) = 0.$$

Predict $p_{s+1} = \frac{i^*}{m} - \frac{1}{rm}$ with probability q and predict $p_{s+1} = \frac{i^*}{m}$ with probability $(1 - q)$.

Proof 58 *From Lemma 6.7.1 we have that:*

$$\Delta_{s+1}(\pi^{\leq s+1}) \leq \eta(y_{s+1} - p_{s+1}) \cdot C_s^i(x_{s+1}) + 2\eta^2 L_s(\pi^{\leq s})$$

where $p_{s+1} \in B(i)$. So it suffices to prove that:

$$\mathbb{E}_{p_{s+1}} [\eta(y_{s+1} - p_{s+1}) \cdot C_s^{B^{-1}(p_{s+1})}(x_{s+1})] \leq \frac{\eta}{rm} L_s(\pi^{\leq s})$$

where $B^{-1}(p_{s+1}) = i$ is the bucket such that $p_{s+1} \in B(i)$. We do this in cases.

Case 1: $C_s^i(x_{s+1}) > 0$ for all i :

In this case $p_{s+1} = 1$ and we have that $\eta(y_{s+1} - p_{s+1}) \leq 0$. Since $C_s^m(x_{s+1}) > 0$, it follows that $\eta(y_{s+1} - p_{s+1}) \cdot C_s^m(x_{s+1}) \leq 0$.

Case 2: $C_s^i(x_{s+1}) < 0$ for all i :

In this case $p_{s+1} = 0$ and we have that $\eta(y_{s+1} - p_{s+1}) \geq 0$. Since $C_s^1(x_{s+1}) < 0$, it follows that $\eta(y_{s+1} - p_{s+1}) \cdot C_s^1(x_{s+1}) \leq 0$.

Case 3: *Everything Else*

In the remaining case, observe that since the quantities $C_s^i(x_{s+1})$ are neither all positive or all negative, there must exist some bucket i^* such that $C_s^{i^*}(x_{s+1})C_s^{i^*+1}(x_{s+1}) \leq 0$. So sampling from the specified distribution is well defined and we can compute:

$$\begin{aligned}
& \mathbb{E}_{p_{s+1}} [\eta(y_{s+1} - p_{s+1}) \cdot C_s^{B^{-1}(p_{s+1})}(x_{s+1})] \\
&= q \cdot \left(\eta \left(y_{s+1} - \left(\frac{i^*}{m} - \frac{1}{rm} \right) \right) \cdot C_s^{i^*}(x_{s+1}) \right) + (1-q) \cdot \left(\eta \left(y_{s+1} - \frac{i^*}{m} \right) \cdot C_s^{i^*+1}(x_{s+1}) \right) \\
&= \eta \left(y_{s+1} - \frac{i^*}{m} \right) \cdot \left(q C_s^{i^*}(x_{s+1}) + (1-q) C_s^{i^*+1}(x_{s+1}) \right) + \eta q \frac{1}{rm} C_s^{i^*}(x_{s+1}) \\
&= \eta q \frac{1}{rm} C_s^{i^*}(x_{s+1}) \\
&\leq \frac{\eta}{rm} \left(\sum_{g \in \mathcal{G}(x_{s+1})} \exp(\eta V_s^{g,i}) - \exp(-\eta V_s^{g,i}) \right) \\
&\leq \frac{\eta}{rm} L_s(\pi^{\leq s})
\end{aligned}$$

We have a concrete algorithm that implements the prediction strategy laid out in Lemma 5.5.2.

Algorithm 22 Online-Multicalibrated-Predictor(\mathcal{G}, m, r, η)**for** $t = 1$ to T **do**Observe x_t and compute

$$C_{t-1}^i(x_t) = \sum_{g \in \mathcal{G}(x_t)} \exp(\eta V_{t-1}^{g,i}) - \exp(-\eta V_{t-1}^{g,i})$$

for all $i \in [m]$.**if** $C_{t-1}^m(x_t) > 0$ **then**Predict $p_t = 1$.**else if** $C_{t-1}^1(x_t) < 0$ **then**Predict $p_t = 0$.**else**Select $i^* \in [m]$ such that $C_{t-1}^{i^*}(x_t) \cdot C_{t-1}^{i^*+1}(x_t) \leq 0$.Compute $q \in [0, 1]$ such that:

$$q \cdot C_{t-1}^{i^*}(x_t) + (1 - q) \cdot C_{t-1}^{i^*+1}(x_t) = 0$$

Predict $p_t = \frac{i^*}{m} - \frac{1}{rm}$ with probability q and predict $p_t = \frac{i^*}{m}$ with probability $1 - q$.Observe y_t Let $\pi^{<t+1} = \pi^{<t} \circ (x_t, p_t, y_t)$

Lets now analyze the expected calibration loss of Algorithm 22. We start by analyzing the expected surrogate loss:

Lemma 6.7.3 Fix any set of groups \mathcal{G} , $m, r \geq 0$ and $0 \leq \eta \leq 1$. Fix any adversary, which together with Online-Multicalibrated-Predictor(\mathcal{G}, m, r, η) (Algorithm 22) fixes a distribution on transcripts π . We have that:

$$\mathbb{E}_{\pi}[L_T(\pi)] \leq 2|\mathcal{G}|m \cdot \exp\left(\frac{T\eta}{rm} + 2T\eta^2\right)$$

Proof 59 Consider the final round T . From Lemma 6.7.2, we have that for all $\pi^{<T}$, x_T, y_T :

$$\begin{aligned} \mathbb{E}_{p_T}[L_T(\pi^{\leq T})] &= L_{T-1}(\pi^{\leq T-1}) + \mathbb{E}_{p_T}[\Delta_T(\pi^{\leq T-1}, (x_T, p_T, y_T))] \\ &\leq L_{T-1}(\pi^{\leq T-1}) + L_{T-1}(\pi^{\leq T-1}) \left(\frac{\eta}{rm} + 2\eta^2\right) \\ &= L_{T-1} \left(1 + \frac{\eta}{rm} + 2\eta^2\right) \\ &\leq L_{T-1} \exp\left(\frac{\eta}{rm} + 2\eta^2\right) \end{aligned}$$

where the last inequality follows from $1 + x \leq \exp(x)$. Now inductively taking

the expectation with respect to $p_{T-1}, p_{T-2}, \dots, p_1$ we get that:

$$\mathbb{E}_\pi[L_T(\pi)] \leq L_0 \exp\left(\frac{\eta}{rm} + 2\eta^2\right)^T = 2|\mathcal{G}|m \cdot \exp\left(\frac{T\eta}{rm} + 2T\eta^2\right)$$

Since $L_0 = \sum_{g \in \mathcal{G}, i \in [m]} (\exp(0) + \exp(0)) = 2|\mathcal{G}|m$.

We are now ready to state the final guarantee of Algorithm 22.

Theorem 36 Fix any set of groups \mathcal{G} , $m, r \geq 0$. Let $\eta = \sqrt{\frac{\log(2|\mathcal{G}|m)}{2T}} < 1$. Fix any adversary, which together with *Online-Multicalibrated-Predictor*(\mathcal{G}, m, r, η) (Algorithm 22) fixes a distribution on transcripts π . We have that π satisfies (α, m) -multicalibration error with respect to \mathcal{G} where:

$$\mathbb{E}_\pi[\alpha] \leq \frac{1}{rm} + 2\sqrt{\frac{2 \ln(2|\mathcal{G}|m)}{T}}$$

In particular, if we choose $r \geq \frac{\sqrt{T}}{\epsilon m \sqrt{2 \ln(2|\mathcal{G}|m)}}$ then we have:

$$\mathbb{E}_\pi[\alpha] \leq (2 + \epsilon)\sqrt{\frac{2 \ln(2|\mathcal{G}|m)}{T}}$$

Proof 60 Recall that (α, m) -multicalibration corresponds to the requirement that $\max_{g \in \mathcal{G}, i \in [m]} |V_T^{g,i}| \leq \alpha T$. Hence we need to show that:

$$\mathbb{E}_\pi \left[\max_{g \in \mathcal{G}, i \in [m]} |V_T^{g,i}| \right] \leq \frac{T}{rm} + 2\sqrt{2T \ln(2|\mathcal{G}|m)}$$

We can compute:

$$\begin{aligned} \exp\left(\eta \mathbb{E}_\pi \left[\max_{g \in \mathcal{G}, i \in [m]} |V_T^{g,i}| \right]\right) &\leq \mathbb{E}_\pi \left[\exp\left(\eta \max_{g \in \mathcal{G}, i \in [m]} |V_T^{g,i}| \right)\right] \\ &= \mathbb{E}_\pi \left[\max_{g \in \mathcal{G}, i \in [m]} \exp\left(\eta |V_T^{g,i}| \right)\right] \\ &\leq \mathbb{E}_\pi \left[\max_{g \in \mathcal{G}, i \in [m]} \left(\exp\left(\eta V_T^{g,i}\right) + \exp\left(-\eta V_T^{g,i}\right) \right)\right] \\ &\leq \mathbb{E}_\pi \left[\sum_{g \in \mathcal{G}, i \in [m]} \left(\exp\left(\eta V_T^{g,i}\right) + \exp\left(-\eta V_T^{g,i}\right) \right)\right] \\ &= \mathbb{E}_\pi [L_T(\pi)] \\ &\leq 2|\mathcal{G}|m \cdot \exp\left(\frac{T\eta}{rm} + 2T\eta^2\right) \end{aligned}$$

where the first inequality follows from Jensen's inequality and the convexity of

$\exp(x)$, and the last inequality follows from Lemma 6.7.3. Taking the log of both sides and dividing by η gives:

$$\mathbb{E}_{\pi} \left[\max_{g \in \mathcal{G}, i \in [m]} |V_T^{g,i}| \right] \leq \frac{\log(2|\mathcal{G}|m)}{\eta} + \frac{T}{rm} + 2T\eta$$

Plugging in our chosen value of η completes the proof.

Insert high probability bound and online to offline reduction

6.7.2 Obtaining Bucketed Quantile Multicalibration

We can analogously define a “bucketed” definition of quantile multicalibration:

Definition 36 (Bucketed Multicalibration Error in the Sequential Setting)

Fix a collection of groups \mathcal{G} , a transcript $\pi = \{(x_1, p_1, y_1), \dots, (x_T, p_T, y_T)\}$, and a bucket coarseness parameter m . The quantile calibration error of π on a group g with respect to bucketing coarseness m and target quantile q is defined to be:

$$Q_{\infty}(\pi, m, g) = \max_{i \in [m]} \frac{n(\pi, g, i)}{n(\pi, g)} \cdot \left| \frac{\sum_{t \in S(\pi, g, i)} (\mathbb{1}[y_t \leq p_t] - q)}{n(\pi, g, i)} \right|$$

We say that π satisfies (α, m) quantile multicalibration with respect to \mathcal{G} if for every $g \in \mathcal{G}$:

$$Q_{\infty}(\pi, m, g) \leq \frac{\alpha T}{n(\pi, g)}$$

Expanding out the definitions we find that equivalently, π satisfies (α, m) quantile multicalibration error with respect to \mathcal{G} if:

$$\max_{g \in \mathcal{G}, i \in [m]} \left| \sum_{t \in S(\pi, g, i)} (\mathbb{1}[y_t \leq p_t] - q) \right| \leq \alpha T$$

The derivation of an algorithm for online quantile multicalibration closely mimics our derivation for mean multicalibration, so we will state lemmas without proof when the proof is exactly analogous, and focus only on differences.

Definition 37 (Quantile Surrogate Loss) For a round $s \leq T$ and a transcript π , recall that $\pi^{\leq s}$ denotes the length s prefix of π . For a group $g \in \mathcal{G}$, and a bucket $i \in [m]$ redefine:

$$V_s^{g,i} = \sum_{t \in S(\pi^{\leq s}, g, i)} (\mathbb{1}[y_t \leq p_t] - q).$$

Fixing a parameter $\eta \in [0, \frac{1}{2}]$, continue to let the surrogate calibration loss function at round s as:

$$L_s(\pi^{\leq s}) = \sum_{\substack{g \in \mathcal{G}, \\ i \in [m]}} (\exp(\eta V_s^{g,i}) + \exp(-\eta V_s^{g,i})).$$

When the transcript $\pi^{\leq s}$ is clear from context, we will simply write L_s .

We can prove a direct analogue of Lemma 6.7.1 for our new quantile surrogate loss function. All we used previously about the $V_s^{g,i}$ quantities was that they were sums of terms bounded between $[-1, 1]$ which remains true in our quantile reformulation.

Lemma 6.7.4 *Fix any partial transcript $\pi^{\leq s+1} = \pi^{\leq s} \circ (x_{s+1}, p_{s+1}, y_{s+1})$ such that $p_{s+1} \in B(i)$. Then for any $\eta \leq 1$, we have that:*

$$\Delta_{s+1}(\pi^{\leq s+1}) \leq \eta(\mathbb{1}[y_{s+1} \leq p_{s+1}] - q) \cdot C_s^i(x_{s+1}) + 2\eta^2 L_s(\pi^{\leq s})$$

where:

$$C_s^i(x_{s+1}) = \sum_{g \in \mathcal{G}(x_{s+1})} \exp(\eta V_s^{g,i}) - \exp(-\eta V_s^{g,i})$$

is a constant depending only on $\pi^{\leq s}$ and x_{s+1} .

We now come to the only lemma whose statement and proof change — the bound on how much our surrogate loss changes in expectation when we play according to our multicalibration strategy (which does not change). Our bound now depends on the Lipschitz parameter of the underlying distributions played by the adversary, and holds in expectation both over the randomness of our prediction p_{s+1} and over the adversary's choice of labels y_{s+1} .

Lemma 6.7.5 *Fix any transcript $\pi^{\leq s}$ and any x_{s+1} . There is a distribution on predictions $p_{s+1} \in [\frac{1}{rm}]$ such that for every ρ -Lipschitz distribution over $y_{s+1} \in [0, 1]$:*

$$\mathbb{E}_{p_{s+1}, y_{s+1}} [\Delta_{s+1}(\pi^{\leq s}, (x_{s+1}, p_{s+1}, y_{s+1}))] \leq L_s(\pi^{\leq s}) \cdot \left(\frac{\eta}{\rho rm} + 2\eta^2 \right)$$

The distribution can be sampled from as follows:

1. If $C_s^i(x_{s+1}) < 0$ for all i then predict $p_{s+1} = 1$
2. If $C_s^i(x_{s+1}) > 0$ for all i then predict $p_{s+1} = 0$
3. Otherwise, find $i^* \in [m-1]$ such that $C_s^{i^*}(x_{s+1})C_s^{i^*+1}(x_{s+1}) \leq 0$ and let $p \in [0, 1]$ be such that

$$p \cdot C_s^{i^*}(x_{s+1}) + (1-p)C_s^{i^*+1}(x_{s+1}) = 0.$$

Predict $p_{s+1} = \frac{i^*}{m} - \frac{1}{rm}$ with probability q and predict $p_{s+1} = \frac{i^*}{m}$ with probability $(1-q)$.

Proof 61 *From Lemma 6.7.4 we have that:*

$$\Delta_{s+1}(\pi^{\leq s+1}) \leq \eta(\mathbb{1}[y_{s+1} \leq p_{s+1}] - q) \cdot C_s^i(x_{s+1}) + 2\eta^2 L_s(\pi^{\leq s})$$

where $p_{s+1} \in B(i)$. So it suffices to prove that:

$$\mathbb{E}_{p_{s+1}, y_{s+1}} [\eta(\mathbb{1}[y_{s+1} \leq p_{s+1}] - q) \cdot C_s^{B^{-1}(p_{s+1})}(x_{s+1})] \leq \frac{\eta}{\rho rm} L_s(\pi^{\leq s})$$

where $B^{-1}(p_{s+1}) = i$ is the bucket such that $p_{s+1} \in B(i)$. We do this in cases.

Case 1: $C_s^i(x_{s+1}) > 0$ for all i :

In this case $p_{s+1} = 0$ and we have that $\eta(\mathbb{1}[y_{s+1} \leq p_{s+1}] - q) \leq 0$. Since $C_s^1(x_{s+1}) > 0$, it follows that $\eta(\mathbb{1}[y_{s+1} \leq p_{s+1}] - q) \cdot C_s^1(x_{s+1}) \leq 0$.

Case 2: $C_s^i(x_{s+1}) < 0$ for all i :

In this case $p_{s+1} = 1$ and we have that $\eta(\mathbb{1}[y_{s+1} \leq p_{s+1}] - q) \geq 0$. Since $C_s^m(x_{s+1}) < 0$, it follows that $\eta(\mathbb{1}[y_{s+1} \leq p_{s+1}] - q) \cdot C_s^m(x_{s+1}) \leq 0$.

Case 3: Everything Else

In the remaining case, observe that since the quantities $C_s^i(x_{s+1})$ are neither all positive or all negative, there must exist some bucket i^* such that $C_s^{i^*}(x_{s+1})C_s^{i^*+1}(x_{s+1}) \leq 0$. So sampling from the specified distribution is well defined and we can compute:

$$\begin{aligned}
& \mathbb{E}_{p_{s+1}, y_{s+1}} [\eta(\mathbb{1}[y_{s+1} \leq p_{s+1}] - q) \cdot C_s^{B^{-1}(p_{s+1})}(x_{s+1})] \\
&= p \cdot \left(\eta \left(\Pr_{y_{s+1}} \left[y_{s+1} \leq \left(\frac{i^*}{m} - \frac{1}{rm} \right) \right] - q \right) \cdot C_s^{i^*}(x_{s+1}) \right) + \\
&\quad (1-p) \cdot \left(\eta \left(\Pr_{y_{s+1}} \left[y_{s+1} \leq \frac{i^*}{m} \right] - q \right) \cdot C_s^{i^*+1}(x_{s+1}) \right) \\
&\leq p \cdot \left(\eta \left(\Pr_{y_{s+1}} \left[y_{s+1} \leq \left(\frac{i^*}{m} \right) \right] + \frac{1}{\rho rm} - q \right) \cdot C_s^{i^*}(x_{s+1}) \right) + \\
&\quad (1-p) \cdot \left(\eta \left(\Pr_{y_{s+1}} \left[y_{s+1} \leq \frac{i^*}{m} \right] - q \right) \cdot C_s^{i^*+1}(x_{s+1}) \right) \\
&= \eta p \frac{1}{\rho rm} C_s^{i^*}(x_{s+1}) \\
&\leq \frac{\eta}{\rho rm} L_s(\pi^{\leq s})
\end{aligned}$$

With our new Lemma 5.5.4 in hand, the rest follows as before:

Algorithm 23 Online-Quantile-Multicalibrated-Predictor(\mathcal{G}, m, r, η)**for** $t = 1$ to T **do**Observe x_t and compute

$$C_{t-1}^i(x_t) = \sum_{g \in \mathcal{G}(x_t)} \exp(\eta V_{t-1}^{g,i}) - \exp(-\eta V_{t-1}^{g,i})$$

for all $i \in [m]$, with $V_{t-1}^{g,i}$ defined as in Definition 37.**if** $C_{t-1}^m(x_t) < 0$ **then**Predict $p_t = 1$.**else if** $C_{t-1}^1(x_t) > 0$ **then**Predict $p_t = 0$.**else**Select $i^* \in [m]$ such that $C_{t-1}^{i^*}(x_t) \cdot C_{t-1}^{i^*+1}(x_t) \leq 0$.Compute $p \in [0, 1]$ such that:

$$p \cdot C_{t-1}^{i^*}(x_t) + (1 - p) \cdot C_{t-1}^{i^*+1}(x_t) = 0$$

Predict $p_t = \frac{i^*}{m} - \frac{1}{rm}$ with probability p and predict $p_t = \frac{i^*}{m}$ with probability $1 - p$.Observe y_t Let $\pi^{<t+1} = \pi^{<t} \circ (x_t, p_t, y_t)$

We get the following final theorem:

Theorem 37 Fix any set of groups \mathcal{G} , $m, r \geq 0$ and $q \in [0, 1]$. Let $\eta = \sqrt{\frac{\log(2|\mathcal{G}|m)}{2T}} < 1$. Fix any adversary who is constrained to playing ρ -Lipschitz distributions, which together with Online-Quantile-Multicalibrated-Predictor(\mathcal{G}, m, r, η) (Algorithm 23) fixes a distribution on transcripts π . We have that π satisfies (α, m) -quantile-multicalibration error with respect to \mathcal{G} and target quantile q where:

$$\mathbb{E}_\pi[\alpha] \leq \frac{1}{\rho m} + 2\sqrt{\frac{2 \ln(2|\mathcal{G}|m)}{T}}$$

We can similarly prove a high probability version of this theorem:

Theorem 38 Fix any set of groups \mathcal{G} , $m, r \geq 0$ and $q \in [0, 1]$. Let $\eta = \sqrt{\frac{\log(2|\mathcal{G}|m)}{2T}} < 1$. Fix $\delta > 0$. Fix any adversary who is constrained to playing ρ -Lipschitz distributions, which together with Online-Quantile-Multicalibrated-Predictor(\mathcal{G}, m, r, η) (Algorithm 23) fixes a distribution on transcripts π . We have that with probability $1 - \delta$ over the randomness of π , π satisfies (α, m) -

quantile-multicalibration error with respect to \mathcal{G} and target quantile q where:

$$\alpha \leq \frac{1}{prm} + 4\sqrt{\frac{2 \ln\left(\frac{2|\mathcal{G}|m}{\delta}\right)}{T}}$$

References and Further Reading

Multicalibration and Group Conditional Mean Consistency (under the name “multiaccuracy”) were introduced in Hébert-Johnson et al. [2018] using a slightly different definition, roughly corresponding to what we refer to our K_∞ metric. Group conditional mean consistency (multiaccuracy) was further studied in Kim et al. [2019]. Several different multicalibration algorithms have been given in the literature, including one based on analyzing the Lagrangian of a linear program Jung et al. [2021], and one based on constructing a branching program via “split” and “merge” operations Gopalan et al. [2022b], which controls an ℓ_1 variant of multicalibration closely related to our K_1 metric. The algorithm and analysis we give here (which controls multicalibration in the K_2 metric) is based on a variant of the original algorithm given by Hébert-Johnson et al. [2018] together with a rounding operation. The algorithm we give for quantile multicalibration and group conditional quantile consistency is from Jung et al. [2022]. Algorithm 13 — the algorithm for obtaining group conditional mean consistency with a one-shot minimization of squared error is due to Parikshit Gopalan. A different analysis than we give here is given in Gopalan et al. [2022a]. Its quantile analogue (Algorithm 14) was given in Jung et al. [2022]. The generalization bounds we give are new as far as we know. The online algorithm for obtaining multicalibration against an adversary is from Gupta et al. [2022]. The full proof of the generalization theorem giving out of sample guarantees for our batch quantile multicalibration algorithm can be found in Jung et al. [2022]. The online algorithm for obtaining quantile multicalibration against an adversary is an adaptation of Bastani et al. [2022] to the ℓ_∞ setting (Bastani et al. [2022] actually give a bound on an ℓ_2 variant of quantile multicalibration, which is stronger than ℓ_∞ multicalibration, but with a polynomial dependence on $|\mathcal{G}|$. Obtaining a bound on ℓ_2 mean or quantile multicalibration with a logarithmic dependence on $|\mathcal{G}|$ remains open as of this writing.)



7

Interlude: Online Adversarial Optimization and the Minimax Theorem

CONTENTS

7.1	Basic Definitions	119
7.2	The Multiplicative Weights Algorithm	120
7.2.1	Using Multiplicative Weights for Convex Optimization	123
7.3	Zero Sum Games	126
7.4	From Sequential Decision Making to The Minimax Theorem ...	127
7.5	Passing Arbitrary Sequential Statistical Tests	130
7.5.1	A Model	131
7.5.2	No Non-Trivial Statistical Tests	132
7.5.3	Implications	134

In this chapter, we'll take a brief interlude from studying learning with conditional guarantees to develop a framework for online decision making. This will form a powerful tool-kit that we will use to derive algorithms in the online adversarial setting, including for multicalibration and related notions of learning with conditional guarantees.

The foundational algorithmic ideas we will introduce in this chapter will a sequential decision maker to make choices such that their performance, in hindsight, can be related to (and shown to be superior to) benchmark policies of various sorts. This turns out to be closely related to optimal play in zero sum games, and in particular, the fundamental “minimax theorem”, which we will both prove using sequential decision making techniques, and use to derive sequential decision making techniques. In this chapter, we'll derive some of these basic algorithms from first principles, and then show how to use them to solve multi-objective optimization problems of a very general sort.

7.1 Basic Definitions

The simplest setting we will consider involves a sequential decision maker who must choose amongst a set of k actions $\mathcal{A} = \{1, \dots, k\}$ each day t . After

making their choice, the decision maker learns a cost $c_i^t \in [0, 1]$ for each action i , and experiences the cost of the action that they chose. Their goal will be to select actions that result in small cumulative cost. This setting is often called the “Learning from Expert Advice”, which is terminology that comes from imagining that the actions correspond to “experts” whose advice the algorithm must choose between. Formally:

Definition 38 (Learning from Expert Advice) *The learner has an action set $\mathcal{A} = \{1, \dots, k\}$. In rounds $t = 1, \dots, T$:*

1. *With knowledge of c^1, \dots, c^{t-1} , The Learner chooses a distribution over actions $p^t \in \Delta\mathcal{A}$.*

2. *With knowledge of p^1, \dots, p^t , the Adversary chooses a vector of costs $c^t \in [0, 1]^k$.*

3. *The Learner experiences (expected) cost $c_L^t = \mathbb{E}_{i \sim p^t}[c_i^t]$*

We let $\pi^t = \{(p^s, c^s)\}_{s=1}^t$ denote the record of interaction after t rounds. After T rounds of interaction, the cumulative (expected) cost of the learner is $C_L^T = \sum_{t=1}^T c_L^t$. The cumulative cost of each action i is $C_i^T = \sum_{t=1}^T c_i^t$.

Observe that in this setting there is an *adversary*, who may choose costs in arbitrary ways, rather than a distribution over costs that the algorithm could have some hope of learning. As a result, it is not immediately clear how we should evaluate algorithms for the learner in this scenario — there is no fixed “optimal action” or “optimal policy” to learn. Instead, we will compare the accumulated cost of the learner to the accumulated cost that various benchmark policies would have obtained for the same sequence of chosen costs. To the extent that the algorithm has higher cost than one of the benchmark policies, we will say that the algorithm *regrets* not having played that benchmark; our goal will be to design algorithms that do not have regret to any benchmark in some fixed class. The simplest class of benchmark policies is the set of *constant* policies, that always play the same fixed action.

Definition 39 (Regret to the Best Fixed Action) *The learner’s regret to action $i \in \mathcal{A}$ after T rounds is:*

$$\text{Reg}(\pi^T, i) = C_L^T - C_i^T = \sum_{t=1}^T (c_L^t - c_i^t)$$

We say that the learner has regret to the best fixed action bounded by α if:

$$\max_{i \in \mathcal{A}} \text{Reg}(\pi^T, i) \leq \alpha$$

7.2 The Multiplicative Weights Algorithm

Algorithm 24 The Multiplicative Weights Algorithm

For each action $i \in \mathcal{A}$, set $w_i^1 = 1$. Let $W^1 = \sum_i w_i^1$.

for $t = 1$ to T **do**

 Play the distribution p^t defined as:

$$p_i^t = \frac{w_i^t}{W^t}$$

 Observe costs c^t and update weights such that for each $i \in \mathcal{A}$:

$$w_i^{t+1} = w_i^t(1 - \eta c_i^t) \quad W^{t+1} = \sum_i w_i^{t+1}$$

Theorem 39 Fix $\eta \leq \frac{1}{2}$. For every sequence of costs and for every action $i \in \mathcal{A}$, the multiplicative weights algorithm obtains regret to action i :

$$\text{Reg}(\pi^T, i) = C_L^T - C_i^T \leq \frac{\ln k}{\eta} + \eta C_i^T$$

Remark 7.2.1 If we know that $\max_i C_i^T \leq B$, we can set $\eta = \sqrt{\frac{\ln k}{B}}$ and Theorem 39 gives us a regret bound of:

$$\max_{i \in \mathcal{A}} \text{Reg}(\pi^T, i) \leq 2\sqrt{B \cdot \ln(k)}$$

Since costs $c_i^t \in [0, 1]$, we always know that $\max_i C_i^T \leq T$, and so we have a worst-case regret bound of:

$$\max_{i \in \mathcal{A}} \text{Reg}(\pi^T, i) \leq 2\sqrt{T \cdot \ln(k)}$$

for every sequence of costs, without assumptions. This bound grows sublinearly with T , which means that the algorithm promises that its average per-round cost approaches that of the best action in hindsight at a rate of $O\left(\sqrt{\frac{\ln k}{T}}\right)$.

Proof 62 The key to the proof will be to relate the cost c_L^t that the learner experiences at each round to the decrease in weight. From the weight update rule, we have that:

$$W^{t+1} = W^t - \eta \sum_i w_i^t c_i^t = W^t(1 - \eta c_L^t)$$

The last equality in this chain follows from the fact that:

$$c_L^t = \sum_i p_i^t c_i^t = \sum_i \frac{w_i^t}{W^t} c_i^t = \frac{1}{W^t} \sum_i w_i^t c_i^t$$

This gives us a way to express how much the cumulative weight W^t decreases at each round. We still have that $W^1 = k$. When we analyzed the halving algorithm, we also knew that $W^{T+1} \geq 1$, since there was at least one action that never had its weight reduced. That is no longer true — but we can still lower bound W^{T+1} in terms of the final weight w_i^{T+1} of any action i , which we can itself bound by the cumulative cost C_i^{T+1} of action i :

$$W^{T+1} \geq w_i^{T+1} = \prod_{t=1}^T (1 - \eta c_i^t) \geq \prod_{t=1}^T (1 - \eta)^{c_i^t} = (1 - \eta)^{C_i^T}$$

Here we used the fact that for $c \in [0, 1]$, $(1 - \eta)^c \leq (1 - \eta c)$. Now we can calculate:

$$\begin{aligned} (1 - \eta)^{C_i^T} &\leq W^{T+1} \\ &= k \prod_{t=1}^T (1 - \eta c_L^t) \\ &\leq k \prod_{t=1}^T \exp(-\eta c_L^t) \\ &= k \exp\left(-\eta \sum_{t=1}^T c_L^t\right) \\ &= k \exp(-\eta C_L^T) \end{aligned}$$

Taking the log of both sides and solving for C_L^T we have that:

$$C_L^T \leq \frac{1}{\eta} \left(\ln(k) + C_i^T \ln\left(\frac{1}{1 - \eta}\right) \right)$$

For $\eta \leq \frac{1}{2}$, $\ln\left(\frac{1}{1 - \eta}\right) \leq \eta + \eta^2$. Therefore we have:

$$C_L^T \leq \frac{\ln k}{\eta} + (1 + \eta) C_i^T$$

and so:

$$\text{Reg}(\pi^T, i) = C_L^T - C_i^T \leq \frac{\ln k}{\eta} + \eta C_i^T$$

Remark 7.2.2 (Rescaling the Costs) We have analyzed Multiplicative Weights for costs $c_i^t \in [0, 1]$. What if the costs fall into a different range?

What if they can be negative? We make two observations: First, consistently translating the cost vector by adding a fixed constant to each coordinate does not change the regret of a sequence of actions: so if costs are in the range $[-C_1, C_2]$, we can always translate them so that they fall into the range $[0, C]$ for $C = C_1 + C_2$. Next, we can rescale costs by dividing them all by C , which scales them back into the range $[0, 1]$, at which point we can apply the Multiplicative Weights Theorem 39. Of course, scaling the costs down by C correspondingly scales the regret down by C , and so the final regret bound will be C times larger than that stated in Theorem 39. The result is that we can guarantee a regret bound of $2(C_1 + C_2)\sqrt{T \cdot \ln(k)}$ when the costs lie in the range $[-C_1, C_2]$ for any $C_1, C_2 \in \mathbb{R}^{\geq 0}$.

7.2.1 Using Multiplicative Weights for Convex Optimization

Multiplicative weights solves a more general problem than we have described thus far. First, we will make a syntactic observation that multiplicative weights can be described as solving a *linear* optimization problem over the simplex $\Delta[k]$. Observe that the weights p^t maintained by the multiplicative weights algorithm are elements of the simplex: $p^t \in \Delta[k]$. The expected cost that the algorithm experiences at each round can be written as an inner product of p^t and c^t :

$$c_L^T = \mathbb{E}_{i \sim p^t} [c_i^t] = \langle p^t, c^t \rangle$$

. Finally, the loss of the best action in hindsight can be written as:

$$\min_{i \in [k]} C_i^T = \min_{p \in \Delta[k]} \langle p, \sum_{t=1}^T c^t \rangle$$

This last observation is because a linear optimization problem over the simplex always obtains its solution at a vertex, which in this case is just a standard basis vector — i.e. the selection of a single action. So we can restate the theorem we proved in the last section as solving an online *linear* optimization problem:

Theorem 40 *For every sequence of costs $c^1, \dots, c^T \in [-C_1, C_2]^k$, the multiplicative weights algorithm selects vectors $p^1, \dots, p^T \in \Delta[k]$ such that:*

$$\sum_{t=1}^T \langle p^t, c^t \rangle \leq \min_{p \in \Delta[k]} \langle p, \sum_{t=1}^T c^t \rangle + 2(C_1 + C_2)\sqrt{T \ln(k)}$$

.

It turns out we can go beyond this, and use multiplicative weights to solve *convex* optimization problems as well.

Definition 40 A function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is convex if for all $x_1, x_2 \in \mathbb{R}^d$, and for all $0 \leq \alpha \leq 1$:

$$f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2)$$

Linear functions are a special case of convex functions in which the inequality always holds with equality.

Definition 41 A function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is L -Lipschitz in the L_1 norm if for all $x_1, x_2 \in \mathbb{R}^d$:

$$|f(x_1) - f(x_2)| \leq L \|x_1 - x_2\|_1$$

If f is L -Lipschitz for some L we simply say that f is Lipschitz-continuous.

Definition 42 Fix a convex function $f : \mathbb{R}^d \rightarrow \mathbb{R}$. A vector $c \in \mathbb{R}^d$ is a subgradient of f at $x \in \mathbb{R}^d$ if for all $x' \in \mathbb{R}^d$:

$$f(x') - f(x) \geq \langle c, x' - x \rangle$$

If f is differentiable, then the gradient $\nabla f(x) = c$ is always a subgradient of f at x .

The following fact (which is not hard to prove, but we will take as given so as not to be led too far astray) will be useful in conceptualizing the reduction that follows in this section. It states that convex Lipschitz functions have bounded gradients, and vice versa.

Lemma 7.2.1 A convex function f is L -Lipschitz (in the L_1 norm) if and only if for every x in its domain, and for every subgradient c of f at x , $\|c\|_\infty = \max_{i \in [d]} |c_i| \leq L$.

Finally, let us define the online convex optimization problem.

Definition 43 (Online Convex Optimization Over the Simplex) In the L -Lipschitz online convex optimization problem:

1. The Learner has an action space equal to the d -dimensional simplex: $\mathcal{A} = \Delta[d]$, and
2. The Adversary has an action space \mathcal{C} consisting of L -lipschitz convex functions $\ell : \mathcal{A} \rightarrow \mathbb{R}^d$,
3. At each round t , the learner chooses an action $a^t \in \mathcal{A}$ and the adversary chooses a loss function $\ell^t \in \mathcal{C}$. The learner experiences cost $c_L^t = \ell^t(a^t)$.

After realizing a transcript π^T , the regret that the learner experiences to action $a \in \mathcal{A}$ is:

$$\text{Reg}(\pi^T, a) = \sum_{t=1}^T (\ell^t(a^t) - \ell^t(a))$$

We will observe a generic reduction that converts an arbitrary algorithm for online linear optimization (like multiplicative weights) into an algorithm for online (Lipschitz) convex optimization with similar regret bounds.

Algorithm 25 A Reduction from Online Convex Optimization to Online Linear Optimization

Given: An algorithm LinearLearn for d -dimensional online linear optimization.

for $t = 1$ to T **do**

From LinearLearn, obtain action a^t .

From the adversary, obtain L -Lipschitz loss function ℓ^t .

Let $c^t \in [0, L]^d$ be a subgradient of ℓ^t at a^t (if ℓ^t is differentiable, $c^t = \nabla \ell^t(a^t)$).

Feed c^t to LinearLearn as a cost vector.

Theorem 41 Suppose LinearLearn is an online linear optimization algorithm that obtains regret to each action $a \in \mathcal{A}$ bounded by $R(T)$ after T rounds, for all sequences of cost vectors $c^t \in [0, L]^d$. The the reduction in Algorithm 41 obtains regret to all fixed actions bounded as:

$$\text{Reg}(\pi^T, a) = \sum_{t=1}^T (\ell^t(a^t) - \ell^t(a)) \leq R(T)$$

for all sequences of L -Lipschitz convex functions.

Proof 63 Fix a comparison action $a \in \mathcal{A}$. We know from the guarantees of the online linear optimization algorithm that:

$$\begin{aligned} R(T) &\geq \sum_{t=1}^T (\langle a^t, c^t \rangle - \langle a, c^t \rangle) \\ &= \sum_{t=1}^T \langle a^t - a, c^t \rangle \\ &\geq \sum_{t=1}^T \ell^t(a^t) - \ell^t(a) \\ &= \text{Reg}(\pi^T, a) \end{aligned}$$

Here the last inequality follows from the fact that c^t is a subgradient of ℓ^t at a^t and Definition 100.

Thus, for any fixed action space \mathcal{A} and Lipschitz parameter L , we can use (e.g.) Multiplicative Weights to solve the online convex optimization problem with regret bounds scaling as $O(\sqrt{T})$.

Remark 7.2.3 Finally, we note that although we have been discussing online convex minimization, all of these algorithms can be used for online concave maximization, simply by flipping the sign of the loss functions ℓ . If ℓ is convex, then $-\ell$ is concave, and a minimizer of ℓ within \mathcal{A} is a maximizer of $-\ell$ within \mathcal{A} .

7.3 Zero Sum Games

A zero-sum game is a strictly competitive game played between two players. We model this by assigning both players *action sets*, and defining a utility function that one player wants to maximize, and the other player wants to minimize. We will call these the *maximization* player (“Max”) and the *minimization* player (“Min”) respectively.

Definition 44 A zero sum game is defined by:

1. A Maximization player endowed with a closed bounded action set $\mathcal{A}_{max} \subset \mathbb{R}^m$,
2. A Minimization player endowed with a closed bounded action set $\mathcal{A}_{min} \subset \mathbb{R}^n$, and
3. A bounded utility function $u : \mathcal{A}_{max} \times \mathcal{A}_{min} \rightarrow [0, B]$.

How should Max and Min play a zero sum game? Lets start with an easy scenario: Suppose Max already knows what action $a_2 \in \mathcal{A}_{min}$ that Min is going to play. In this case, he should play the action a_1 that maximizes the utility given Min’s action:

$$a_1 \in \arg \max_{a \in \mathcal{A}_{max}} u(a, a_2)$$

Similarly, if Min already knows what action $a_1 \in \mathcal{A}_{max}$ that Max is going to play, she should play the action a_2 that *minimizes* the utility given Max’s actions:

$$a_2 \in \arg \min_{a \in \mathcal{A}_{min}} u(a_1, a)$$

We call these *best responses* for Max and Min respectively:

Definition 45 The set of best responses for Max given an action $a_2 \in \mathcal{A}_{min}$ for Min is:

$$Br_{max}(a_2) = \arg \max_{a \in \mathcal{A}_{max}} u(a, a_2)$$

The set of best responses for Min given an action $a_1 \in \mathcal{A}_{max}$ for Max is:

$$Br_{min}(a_1) = \arg \min_{a \in \mathcal{A}_{min}} u(a_1, a)$$

Remark 7.3.1 *The fact that we have assumed that the action spaces \mathcal{A}_{max} and \mathcal{A}_{min} are closed and bounded implies that the best response sets are well defined.*

If Max and Min are playing a pair of actions (a_1, a_2) , and either player is not playing a best response to their opponent's action, then they will wish to change their action to a best response. If neither player wishes to change their action, we call this pair of actions an *equilibrium*.

Definition 46 *A pair of actions $(a_1, a_2) \in \mathcal{A}_{max} \times \mathcal{A}_{min}$ are a Nash Equilibrium if both:*

$$a_1 \in Br_{max}(a_2) \quad a_2 \in Br_{min}(a_1)$$

7.4 From Sequential Decision Making to The Minimax Theorem

An equilibrium is a fixed point of simultaneous play, and so in general its not clear how to go about computing one. Lets start with what seems like an easier problem: How should Max play if he must first commit to his action a_1 and announce it to Min, who will then get an opportunity to best respond? Knowing that Min will play a best response, he should anticipate this, and play so as to *maximize* his utility after Min chooses her action to *minimize* it. That is, he should play a “maximin” strategy. Similarly, if Min must go first and commit to her action and let Max best respond, she should play a “minimax” strategy:

Definition 47 *a_1 is a maximin strategy for Max if:*

$$a_1 \in \arg \max_{a \in \mathcal{A}_{max}} \min_{a_2 \in \mathcal{A}_{min}} u(a, a_2)$$

Similarly, a_2 is a minimax strategy for Min if:

$$a_2 \in \arg \min_{a \in \mathcal{A}_{min}} \max_{a_1 \in \mathcal{A}_{max}} u(a_1, a)$$

We can similarly think about the minimax and maximin *values* of the game: what utility Max and Min respectively can guarantee if they must commit to their strategies up front and announce them:

Definition 48 *The minimax value of a game $v_{minimax}$ is:*

$$v_{minimax} = \min_{a_2 \in \mathcal{A}_{min}} \max_{a_1 \in \mathcal{A}_{max}} u(a_1, a_2)$$

The maximin value of a game $v_{maximin}$ is:

$$v_{maximin} = \max_{a \in \mathcal{A}_{max}} \min_{a_2 \in \mathcal{A}_{min}} u(a, a_2)$$

In a zero sum game, it can only be a disadvantage to go first, intuitively because you are revealing information to your opponent, without restricting her action space. This means that the minimax value of the game can only be larger than the maximin value:

Lemma 7.4.1 *In any zero sum game:*

$$v_{\minimax} \geq v_{\maximin}$$

Proof 64 *Let a_1 and a_2 be maximin and minimax strategies for Max and Min respectively. Then:*

$$v_{\minimax} = \max_{a \in \mathcal{A}_{\max}} u(a, a_2) \quad v_{\maximin} = \min_{a \in \mathcal{A}_{\min}} u(a_1, a)$$

So we have:

$$\begin{aligned} v_{\minimax} &= \max_{a \in \mathcal{A}_{\max}} u(a, a_2) \\ &\geq u(a_1, a_2) \\ &\geq \min_{a \in \mathcal{A}_{\min}} u(a_1, a) \\ &= v_{\maximin} \end{aligned}$$

The fundamental fact about zero sum games is that (subject to some regularity conditions on the game), going first is not in fact a disadvantage: in particular, $v_{\maximin} = v_{\minimax}$. The regularity conditions that are needed are exactly those that allow players to use online convex optimization algorithms to obtain diminishing regret to the best action in their action space.

Theorem 42 *Fix a zero-sum game such that:*

1. *The strategy sets $\mathcal{A}_{\min} \subseteq \mathbb{R}^n$ and $\mathcal{A}_{\max} \subseteq \mathbb{R}^m$ are closed, bounded, and convex,*
2. *The utility function $u : \mathcal{A}_{\max} \times \mathcal{A}_{\min} \rightarrow [0, B]$ satisfies:*
 - (a) *For all $a_2 \in \mathcal{A}_{\min}$, $u(\cdot, a_2)$ is concave and Lipschitz continuous in its first argument, and*
 - (b) *For all $a_1 \in \mathcal{A}_{\max}$, $u(a_1, \cdot)$ is convex and Lipschitz continuous in its second argument.*

Then:

$$\max_{a_1 \in \mathcal{A}_{\max}} \min_{a_2 \in \mathcal{A}_{\min}} u(a_1, a_2) = \min_{a_2 \in \mathcal{A}_{\min}} \max_{a_1 \in \mathcal{A}_{\max}} u(a_1, a_2)$$

Remark 7.4.1 *The simplest kind of zero sum games involve finite action spaces A_1, A_2 . Finite action spaces are not convex, and so to apply the minimax theorem, it is necessary to convexify them by letting players use probability distributions over their actions: $\mathcal{A}_{\max} = \Delta A_1$, $\mathcal{A}_{\min} = \Delta A_2$. When we do this, we extend the utility function u from the domain $A_1 \times A_2$ to the domain $\Delta A_1 \times \Delta A_2$ by defining for any pair $(p_1, p_2) \in \Delta A_1 \times \Delta A_2$, $u(p_1, p_2) = \mathbb{E}_{a_1 \sim p_1, a_2 \sim p_2} [u(a_1, a_2)]$. This function is linear in p_1 and p_2 (by linearity of expectation), and so convex and concave in each argument.*

Proof 65 (Proof of Theorem 42) We know from Lemma 7.4.1 that $v_{\minimax} \geq v_{\maximin}$. Suppose for point of contradiction that the inequality is strict, and let $\epsilon = v_{\minimax} - v_{\maximin} > 0$.

We now imagine repeated play of the game between Min and Max for T rounds, which generates a sequence of action pairs $\{(a^t, b^t)\}_{t=1}^T$. Since for every $a^t \in \mathcal{A}_{\max}$, $u(a^t, \cdot)$ is Lipschitz and convex in its second argument, we will let Min choose her action every day using an online convex optimization algorithm (like Multiplicative Weights, using the reduction from Theorem 75). We will let Max best respond to Min's action each day: $a^t \in Br_{\max}(b^t)$; we will then feed Min's online convex optimization algorithm the loss function $\ell^t(b^t) = u(a^t, b^t)$, which provides the feedback she needs to select b^{t+1} . Let us now analyze the cumulative utility of the game. We know two things: First, because Min's action space is convex and bounded, and the loss functions $\ell^t(b^t) = u(a^t, b^t)$ we feed to Min's learning algorithm are convex and Lipschitz, she has an $O(\sqrt{T})$ regret guarantee to the best fixed action in hindsight (the constants in the bound depend on things like the Lipschitz constant L and the diameter of the action space, so we elide them with Big- O notation):

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^T u(a^t, b^t) &\leq \min_{b \in \mathcal{A}_{\min}} \frac{1}{T} \sum_{t=1}^T u(a^t, b) + O\left(\frac{1}{\sqrt{T}}\right) \\ &\leq \min_{b \in \mathcal{A}_{\min}} u\left(\frac{1}{T} \sum_{t=1}^T a^t, b\right) + O\left(\frac{1}{\sqrt{T}}\right) \\ &\leq v_{\maximin} + O\left(\frac{1}{\sqrt{T}}\right) \end{aligned}$$

Here, in the second inequality we use the fact that for every $b \in \mathcal{A}_{\min}$, $u(\cdot, b)$ is concave in its first argument, and apply Jensen's inequality. We also use the fact that \mathcal{A}_{\max} is a convex set, which implies that $\left(\frac{1}{T} \sum_{t=1}^T a^t\right) \in \mathcal{A}_{\max}$.

On the other hand, since for every t , $a^t \in Br_{\max}(b^t)$, we also have that:

$$u(a^t, b^t) \geq v_{\minimax}$$

Combining these two bounds gives:

$$v_{\minimax} \leq \frac{1}{T} \sum_{t=1}^T u(a^t, b^t) \leq v_{\maximin} + O\left(\frac{1}{\sqrt{T}}\right)$$

Thus by taking T to be sufficiently large we can obtain:

$$v_{\minimax} < v_{\maximin} + \epsilon$$

which contradicts our initial assumption. Thus we must have that $v_{\minimax} = v_{\maximin}$, proving the theorem.

The Minimax theorem (Theorem 42) allows us to speak of the *value* v of a zero sum game — we don't have to specify whether we mean the minimax or maximin value, as they are the same.

Definition 49 *In any zero-sum game for which the conditions of the minimax Theorem hold, we define the value of the game as the unique value $v \in \mathbb{R}$ such that:*

$$v_{\text{minimax}} = v_{\text{maximin}} = v$$

A simple consequence of the minimax theorem is that Nash equilibria in zero sum games are pairs of minimax/maximin strategies.

Lemma 7.4.2 *Fix a zero sum game in which the Minimax theorem holds. A pair of strategies $(a_1, a_2) \in \mathcal{A}_{\text{max}} \times \mathcal{A}_{\text{min}}$ is a Nash equilibrium if and only if a_1 is a maximin strategy and a_2 is a minimax strategy.*

Proof 66 *Let v be the value of the zero sum game (whose existence is guaranteed by the minimax theorem). First suppose that a_1 and a_2 are maximin and minimax strategies respectively. We must have that $u(a_1, a_2) \geq v$, and $u(a_1, a_2) \leq v$, and so $u(a_1, a_2) = v$. Thus we have that $a_1 \in Br_{\text{max}}(a_2)$ (since a_2 is a minimax strategy), and $a_2 \in Br_{\text{min}}(a_1)$ (since a_1 is a maximin strategy), implying that (a_1, a_2) are a Nash equilibrium.*

Next, suppose that (a_1, a_2) are a Nash equilibrium. Since $a_1 \in Br_{\text{max}}(a_2)$, and since there is a maximin strategy in \mathcal{A}_{max} , we must have that $u(a_1, a_2) \geq v$. Similarly, since $a_2 \in Br_{\text{min}}(a_1)$, we must have that $u(a_1, a_2) \leq v$ and so $u(a_1, a_2) = v$. Since a_2 is a best response to a_1 and vice versa, we have that a_1 and a_2 are maximin and minimax strategies respectively.

7.5 Passing Arbitrary Sequential Statistical Tests

In this Section we will apply the minimax theorem to show a very general result about forecasting in online adversarial settings, that will ultimately amount to this: there exist sequential learning algorithms that—in isolation—are able to pass *any statistical test* that is guaranteed to pass an informed forecaster. Here, an informed forecaster is one who knows, and correctly communicates every day, the distribution of the outcome. On the other hand, the learning algorithms we will develop are obligated to pass the same test even when facing an adversarial sequence of outcomes, and so definitionally cannot know the true distribution of outcomes.

The very general result we'll prove in this section was originally interpreted as a very negative result — showing that no empirical test could be used to distinguish an informed from an uninformed forecaster. But we will argue that

it is in fact a striking positive result, showing that sequential learning algorithms can accomplish a wide variety of surprising feats, even when operating in an adversarial environment. By appropriately instantiating the statistical test, we'll see that sequential learning algorithms can make predictions that are (multi)-calibrated, will be as accurate as any of a fixed set of benchmark prediction methods not just overall, but conditional on any fixed set of events, and more — in general, sequential learning algorithms can be made to satisfy any condition that will always be satisfied by an informed predictor.

The cost of the generality of the theorem we prove in this section is that it will be non-constructive: we will argue for the *existence* of algorithms with remarkable properties, but will not actually give any concrete algorithms. But in the next chapter, we'll see how to make our algorithms constructive and efficient for statistical tests that can be described by testing a set of linear inequalities, which will include both multicalibration and conditional “regret” bounds with respect to any fixed set of benchmark policies.

7.5.1 A Model

We'll think about an interaction between a *forecaster* and *nature* that takes place over a series of rounds $t = 1, \dots, T$. In each round t , the forecaster will observe context $x_t \in \mathcal{X}$ and produce a forecast $p_t \in \mathcal{A}$ and nature will produce an outcome $y_t \in \mathcal{B}$. To make our lives simple, we'll assume that \mathcal{X} , \mathcal{A} and \mathcal{B} are finite sets, but this can be extended to convex, compact sets. The sequence of forecasts and outcomes will accumulate into a transcript, $\pi^T = ((x_1, p_1, y_1), \dots, (x_T, p_T, y_T))$. We'll write $\Pi^T = (\mathcal{A} \times \mathcal{B})^T$ for the domain of all possible transcripts of length T . Nature here can be modelled as an algorithm mapping $\Pi^* \rightarrow \mathcal{X} \times \Delta\mathcal{B}$.

We will take the perspective of a tester who aims to distinguish between two scenarios: one in which the forecaster is *informed*, and one in which the forecaster is *uninformed*. The informed forecaster first learns the distribution $\mathcal{D}^t \in \Delta\mathcal{B}$ that nature will play at every round, and then correctly forecasts some property Γ of the forecaster's distribution, $p_t = \Gamma(\mathcal{D}^t)$. For example, if the outcome space is binary $\mathcal{B} = \{0, 1\}$ (indicating e.g. whether it will rain on day t or not), the forecaster might report the mean of the distribution $p_t = \mathbb{E}_{\mathcal{D}^t}[y_t]$. This scenario is shown in Algorithm 26.

Algorithm 26 An Informed Forecaster

for $t = 1$ to T **do**

Nature chooses $x^t \in \mathcal{X}$ and reveals it to the forecaster.

Nature chooses a distribution $\mathcal{D}^t \in \Delta\mathcal{B}$, and reports it to the forecaster.

The forecaster reports $p_t = \Gamma(\mathcal{D}^t)$

The outcome $y_t \sim \mathcal{D}^t$ is sampled and realized.

The final transcript $\pi^T = ((x_t, p_1, y_1), \dots, (x_T, p_T, y_T))$ is output.

The uninformed forecaster on the other hand does not have the benefit of observing \mathcal{D}^t before they make their forecast, and instead must make algorithmic forecasts at each day t as a function only of the transcript observed π^{t-1} so far and the current context x^t . We write $\text{Alg}(\pi^{t-1}, x^t)$ to denote the distribution they sample their forecast from at round t . The scenario is shown in Algorithm 27.

Algorithm 27 An Uninformed Forecaster

for $t = 1$ to T **do**

Nature chooses $x^t \in \mathcal{X}$ and reveals it to the forecaster.

Nature chooses a distribution $\mathcal{D}^t \in \Delta\mathcal{B}$, but keeps it secret from the forecaster.

The forecaster predicts p_t sampled from $\text{Alg}(\pi^{t-1}, x^t)$.

The outcome $y_t \sim \mathcal{D}^t$ is sampled and realized.

The final transcript $\pi^T = ((x_t, p_1, y_1), \dots, (x_T, p_T, y_T))$ is output.

A test $\mathcal{T} : \Pi^T \rightarrow \{0, 1\}$ is an arbitrary function whose goal is to distinguish, based only on the final transcript π^T , whether a forecaster was operating in the informed scenario (Algorithm 26) or the uninformed scenario (Algorithm 27). We'll use the convention that $\mathcal{T}(\pi^T) = 1$ means that the test determines that the forecaster was informed, and $\mathcal{T}(\pi^T) = 0$ means that the test determines that the forecaster was uninformed. Since nature may be stochastic, \mathcal{T} is a statistical test, and so we cannot expect it to be perfect. However we'll ask that our tests have a low false negative rate (i.e. they should rarely assert that an informed forecaster is uninformed):

Definition 50 A test \mathcal{T} has a false negative rate of at most δ if for every algorithm for nature, the distribution on transcripts π^T generated by an informed forecaster (Algorithm 26) satisfies:

$$\mathbb{E}_{\pi^T}[\mathcal{T}(\pi^T)] \geq 1 - \delta$$

One (trivial) way to give a test that has a false negative rate of δ is to ignore the transcript and simply output $\mathcal{T}(\pi^T) = 1$ with probability $1 - \delta$. Of course, a test like this will have a *false positive rate* of $1 - \delta$ — i.e. this is the rate at which it will incorrectly assert that an uninformed forecaster is informed. To have an informative test, we'd like that the false positive rate be smaller than $1 -$ the false negative rate.

Definition 51 A test \mathcal{T} has a false positive rate of at least $1 - \delta$ if there exists an algorithm Alg for the forecaster such that for every algorithm for nature, the distribution on transcripts π^T generated by the uninformed forecaster (Algorithm 27) satisfies:

$$\mathbb{E}_{\pi^T}[\mathcal{T}(\pi^T)] \geq 1 - \delta$$

7.5.2 No Non-Trivial Statistical Tests

The main result we will prove is that for every δ , any test \mathcal{T} that has a false negative rate of at most δ must also have a false positive rate of at least $1 - \delta$ — i.e. the test is not informative. We'll prove this via a straightforward application of the Minimax theorem, that we proved as Theorem 42. This will make the proof non-constructive, but is a good demonstration of the deceptive power of the minimax theorem. The result will follow from analyzing the following two player zero sum game:

Definition 52 *The testing game is defined by a time horizon T and a statistical test $\mathcal{T} : \Pi^T \rightarrow \{0, 1\}$*

1. *The set of pure strategies for the forecaster is the set of all deterministic forecasting rules $\{f : \Pi^{<T} \times \mathcal{X} \rightarrow \mathcal{A}\}$. A mixed strategy defines a randomized forecasting rule $f : \Pi^{<T} \times \mathcal{X} \rightarrow \Delta\mathcal{A}$.*

2. *The set of pure strategies for nature is the set of all deterministic rules $\{g : \Pi^{<T} \rightarrow \mathcal{B}\}$ mapping transcripts to outcomes. A mixed strategy defines a distribution over outcomes for every transcript: $g : \Pi^{<T} \rightarrow \Delta\mathcal{B}$.*

3. *Fixing a pure strategy f, g for each player fixes a transcript $\pi^T(f, g)$. Similarly, fixing a mixed strategy for both the forecaster and nature defines a distribution over transcripts π^T that we will write as $\mathcal{D}(f, g) \in \Delta\Pi^T$, which is simply the distribution on transcripts $\pi^T(f, g)$ that results from sampling f and g independently from their corresponding distributions. The utility function that the forecaster wishes to maximize and that nature wishes to minimize is:*

$$u(f, g) = \mathbb{E}_{\pi^T \sim \mathcal{D}(f, g)} [\mathcal{T}(\pi^T)]$$

We can verify that this game satisfies the conditions of the minimax theorem (Theorem 42). In particular, both players have finite sets of pure strategies, and so the simplices over them are finite dimensional, closed, bounded, convex sets. The objective function is linear in the mixed strategies of both players by the linearity of expectation. With this construction in mind, we are now ready to prove the main result of this chapter; for every δ and for every test \mathcal{T} that has false negative rate at most δ , there exists a learning algorithm for the forecaster that witnesses that the test also has false positive rate at least $1 - \delta$. Thus the tradeoff between false negative and false positive rates even for the best test is no better than the trivial diagonal, which can be achieved by “tests” that ignore the transcript and output $\mathcal{T} = 1$ with probability $1 - \delta$.

Theorem 43 *Fix any test \mathcal{T} that has a false negative rate at most δ : i.e. any test \mathcal{T} such that for every algorithm for nature, the distribution over*

transcripts generated by an informed forecaster (Algorithm 26) satisfies:

$$\mathbb{E}_{\pi^T}(\mathcal{T}(\pi^T)) \geq 1 - \delta$$

Then \mathcal{T} must also have false positive rate at least $1 - \delta$. In other words, there exists an algorithm Alg for the forecaster such that for every algorithm for nature, the distribution on transcripts generated by an uninformed forecaster (Algorithm 27) also satisfies:

$$\mathbb{E}_{\pi^T}(\mathcal{T}(\pi^T)) \geq 1 - \delta$$

Proof 67 An algorithm for the forecaster can be viewed as a (mixed) strategy f in the zero-sum “testing game” defined in Definition 52. What we need to show is that there exists a mixed strategy f^* such that:

$$\min_g \mathbb{E}_{\pi^T \sim \mathcal{D}(f^*, g)}[\mathcal{T}(\pi^T)] \geq 1 - \delta$$

or equivalently that:

$$\max_f \min_g \mathbb{E}_{\pi^T \sim \mathcal{D}(f, g)}[\mathcal{T}(\pi^T)] \geq 1 - \delta$$

By the minimax theorem (Theorem 42) we can equivalently lower bound:

$$\min_g \max_f \mathbb{E}_{\pi^T \sim \mathcal{D}(f, g)}[\mathcal{T}(\pi^T)]$$

This is much easier to analyze. Once nature’s strategy g is fixed, we can construct the following strategy f for the forecaster. At round t , the transcript $\pi^{\leq t-1}$ is fixed, and so the nature’s distribution $\mathcal{D}^t = g(\pi^{\leq t-1})$ is also fixed. The forecaster can simply forecast $p_t = f(\pi^{\leq t-1}) = \Gamma(\mathcal{D}^t)$. But note that with this strategy and order of play, the forecaster is acting as an informed forecaster exactly as they would in Algorithm 26. By hypothesis, since the test has a false negative rate of at most δ , we must therefore have that

$$\min_g \max_f \mathbb{E}_{\pi^T \sim \mathcal{D}(f, g)}[\mathcal{T}(\pi^T)] \geq 1 - \delta$$

The minimax theorem therefore lets us conclude that

$$\max_f \min_g \mathbb{E}_{\pi^T \sim \mathcal{D}(f, g)}[\mathcal{T}(\pi^T)] \geq 1 - \delta$$

i.e. there is a strategy f^* that passes test \mathcal{T} with probability at least $1 - \delta$ against any adversary.

7.5.3 Implications

Lets briefly discuss the implications of Theorem 43. First and foremost, it tells us that any property that a transcript π^T is *guaranteed* to satisfy with high probability in the informed forecasting scenario can also be guaranteed to be satisfied in the adversarial forecasting scenario. So, for example, we know that not only is it possible to satisfy (multi)-calibration asymptotically in the online adversarial setting for any fixed set of groups, but we can obtain calibration error *rates* equal to those that we could obtain in the informed setting. How come? We simply design a test \mathcal{T} that on input transcript π^T checks the calibration error on each group in a fixed set \mathcal{G} and verifies that it is below some pre-defined threshold. Similarly, we can obtain multi-group accuracy in an adversarial sequential prediction scenario with respect to any loss function ℓ , any set of benchmark policies \mathcal{H} , and any set of groups \mathcal{G} — and we can do so at the same rates that would be obtainable in the informed forecasting scenario. Of course the argument in this section gives us no indication of what the forecaster’s learning algorithm should be — and depending on what the test \mathcal{T} is, no guarantee that there is an efficient algorithm. But we’ll see in the next chapter that we can derive efficient algorithms for passing any test \mathcal{T} that amounts to bounding a fixed set of *linear* quantities — i.e. quantities that are derived by summing up over the rounds t of the transcript. This will capture both multicalibration and multi-group accuracy.



8

Efficient Sequential Adversarial Learning with Conditional Guarantees

CONTENTS

8.1	Computing Minimax Strategies	137
8.2	A General Framework for Multiobjective Sequential Learning ..	140
8.3	Conditional Regret Guarantees	146
8.3.1	Action Independent Subsequences	149
8.3.1.1	Adaptive Regret	151
8.3.1.2	Group-wise Regret	152
8.3.2	General Subsequences	153
8.3.2.1	Swap Regret	155
8.3.2.2	Mixing and Matching Guarantees	157
8.4	Sequential Adversarial Multicalibration	157
8.4.1	A Bucketed Calibration Definition	158
8.4.2	Bounding Sequential Multicalibration Error	159
	Bibliographic Notes and Further Reading	163

In Section 7.5 we used the minimax theorem to prove a very general result, that (we argued) implied the existence of sequential forecasting algorithms that were guaranteed to be able to satisfy multicalibration, multi-group accuracy, and many other things. But the argument was non-constructive, and we never argued even for the existence of computationally efficient algorithms. In this Chapter we'll develop machinery to give computationally efficient algorithms for a host of multi-objective optimization problems that will capture both multicalibration and multi-group accuracy, along with a host of other objectives that can be specified as satisfying a fixed set of linear constraints. We'll build up our tool-kit, starting by giving constructive algorithms to compute approximate minimax strategies, then using these algorithms to efficiently solve multi-objective optimization problems, and then finally instantiating the algorithms we've derived with particular learning goals with conditional guarantees.

8.1 Computing Minimax Strategies

So far we've used the existence of online convex optimization algorithms to prove the minimax theorem, but sometimes it will be useful for us to actually be able to compute approximate minimax (and maximin) equilibria in particular zero sum games. We can use online convex optimization algorithms to do this as well. There are a few variants, each of which is sometimes useful. First we define an approximate minimax and maximin equilibrium:

Definition 53 Fix a zero sum game $(\mathcal{A}_{max}, \mathcal{A}_{min}, u)$. An ϵ -approximate maximin strategy is an action $a \in \mathcal{A}_{max}$ such that for all $b \in \mathcal{A}_{min}$:

$$u(a, b) \geq \max_{a' \in \mathcal{A}_{max}} \min_{b' \in \mathcal{A}_{min}} u(a', b') - \epsilon$$

Similarly, an ϵ -approximate minimax strategy is an action $b \in \mathcal{A}_{min}$ such that for all $a \in \mathcal{A}_{max}$:

$$u(a, b) \leq \min_{b' \in \mathcal{A}_{min}} \max_{a' \in \mathcal{A}_{max}} u(a', b') + \epsilon$$

Our algorithms will be of two types, both of which simulate repeated play of the zero sum game over some number of rounds T . Either we will play two no-regret/online convex optimization algorithms against one another, or we will have one player play using an online convex optimization algorithm, and the other player “best respond” at each round. In fact, we don't necessarily need the “best response” player to play a best response — its enough if they play a strategy that achieves the value of the game at each round (i.e. they do not need to be able to exploit their opponent if their opponent is playing badly).

Definition 54 Fix a zero sum game satisfying the conditions of the minimax theorem, and let v be the value of the game. A value oracle for Max is a mapping $Val : \mathcal{A}_{min} \rightarrow \mathcal{A}_{max}$ such that for all $b \in \mathcal{A}_{min}$, $Val(b) = a$ such that:

$$u(a, b) \geq v$$

Similarly a value oracle for Min is a mapping $Val : \mathcal{A}_{max} \rightarrow \mathcal{A}_{min}$ such that for all $a \in \mathcal{A}_{max}$, $Val(a) = b$ such that:

$$u(a, b) \leq v$$

Remark 8.1.1 Note that computing an actual best response — $Val(b) = Br_{max}(b)$ for Max, and $Val(a) = Br_{min}(a)$ for Min gives a value oracle, but it might sometimes be easier to implement a value oracle than to compute the best response in a game.

Algorithm 28 Computing a Minimax Equilibrium: Value Oracle vs. No Regret

Given: A zero sum game $(\mathcal{A}_{\max}, \mathcal{A}_{\min}, u)$ satisfying the conditions of the minimax theorem, a Value oracle $Val : \mathcal{A}_{\min} \rightarrow \mathcal{A}_{\max}$ for Max, an online convex optimization algorithm `OnlineConvex` operating over action space \mathcal{A}_{\min} and loss space $\{\ell = u(a, \cdot)\}_{a \in \mathcal{A}_{\max}}$ that promises regret $R(T)$ to every action $b \in \mathcal{A}_{\min}$ after T rounds, and an approximation parameter ϵ .

Let T be such that $R(T)/T \leq \epsilon$

for $t = 1$ to T **do**

Get action b^t from `OnlineConvex`,

Let $a^t = Val(b^t)$

Feed loss $\ell^t = u(a^t, \cdot)$ to `OnlineConvex`.

Let $\bar{a} = \frac{1}{T} \sum_{t=1}^T a^t$

Return \bar{a}

Theorem 44 *The action \bar{a} output by Algorithm 28 is an ϵ -approximate Maximin strategy.*

Proof 68 *By the regret bound of `OnlineConvex` we know for every $b \in \mathcal{A}_{\min}$:*

$$\begin{aligned} \epsilon &\geq \frac{1}{T} \sum_{t=1}^T (u(a^t, b^t) - u(a^t, b)) \\ &\geq \frac{1}{T} \left(\sum_{t=1}^T u(a^t, b^t) \right) - u(\bar{a}, b) \end{aligned}$$

where the last inequality follows from the fact that $u(\cdot, b^*)$ is convex in its first argument and Jensen's inequality.

Rearranging we have that for all $b \in \mathcal{A}_{\min}$:

$$\begin{aligned} u(\bar{a}, b) &\geq \frac{1}{T} \left(\sum_{t=1}^T u(a^t, b^t) \right) - \epsilon \\ &\geq v - \epsilon \end{aligned}$$

where the last inequality follows from the fact that a^t is selected by a value oracle $Val(b^t)$, and so for all t , $u(a^t, b^t) \geq v$. Thus \bar{a} is an ϵ -approximate Maximin strategy.

Remark 8.1.2 *Observe that we didn't really need an exact value oracle. If the guarantee was that for $a^t = Val(b^t)$, $u(a^t, b^t) \geq v - \epsilon$, then we would have found a 2ϵ -approximate Maximin strategy, so an approximate Value oracle is enough.*

Note also that the situation is symmetric: If we want to compute an ϵ -approximate Minimax strategy for Min, we can reverse the role of the players and have Max play a no regret algorithm and have Min play according to an (approximate) value oracle.

There is another general way to compute Minimax and Maximin strategies in a zero-sum game: have *both* players play the game for T rounds so that they have at most ϵ average regret. They could do this by both playing according to online convex optimization algorithms, or we could play an online convex optimization algorithm against a best response (not just a value) oracle.

Theorem 45 *Fix a zero sum game $(\mathcal{A}_{max}, \mathcal{A}_{min}, u)$ satisfying the conditions of the minimax theorem, and let $\{(a^t, b^t)\}_{t=1}^T$ be a sequence of action pairs that mutually have ϵ average regret with respect to each other — i.e. such that:*

$$\frac{1}{T} \min_{b \in \mathcal{A}_{min}} \sum_{t=1}^T u(a^t, b) + \epsilon \geq \frac{1}{T} \sum_{t=1}^T u(a^t, b^t) \geq \frac{1}{T} \max_{a \in \mathcal{A}_{max}} \sum_{t=1}^T u(a, b^t) - \epsilon$$

Let

$$\bar{a} = \frac{1}{T} \sum_{t=1}^T a^t \quad \bar{b} = \frac{1}{T} \sum_{t=1}^T b^t$$

Then \bar{a} is a 2ϵ -approximate maximin strategy and \bar{b} is a 2ϵ -approximate minimax strategy.

Proof 69 *We prove the claim about \bar{a} : The claim about \bar{b} follows exactly symmetrically.*

From the right hand side of the no regret guarantee together with the convexity of u in its second argument (and Jensen's inequality), we have that:

$$\begin{aligned} \frac{1}{T} \sum_{t=1}^T u(a^t, b^t) &\geq \max_{a \in \mathcal{A}_{max}} \frac{1}{T} \sum_{t=1}^T u(a, b^t) - \epsilon \\ &\geq \max_{a \in \mathcal{A}_{max}} u(a, \bar{b}) - \epsilon \\ &\geq \min_{b \in \mathcal{A}_{min}} \max_{a \in \mathcal{A}_{max}} u(a, b) - \epsilon \\ &= v - \epsilon \end{aligned}$$

From the left hand side of the no-regret guarantee together with the concavity of u in its first argument (and Jensen's inequality) we have that for every $b \in \mathcal{A}_{min}$:

$$\begin{aligned} u(\bar{a}, b) &\geq \frac{1}{T} \sum_{t=1}^T u(a^t, b) \\ &\geq \frac{1}{T} \sum_{t=1}^T u(a^t, b^t) - \epsilon \\ &\geq v - 2\epsilon \end{aligned}$$

Which establishes that \bar{a} is a 2ϵ -approximate maximin strategy.

8.2 A General Framework for Multiobjective Sequential Learning

So far we've viewed sequential learning as having a single goal: obtaining a diminishing regret guarantee, always as computed over the entire sequence. Similarly we have studied zero sum games in which there is a single, one dimensional objective function that one player wants to maximize and the other wants to minimize.

In this section, things will get more complicated: in a sequential learning setting, there might be more than one objective that we simultaneously want to control. We'll show how to use the tools we have developed for online convex optimization to solve this problem. We'll give a number of applications of this technique that will connect back to our goal of learning (in adversarial settings) with various conditional guarantees — each of which will translate into a different objective in the framework we are about to derive

We consider a sequential decision making setting in which the learner has d distinct cost functions, each of which they would like to keep as small as possible over a sequential interaction. We define a very general setting, in which the action space for the learner and the adversary, as well as the nature of each of the loss functions can change at each round. We will not always need this generality, but when we do it will come in handy. In the general setting, the adversary gets to choose the action spaces at every round, as well as the “loss increment functions” for each of the d loss objectives. These can differ at each round. The learner then selects an action, the adversary responds, and loss accrues to each loss objective according to the loss increment function for that objective.

Definition 55 (The Multiobjective Optimization Game) *The d -objective optimization game proceeds in rounds $t = 1, \dots, T$. At each round t :*

1. *The adversary selects the environment for round t which comprises:*

- (a) *Closed convex, finite dimensional action spaces $\mathcal{A}^t, \mathcal{B}^t \subset \mathbb{R}^m$ for the learner and adversary respectively, and*
- (b) *A continuous loss increment function ℓ_i^t for each of the d losses $i \in [d]$. Each $\ell_i^t : \mathcal{A}^t \times \mathcal{B}^t \rightarrow [-1, 1]$ is convex in its first argument and concave in its second argument.*

2. *The Learner selects an action $a^t \in \mathcal{A}^t$ and reveals it to the adversary.*

3. *The Adversary selects an action $b^t \in \mathcal{B}^t$.*

4. *The Learner suffers and observes loss $\ell_i^t(a^t, b^t)$, accruing to each of the loss objectives $i \in [d]$.*

At the end of T rounds, the cumulative loss that the learner suffers for each objective $i \in [d]$ is: $L_i^T = \sum_{t=1}^T \ell_i^t(a^t, b^t)$.

The goal of the learner in interacting within the multiobjective optimization game is to upper bound the maximum accumulated loss across all objectives: i.e. she wants to minimize $\max_i L_i^T$. What is a reasonable benchmark? Here we introduce the “Adversary Moves First” value for the environment at round t :

Definition 56 Fix the environment at round t , defined by $(\mathcal{A}^t, \mathcal{B}^t, \{\ell_i^t\}_{i=1}^d)$. The adversary moves first value at round t is:

$$v_A^t = \max_{b \in \mathcal{B}^t} \min_{a \in \mathcal{A}^t} \max_{i \in [d]} \ell_i^t(a, b)$$

Informally, the adversary moves first value is the smallest upper bound on the d loss increments at round t that the learner could guarantee, if she had the advantage of first observing the adversary’s chosen action b^t before deciding on her own best response a^t . Note that the order of play here is reversed compared to how it actually proceeds in the multiobjective optimization game. In the actual interaction, the learner moves first and then best responds. It is tempting to appeal to the minimax theorem here to assert that the learner can do just as well in this play order, but this turns out not to be the case. Although the loss increments in each coordinate $\ell_i^t(a, b)$ are indeed convex/concave, the maximum over coordinates $\max_{i \in [d]} \ell_i^t(a, b)$ does not preserve concavity, and hence the conditions of the minimax theorem are not satisfied. Indeed, in this setting, the minimax theorem simply doesn’t hold:

Example 4 Suppose the action spaces for both the learner and the adversary are the d -dimensional simplex: $\mathcal{A} = \mathcal{B} = \Delta[d]$. Let the loss increment in coordinate i be $\ell_i(a, b) = (b_i - a_i)$, the difference between the weight that the adversary and the learner place on coordinate i . If the adversary moves first and plays b , the learner can best respond and play $a = b$, guaranteeing that the loss increment in every coordinate is equal to 0: thus the adversary moves first value $v_A = 0$ for this environment.

On the other hand, suppose the learner moves first: for every vector $a \in \mathcal{A}$ that she might choose, there is a coordinate i such that $a_i \leq \frac{1}{d}$. The adversary can best respond by playing a vector b that places all of its weight on this coordinate i : $b_i = 1$. Hence $\ell_i(a, b) \geq 1 - \frac{1}{d}$. Thus there is a large gap between the adversary moves first value for this environment and the “learner moves first” value — the minimax theorem does not hold.

The above example shows that trying to obtain maximum loss equal to the adversary-moves-first value of the game in a 1-round interaction is impossible. Nevertheless, we will be able to *approach* the average of the adversary-moves-first values of the environments over a larger sequence of T interactions.

Definition 57 (Adversary Moves First Regret) Fix a transcript of interaction in the multiobjective optimization game $\pi^T = \{(\mathcal{A}^t, \mathcal{B}^t, \{\ell_i^t\}_{i=1}^d), a^t, b^t\}_{t=1}^T$. The adversary moves first regret of this transcript is:

$$\text{Reg}_{AMF}(\pi^T) = \max_{i \in [d]} L_i^T - \sum_{t=1}^T v_A^t = \max_{i \in [d]} \left(\sum_{t=1}^T (\ell_i^t(a^t, b^t) - v_A^t) \right)$$

Our goal will be to design algorithms that guarantee that the Adversary Moves First regret grows sublinearly with T . In most of our applications, we will define the loss increments so that the adversary moves first value of each environment is 0: $v_A^t = 0$ for all t . In this case, the adversary moves first regret is simply the maximum accumulated loss in any coordinate: $\max_i L_i^T$.

We give the algorithm in Algorithm 29, which is a reduction from the problem of guaranteeing AMF regret bounded by $R(T)$ to the problem (that we have already solved) of selecting a distribution over d actions at every round to obtain cost that is as large as the cumulative cost of the highest cost action in hindsight, up to a regret bound of $R(T)$. (i.e. the online linear optimization problem over the probability simplex $\Delta[d]$). We've already seen how to solve this problem using several algorithms (e.g. multiplicative weights, online gradient descent, exponential weights, etc.). Note that we have generally framed online linear optimization as the problem of *minimizing* cumulative cost, whereas here it is more convenient to use an online linear optimization algorithm that *maximizes* cumulative cost — but we can obtain an online linear maximization algorithm from an online linear minimization algorithm simply by negating the cost vectors.

The intuition for the reduction is simple. We run a sequential linear optimization algorithm over distributions on a set of d actions, with each action corresponding to one of the loss objectives in the Multiobjective Optimization Game. From its regret bound, the sequential linear optimization algorithm is guaranteed to experience cumulative cost that is nearly as large as that of the action with largest cumulative cost in hindsight — which corresponds to the cumulative loss of the maximum of the d loss objectives in the Multiobjective Optimization Game. This is exactly the quantity that we wish to control. Thus if we want to upper bound the cumulative loss of the highest loss objective in the Multiobjective Optimization Game, it suffices to upper bound the cumulative loss of the sequential linear optimization algorithm. Of course we don't know what this is going to be, because we don't know what action the adversary will choose at each round: but what we can try to do is upper bound the loss of the sequential linear optimization algorithm in the worst case over the adversary's action. That is exactly what the reduction does: it defines a zero sum game in which the objective is the loss that the sequential linear optimization algorithm will experience, and plays a minimax strategy in that game. Thus the cumulative loss of the online linear optimization algorithm, and hence the cumulative loss of the maximum coordinate of the multiobjective optimization game, is upper bounded by the sum of the values

of the zero-sum games we have defined along the way, which turn out to be exactly the Adversary Moves First values v_A^t of each round of the multiobjective optimization game.

Algorithm 29 Reduction from AMF to Simple Regret

Given A sequential linear maximization algorithm Alg operating over action space $\Delta[d]$ and accepting cost vectors in $[-1, 1]^d$.

for $t = 1$ to T **do**

Obtain distribution p^t from Alg .

Define a zero-sum game in which the minimization player's action are \mathcal{A}^t , the maximization players actions are \mathcal{B}^t , and the utility function is:

$$u^t(a, b) = \sum_{i=1}^d p_i^t \ell_i^t(a, b)$$

Compute a minimax equilibrium strategy of this game a^t for the minimization player and select action a^t

Observe the adversary's action b^t and report cost vector c^t to Alg , defined such that in each coordinate $i \in [d]$:

$$c_i^t = \ell_i^t(a^t, b^t)$$

The result is that if we instantiate Algorithm 29 with an online linear optimization algorithm that has regret bound $R(T)$, then we will obtain an AMF regret bound of $R(T)$ as well in the multi-objective optimization problem!

Theorem 46 *Suppose Algorithm 29 is instantiated with a sequential linear optimization algorithm operating over $\Delta[d]$ that has the guarantee that for any sequence of losses of length T bounded in $[-1, 1]$, it generates a transcript π^T with regret at most $R(T)$:*

$$\max_{i \in [d]} \text{Reg}(\pi^T, i) \leq R(T)$$

Then in any d -objective optimization game, after T rounds, Algorithm 29 obtains AMF regret at most $R(T)$:

$$\max_{i \in [d]} \left(\sum_{t=1}^T (\ell_i^t(a^t, b^t) - v_A^t) \right) \leq R(T)$$

Proof 70 *From the regret bound of the online linear maximization algorithm, we know that for all $j \in [d]$:*

$$\begin{aligned} \sum_{t=1}^T (c_j^t - \langle p^t, c^t \rangle) &= \sum_{t=1}^T \left(\ell_j^t(a^t, b^t) - \sum_{i=1}^d p_i^t \ell_i^t(a^t, b^t) \right) \\ &\leq R(T) \end{aligned}$$

Or, rearranging:

$$\max_{j \in [d]} L_j^T = \max_{j \in [d]} \sum_{t=1}^T \ell_j^t(a^t, b^t) \leq \sum_{t=1}^T \sum_{i=1}^d p_i^t \ell_i^t(a^t, b^t) + R(T)$$

Algorithm 29 selects a^t at every round so that:

$$a^t \in \arg \min_{a \in \mathcal{A}^t} \max_{b \in \mathcal{B}^t} \sum_{i=1}^d p_i^t \ell_i^t(a, b)$$

Because for each coordinate i , we know that $\ell_i^t(a, b)$ is convex in a and concave in b , and linear combinations of convex/concave functions are convex/concave, the utility function $u^t(a, b) = \sum_{i=1}^d p_i^t \ell_i^t(a, b)$ satisfies the conditions of the minimax theorem (Theorem 42), and so we know that:

$$\begin{aligned} \min_{a \in \mathcal{A}^t} \max_{b \in \mathcal{B}^t} \sum_{i=1}^d p_i^t \ell_i^t(a, b) &= \max_{b \in \mathcal{B}^t} \min_{a \in \mathcal{A}^t} \sum_{i=1}^d p_i^t \ell_i^t(a, b) \\ &\leq \max_{b \in \mathcal{B}^t} \min_{a \in \mathcal{A}^t} \max_{i \in [d]} \ell_i^t(a, b) \\ &= v_A^t \end{aligned}$$

Hence we know that at every round t ,

$$\sum_{i=1}^d p_i^t \ell_i^t(a^t, b^t) \leq v_A^t$$

Finally, this lets us conclude that for every loss coordinate j :

$$\begin{aligned} L_j^T &= \max_{j \in [d]} \sum_{t=1}^T \ell_j^t(a^t, b^t) \\ &\leq \sum_{t=1}^T \sum_{i=1}^d p_i^t \ell_i^t(a^t, b^t) + R(T) \\ &\leq \sum_{t=1}^T v_A^t + R(T) \end{aligned}$$

Or in other words, the AMF regret is bounded by $R(T)$.

Finally, we can instantiate Algorithm 29 with a particular online linear maximization algorithm over the simplex $\Delta[d]$ — the exponential weights algorithm we derived in Algorithm 43. The exponential weights algorithm has a particularly simple form for the weights p_i^t , and a good concrete regret bound. We give this instantiation below in Algorithm 30.

Algorithm 30 Multiobjective Optimization with Exponential Weights**for** $t = 1$ to T **do**Define the distribution $p^t \in \Delta[d]$ as:

$$p_i^t = \frac{\exp\left(\frac{\eta}{2} \sum_{t'=1}^{t-1} \ell_i^{t'}(a^{t'}, b^{t'})\right)}{\sum_{j=1}^d \exp\left(\frac{\eta}{2} \sum_{t'=1}^{t-1} \ell_j^{t'}(a^{t'}, b^{t'})\right)}$$

Define a zero-sum game in which the minimization player's action are \mathcal{A}^t , the maximization players actions are \mathcal{B}^t , and the utility function is:

$$u^t(a, b) = \sum_{i=1}^d p_i^t \ell_i^t(a, b)$$

Compute a minimax equilibrium strategy of this game a^t for the minimization player and select action a^t **Theorem 47** *In any d -objective optimization game, after T rounds, Algorithm 30 obtains AMF regret at most $4\sqrt{T \ln d}$:*

$$\max_{i \in [d]} \left(\sum_{t=1}^T (\ell_i^t(a^t, b^t) - v_A^t) \right) + 4\sqrt{T \ln d}$$

Proof 71 *We simply instantiate Theorem 46 with the regret bound proven for the exponential weights algorithm in Theorem 77. Note that there we stated the regret bound for Exponential Weights when the costs were scaled in $[0, 1]$ and the algorithm was minimizing cost. In our case, the costs are scaled in $[-1, 1]$ and we are maximizing cost. We simply apply the cost transformation described in Remark 7.2.2 and negate the cost vectors to obtain the given algorithm and regret bound.*

8.3 Conditional Regret Guarantees

As our first application of online multiobjective optimization, we will design algorithms for choosing amongst k actions that have diminishing regret to the best action in hindsight — not just on average over the whole sequence, but also conditional on the round t being contained in various subsequences, which may intersect (i.e. a single round can be a member of many subsequences) and which may be defined by (among other things) context available to us before we choose an action (which will give us group-wise accuracy guarantees) and the actions we choose to play.

First we introduce a general/abstract framework for subsequence regret. In our formulation, there will be d subsequences on which the learner will want to guarantee that their cost is comparable to the cost of the best action in hindsight. Whether or not each round t is included in a subsequence i will be determined by a subsequence selection function $E(t, a^t, x^t)$ which can depend on the round t , the action a^t chosen by the learner at that round, and x^t , which represents any additional context or outside information available to the learner before round t . This is expressive enough to define subsequences like “Rounds 1000 through 2000”, “Rounds on which we play action 3”, “Rounds on which the person with features x^t we are making a decision about is female”, and combinations thereof. In fact the subsequence selection functions can also depend on the history in arbitrary ways, and so can encode things like “rounds *following* rounds in which we played action 5”, and so on — but we will elide this from our notation to avoid clutter. Although our language of “subsequence selection” is suggestive that each round will either be a part of a given subsequence or not (and most of our applications will be of this sort), we will define subsequence selection functions more generally so that they can take values in $[0, 1]$, which we can interpret as having the ability to fractionally select rounds to participate in each subsequence.

Definition 58 (Subsequence Selection and Regret) *Fix an action space $\mathcal{A} = [k]$ and a context space \mathcal{X} . Let \mathcal{E} be a collection of m subsequence selection functions $E : [T] \times \mathcal{A} \times \mathcal{X} \rightarrow [0, 1]$. The interaction between the learner and the adversary proceeds in rounds $t = 1$ to T :*

1. *The learner observes a context $x^t \in \mathcal{X}$ (if any) that may be chosen by the adversary.*
2. *The learner chooses a distribution over actions $p^t \in \Delta\mathcal{A}$*
3. *The adversary chooses a cost vector $c^t \in [0, 1]^k$.*
4. *The learner experiences expected cost $\langle p^t, c^t \rangle$.*

Fix a transcript $\pi^T = \{(x^t, p^t, c^t)\}_{t=1}^T$. The expected regret to action $i \in [k]$ on subsequence $E \in \mathcal{E}$ is:

$$\text{Reg}(\pi^T, E, i) = \sum_{t=1}^T \mathbb{E}_{j \sim p^t} [E(t, j, x^t) (c_j^t - c_i^t)]$$

The learner has expected \mathcal{E} -subsequence regret bounded by α if:

$$\max_{E \in \mathcal{E}, i \in [k]} \text{Reg}(\pi^T, E, i) \leq \alpha$$

Finally we introduce some terminology. If a collection of events \mathcal{E} contains events E that are independent of the played action (i.e. such that for all $a, a' \in \mathcal{A}$, we always have $E(t, a, x) = E(t, a', x)$), then we say that \mathcal{E} is an action independent collection of events. When writing the subsequence selection functions E , we elide dependence on parameters that are not used: for

example, if a subsequence depends only on the action chosen we will write $E(a)$ rather than $E(t, a, x)$, etc.

Remark 8.3.1 Observe that the “simple” regret we have mostly worked with thus far is just the special case of subsequence regret for the single subsequence that includes all rounds: $E(t) = 1$ for all t .

We can cast the problem of obtaining no-subsequence regret over k actions and a collection of subsequences \mathcal{E} of size $|\mathcal{E}| = m$ as a $d = (m \cdot k)$ -objective optimization game.

Definition 59 (Subsequence Regret Multiobjective Optimization Game)

Fix a collection of m subsequence selection functions \mathcal{E} and k actions $[k]$. We define a $d = m \cdot k$ multiobjective optimization game in which the environment at each round t is:

1. The action space for the learner is $\mathcal{A}^t = \Delta[k]$ and the action space for the adversary is $\mathcal{B}^t = [0, 1]^k$.
2. The loss increment functions are defined as follows. For each subsequence indicator function $E \in \mathcal{E}$ and action $i \in [k]$, we define loss increment:

$$\ell_{E,i}^t(p^t, c^t) = \mathbb{E}_{j \sim p^t} [E(t, j, x^t) (c_j^t - c_i^t)]$$

We can immediately apply Theorem 47 to get a bound on any subsequence regret problem:

Theorem 48 Fix a collection of m subsequence selection functions \mathcal{E} and a set of k actions. If we run Algorithm 30 on the multiobjective optimization game defined in Definition 59, then against any sequence of costs, we generate a transcript π^T that has subsequence regret bounded by:

$$\max_{E \in \mathcal{E}, i \in [k]} \text{Reg}(\pi^T, E, i) \leq 4\sqrt{T(\ln m + \ln k)}$$

Theorem 49 We first verify that the construction in Definition 59 satisfies the conditions required of a multiobjective optimization game. Indeed, the action sets $\mathcal{A}^t, \mathcal{B}^t$ are closed and convex. The loss functions are linear in both \mathcal{A}^t and \mathcal{B}^t (and hence convex/concave), and bounded in $[-1, 1]$ as required.

Next, we verify that the adversary-moves-first value of the game at each round t is 0: $v_A^T = 0$. To see this, fix any action c for the adversary. Let $i^*(c) \in \arg \min_{i \in [k]} c_i$ be a minimum coordinate of the cost vector c . If we let $p(c)$ be the distribution that places all of its weight on action $i^*(c)$. We have

that:

$$\begin{aligned}
v_A^t &= \max_{c \in \mathcal{B}^t} \min_{p \in \mathcal{A}^t} \max_{E \in \mathcal{E}, i \in [k]} \ell_{E,i}^t(p, c) \\
&\leq \max_{c \in \mathcal{B}^t, E \in \mathcal{E}, i \in [k]} \ell_{E,i}^t(p(c), c) \\
&= \max_{c \in \mathcal{B}^t, E \in \mathcal{E}, i \in [k]} E(t, i^*(c), x^t)(c_{i^*(c)} - c_i) \\
&\leq 0
\end{aligned}$$

where the last inequality follows from the fact that by definition of i^* , $c_{i^*(c)} \leq c_i$ for all i .

The bound then follows from Theorem 47. For every E, i :

$$\begin{aligned}
\text{Reg}(\pi^T, E, i) &= \mathbb{E}_{j \sim p^t} [E(t, j, x^t)(c_j^t - c_i^t)] \\
&= \sum_{t=1}^T \ell_{E,i}^t(p^t, c^t) \\
&= \sum_{t=1}^T (\ell_{E,i}^t(p^t, c^t) - v_A^t) \\
&\leq 4\sqrt{T \ln(m \cdot k)}
\end{aligned}$$

Remark 8.3.2 Note that this is the same (!) bound we get for simple regret with Multiplicative Weights or Exponential Weights (Theorem 77), except the $\ln k$ term has been replaced with a $\ln(mk)$ term, where m is the number of subsequences we are interested in.

So we have an extremely general technique for minimizing regret simultaneously across *any* collection of m subsequences, with regret bounds growing only logarithmically in m . But running Algorithm 30 generically requires computing a minimax equilibrium at each step to find the distribution p^t to play. In certain cases we can do better and get a closed form for p^t .

8.3.1 Action Independent Subsequences

Recall that a set of subsequence selection functions \mathcal{E} is *action independent* if for every $E \in \mathcal{E}$, for every t, x , and for every $a, a' \in \mathcal{A}$: $E(t, a, x) = E(t, a', x)$. In other words, the subsequences do not depend on the actions chosen by the algorithm. We can write the subsequence selection functions as $E(t, x)$ in this case. Just as we did when deriving the exponential weights algorithm in Section B.4, in this case we can derive a closed form for the minimax equilibrium that needs to be played at each round by Algorithm 30.

Algorithm 31 Getting Action Independent Subsequence Regret

Given A collection \mathcal{E} of m action-independent subsequence selection functions.

for $t = 1$ to T **do**

Observe context x^t

Define the distribution $p^t \in \Delta[k]$ as:

$$p_i^t = \frac{\sum_{E \in \mathcal{E}} E(t, x^t) \cdot \exp\left(\frac{\eta}{2} \sum_{t'=1}^{t-1} E(t', x^{t'}) (\langle p^{t'}, c^{t'} \rangle - c_i^{t'})\right)}{\sum_{j=1}^k \sum_{E \in \mathcal{E}} E(t, x^t) \cdot \exp\left(\frac{\eta}{2} \sum_{t'=1}^{t-1} E(t', x^{t'}) (\langle p^{t'}, c^{t'} \rangle - c_j^{t'})\right)}$$

Play distribution p^t .

Remark 8.3.3 *The algorithm has a simple form: For each subsequence E , it computes a weight proportional to the exponential of the regret to action i on that subsequence, and sums up the weights, scaling each one by $E(t, x^t)$, the degree to which subsequence E is active at round t . Then it gives action i weight proportional to this sum.*

For binary subsequences (in which each round t is either contained in the subsequence $E(t, x^t) = 1$ or not $E(t, x^t) = 0$, it simply zeros out the weight for each of the subsequences that are inactive at round t .

Theorem 50 *For any collection \mathcal{E} of m action independent subsequence selection functions, Algorithm 31 implements Algorithm 30 for the Subsequence Regret Multiobjective Optimization Game (Definition 59), and hence obtains the regret bound from Theorem 48:*

$$\max_{E \in \mathcal{E}, i \in [k]} \text{Reg}(\pi^T, E, i) \leq 4\sqrt{T(\ln m + \ln k)}$$

Proof 72 *Algorithm 30 needs to play a distribution p^t at each round that is a minimax equilibrium for the game with utility function defined as:*

$$\begin{aligned} u^t(p, c) &= \sum_{E \in \mathcal{E}, i \in [k]} \frac{\exp\left(\frac{\eta}{2} \sum_{t'=1}^{t-1} \ell_{E,i}^{t'}(p^{t'}, c^{t'})\right)}{\sum_{E', j} \exp\left(\frac{\eta}{2} \sum_{t'=1}^{t-1} \ell_{E',j}^{t'}(p^{t'}, c^{t'})\right)} \ell_{E,i}^t(p, c) \\ &= \sum_{E \in \mathcal{E}, i \in [k]} \frac{\exp\left(\frac{\eta}{2} \sum_{t'=1}^{t-1} E(t', x^{t'}) (\langle p^{t'}, c^{t'} \rangle - c_i^{t'})\right)}{\sum_{E', j} \exp\left(\frac{\eta}{2} \sum_{t'=1}^{t-1} E'(t', x^{t'}) (\langle p^{t'}, c^{t'} \rangle - c_j^{t'})\right)} E(t, x^t) (\langle p, c \rangle - c_i) \end{aligned}$$

This game has value 0. Thus we need to show that for the distribution p^t defined in Algorithm 31, for all cost vectors c , $u^t(p^t, c) \leq 0$.

Observe from the way we have defined p^t , can write $u^t(p, c)$ as:

$$\begin{aligned} u^t(p, c) &= \sum_{i=1}^k \frac{1}{\Phi^t} p_i^t (\langle p, c \rangle - c_i) \\ &= \frac{1}{\Phi^t} \sum_{i=1}^k p_i^t \left(\sum_{j=1}^k p_j c_j - c_i \right) \end{aligned}$$

where Φ^t is a normalization factor defined as

$$\Phi^t = \frac{\sum_{E', j} \exp\left(\frac{\eta}{2} \sum_{t'=1}^{t-1} E'(t, x^{t'}) (\langle p^{t'}, c^{t'} \rangle - c_j^{t'})\right)}{\sum_{E', j} E'(t, x^t) \exp\left(\frac{\eta}{2} \sum_{t'=1}^{t-1} E'(t, x^{t'}) (\langle p^{t'}, c^{t'} \rangle - c_j^{t'})\right)}$$

So, plugging in $p = p^t$ we have that for all c :

$$\begin{aligned} u^t(p^t, c) &= \frac{1}{\Phi^t} \sum_{i=1}^k p_i^t \left(\sum_{j=1}^k p_j^t c_j - c_i \right) \\ &= \frac{1}{\Phi^t} \left(\sum_{i=1}^k \sum_{j=1}^k p_i^t p_j^t c_j - \sum_{i=1}^k p_i^t c_i \right) \\ &= \frac{1}{\Phi^t} \left(\sum_{i=1}^k \sum_{j=1}^k p_j^t p_i^t c_i - \sum_{i=1}^k p_i^t c_i \right) \\ &= \frac{1}{\Phi^t} \sum_{i=1}^k p_i^t c_i \left(\sum_j p_j^t - 1 \right) \\ &= 0 \end{aligned}$$

Since p^t forms a probability distribution and so $\sum_j p_j^t = 1$.

So what are some interesting examples of action-independent subsequence regret? We briefly explore a few:

8.3.1.1 Adaptive Regret

The basic regret guarantees we have proven roughly speaking promise the following: On average, over the sequence of rounds $1, \dots, T$, the cost of the algorithm is at most the cost of the best (lowest cost) fixed action in hindsight — up to a regret bound of $O(\sqrt{T})$. But the best fixed action in hindsight is the action that has the lowest cost after all T rounds, and if we start keeping track of regret at some round $t' > 0$ — or stop keeping track of regret at some round $t < T$, then we do not necessarily have the same guarantees. On

the other hand, an *adaptive* regret guarantee asks that simultaneously for all $0 < t' < t \leq T$, our algorithm has low regret to the best fixed action as evaluated on the subsequence $[t', t]$.

Definition 60 (Adaptive Regret) Fix an action space $\mathcal{A} = [k]$ and a transcript $\pi^T = \{p^s, c^s\}_{s=1}^T$. The expected regret to action $i \in [k]$ on the subsequence $[t', t]$ for $0 < t' < t \leq T$ is:

$$\text{Reg}(\pi^T, [t', t], i) = \sum_{s=t'}^t (\langle p^s, c^s \rangle - c_i^s)$$

The learner has expected adaptive regret bounded by α if:

$$\max_{i \in [k], 0 < t' < t \leq T} \text{Reg}(\pi^T, [t', t], i) \leq \alpha$$

We observe that adaptive regret is a special case of subsequence regret on the action-independent subsequences $\mathcal{E}_{\text{Adapt}} = \{E_{s',s}\}_{0 < s' < s \leq T}$, where $E_{s',s}(t) = 1$ if $s' \leq t \leq s$ and $E_{s',s}(t) = 0$ otherwise. Observe that $|\mathcal{E}_{\text{Adapt}}| \leq T^2$, and so we can apply Theorem 48 to immediately conclude:

Theorem 51 When instantiated with $\mathcal{E} = \mathcal{E}_{\text{Adapt}}$, Algorithm 31 obtains adaptive regret bounded by:

$$\max_{i \in [k], 0 < t' < t \leq T} \text{Reg}(\pi^T, [t', t], i) \leq 4\sqrt{T(2 \ln T + \ln k)}$$

8.3.1.2 Group-wise Regret

Sometimes, we will receive information or *context* $x^t \in \mathcal{X}$ about the decision we are about to make at round t before we make it. For example, when making a weather prediction, we might get to observe atmospheric measurements; when making healthcare decisions we will observe an individual's medical history and current vitals, etc. We might want to obtain diminishing regret to the best fixed action in hindsight not just overall, but also *conditional* on relevant pieces of information. For example, when we are making decisions about people and we have fairness concerns, maybe we want to have no regret overall, but also on subsequences of people corresponding to men and women, and also on subsequences of people corresponding to different ethnicities, and also on subsequences of people corresponding to different income brackets, etc. We might also want to condition on subsequences of people corresponding to individuals that have features that we think are relevant to the task at hand — in a medical setting, perhaps subsequences of people with high blood pressure, subsequences of people with a family history of diabetes, etc. The key thing is that these subsequences are intersecting: a single person will have a gender, an ethnicity, an income bracket, a unique medical history, etc. So it would not make sense to try and run a different no regret algorithm for people

with each of the characteristics that we care about, since what would we do when we encountered someone who fit into more than one group?

We can nevertheless ask for a single algorithm to make decisions that has diminishing regret on *all* of the subsequences defined by the groups we care about.

Definition 61 (Group-Wise Regret) Fix an action space $\mathcal{A} = [k]$, a context space \mathcal{X} , a collection of groups $\mathcal{G} \in 2^{\mathcal{X}}$ and a transcript $\pi^T = \{x^t, p^t, c^t\}_{t=1}^T$. The expected regret to action $i \in [k]$ on the subsequence corresponding to group $G \in \mathcal{G}$ is::

$$\text{Reg}(\pi^T, G, i) = \sum_{t: x^t \in G} (\langle p^s, c^s \rangle - c_i^s)$$

The learner has expected groupwise regret over \mathcal{G} bounded by α if:

$$\max_{G \in \mathcal{G}} \text{Reg}(\pi^T, G, i) \leq \alpha$$

We observe that groupwise regret is a special case of subsequence regret on the action-independent subsequences $\mathcal{E}_{\mathcal{G}} = \{E_G\}_{G \in \mathcal{G}}$, where $E_G(x^t) = 1$ if $x^t \in G$ and $E_G(x^t) = 0$ otherwise. Observe that $|\mathcal{E}_{\mathcal{G}}| = |\mathcal{G}|$, and so we can apply Theorem 48 to immediately conclude:

Theorem 52 Fix any collection of groups $\mathcal{G} \in 2^{\mathcal{X}}$. When instantiated with $\mathcal{E} = \mathcal{E}_{\mathcal{G}}$, Algorithm 31 obtains adaptive regret bounded by:

$$\max_{G \in \mathcal{G}} \text{Reg}(\pi^T, G, i) \leq 4\sqrt{T(\ln |\mathcal{G}| + \ln k)}$$

8.3.2 General Subsequences

The general form of Algorithm 31 was to weight each action with probability proportional to the (exponential of) the regret to that action on each subsequence, weighted by the current activation of each of the subsequences. This worked when the subsequences had activations $E(t, x^t)$ that were independent of the action a^t chosen at round t — but in general, does not make sense when the events $E(t, x^t, a^t)$ depend on the action chosen. The solution (since we need to select an action from a distribution that depends in some way on the subsequence activations, which in turn depend on which action we pick) is to compute a distribution by solving for a fixed point. This is what the general subsequence regret algorithm below (Algorithm 32) does. The algorithm will seem mysterious at first blush, but we will derive it from first principles in the proof of Theorem 53.

Algorithm 32 Getting General Diminishing Subsequence Regret

Given A collection \mathcal{E} of m subsequence selection functions.

for $t = 1$ to T **do**

Observe context x^t

For each $E \in \mathcal{E}$ and each $i \in [k]$, define the weights:

$$w_{E,i}^t = \frac{\exp\left(\frac{\eta}{2} \sum_{t'=1}^{t-1} \mathbb{E}_{j' \sim p^{t'}} \left[E(t', x^{t'}, j') (c_{j'}^{t'} - c_i^{t'}) \right]\right)}{\sum_{E',j} \exp\left(\frac{\eta}{2} \sum_{t'=1}^{t-1} \mathbb{E}_{j' \sim p^{t'}} \left[E'(t', x^{t'}, j') (c_{j'}^{t'} - c_j^{t'}) \right]\right)}$$

Define the $k \times k$ matrix A so that:

$$A_{i,j} = \frac{\sum_{E \in \mathcal{E}} w_{E,j}^t E(t, x^t, i)}{\sum_{i' \in [k]} \sum_{E \in \mathcal{E}} w_{E,i'}^t E(t, x^t, j)}$$

Compute $p^t \in \Delta[k]$, a probability distribution such that:

$$Ap^t = p^t$$

i.e. an eigenvector of A with eigenvalue 1.

Play the distribution p^t .

Theorem 53 *Algorithm 32 is well defined (i.e. the claimed eigenvector exists). Moreover, for any collection \mathcal{E} of m subsequence selection functions, Algorithm 31 implements Algorithm 30 for the Subsequence Regret Multiobjective Optimization Game (Definition 59), and hence obtains the regret bound from Theorem 48:*

$$\max_{E \in \mathcal{E}, i \in [k]} \text{Reg}(\pi^T, E, i) \leq 4\sqrt{T(\ln m + \ln k)}$$

Proof 73 *We will derive the algorithm from first principles, which will help build intuition, rather than just proving that it is correct “out of nowhere”.*

Algorithm 30 needs to play a distribution p^t at each round that is a minimax equilibrium for the game with utility function defined as:

$$\begin{aligned} u^t(p, c) &= \sum_{E \in \mathcal{E}, i \in [k]} w_{E,i}^t \ell_{E,i}^t(p, c) \\ &= \sum_{E \in \mathcal{E}, i \in [k]} w_{E,i}^t \mathbb{E}_{j \sim p} [E(t, x^t, j) (c_j - c_i)] \end{aligned}$$

where

$$w_{E,i}^t = \frac{\exp\left(\frac{\eta}{2} \sum_{t'=1}^{t-1} \ell_{E,i}^{t'}(p^{t'}, c^{t'})\right)}{\sum_{E',j} \exp\left(\frac{\eta}{2} \sum_{t'=1}^{t-1} \ell_{E',j}^{t'}(p^{t'}, c^{t'})\right)}$$

We can expand out the expectation and inner product in the utility function:

$$\begin{aligned}
u^t(p, c) &= \sum_{E \in \mathcal{E}, i \in [k]} w_{E,i}^t \mathbb{E}_{j \sim p} [E(t, x^t, j) (c_j - c_i)] \\
&= \sum_{i \in [k]} \sum_{j \in [k]} \sum_{E \in \mathcal{E}} w_{E,i}^t p_j E(t, x^t, j) (c_j - c_i) \\
&= \sum_{i \in [k]} \sum_{j \in [k]} \sum_{E \in \mathcal{E}} w_{E,i}^t E(t, x^t, j) p_j c_j - \sum_{i \in [k]} \sum_{j \in [k]} \sum_{E \in \mathcal{E}} w_{E,j}^t E(t, x^t, i) p_i c_j \\
&= \sum_{j \in [k]} c_j \underbrace{\left(\sum_{i \in [k]} \sum_{E \in \mathcal{E}} w_{E,i}^t E(t, x^t, j) p_j - \sum_{i \in [k]} \sum_{E \in \mathcal{E}} w_{E,j}^t E(t, x^t, i) p_i \right)}_{\chi_j^t(p)}
\end{aligned}$$

So:

$$u^t(p, c) = \sum_{j \in [k]} c_j \chi_j^t(p)$$

We know that the value of this game is 0, and so if p^* is a minimax strategy, it must be that for all $c \in [0, 1]^k$, $u^t(p^*, c) = \sum_{j \in [k]} c_j \chi_j^t(p^*) \leq 0$. In order for this to be true, it must be that for all j , $\chi_j^t(p^*) \leq 0$ (since otherwise if there was such a j such that $\chi_j^t(p^*) > 0$ the adversary could set $c_j = 1$ and $c_{j'} = 0$ for all $j' \neq j$, which would lead to $u^t(p^*, c) > 0$, a contradiction). Moreover, we can see by symmetry that for all p , $\sum_{j=1}^k \chi_j^t(p) = 0$. Therefore all of the inequalities must be equalities: for all j , $\chi_j^t(p) = 0$. From this we get a set of equalities characterizing p^* : For all j :

$$p_j^* = \frac{\sum_{i \in [k]} \sum_{E \in \mathcal{E}} w_{E,j}^t E(t, x^t, i) p_i^*}{\sum_{i \in [k]} \sum_{E \in \mathcal{E}} w_{E,i}^t E(t, x^t, j)}$$

Note that if we define the $k \times k$ matrix A such that

$$A_{i,j} = \frac{\sum_{E \in \mathcal{E}} w_{E,j}^t E(t, x^t, i)}{\sum_{i' \in [k]} \sum_{E \in \mathcal{E}} w_{E,i'}^t E(t, x^t, j)}$$

Then these constraints simplify to:

$$Ap^* = p^*$$

In other words, p is an eigenvector of A with eigenvalue 1. Observe that since we know the game has value 0, we know that such a p^* must exist (i.e. this matrix must have an eigenvector with eigenvalue 1), and p^* must therefore be a minimax strategy.

8.3.2.1 Swap Regret

A particular form of conditional regret that turns out to have close connections to calibration, as well as to agent incentives in strategic interactions, is called *swap regret*. Informally, a sequence of actions has no swap regret if playing that sequence is guaranteed to have higher utility (or lower cost) compared to playing any fixed function of the actions suggested. We define it formally below:

Definition 62 Fix an action space for a learner \mathcal{A} and a class Φ of “action modification rules” $\phi : \mathcal{A} \rightarrow \mathcal{A}$. Given a sequence of action distributions and cost vectors $\pi^T = \{(p^1, c^1), \dots, (p^T, c^T)\}$, the learner’s regret to a strategy modification rule $\phi \in \Phi$ is:

$$\text{Reg}(\pi^T, \phi) = \sum_{t=1}^T \mathbb{E}_{a^t \sim p^t} \left[\left(c_{a^t}^t - c_{\phi(a^t)}^t \right) \right]$$

We say that the learner has Φ -regret bounded by α if:

$$\max_{\phi \in \Phi} \text{Reg}(\pi^T, \phi) \leq \alpha$$

If Φ is the set of all action modification rules $\phi : \mathcal{A} \rightarrow \mathcal{A}$ we refer to Φ regret as *swap regret*.

We first observe that in a k action game, swap regret is at most k times larger than the *subsequence regret* over the set of k subsequences $\mathcal{E} = \{E_a\}_{a \in [k]}$ corresponding to rounds in which the learner i played a particular action a : $E_a(a_i^t) = \mathbb{1}[a_i^t = a]$. This kind of subsequence regret is also called *internal regret*.

Definition 63 A learner has *internal regret bounded by α* if they have subsequence regret bounded by α for the collection of subsequences $\mathcal{E}_{Int} = \{E_a\}_{a \in [k]}$, where for each a , $E_a(a^t) = \mathbb{1}[a^t = a]$. In other words, if for each $i \in [k]$ and for each $a \in [k]$:

$$\text{Reg}(\pi^T, E_a, i) = \sum_{t=1}^T p_a^t (c_a^t - c_i^t) \leq \alpha$$

Lemma 8.3.1 If a k -action learner has internal regret bounded by α on transcript π^T , then their swap regret is bounded by $k\alpha$ — i.e. for every $\phi : \mathcal{A} \rightarrow \mathcal{A}$:

$$\sum_{t=1}^T \mathbb{E}_{a^t \sim p^t} \left[\left(c_{a^t}^t - c_{\phi(a^t)}^t \right) \right] \leq k\alpha$$

Proof 74 We calculate:

$$\begin{aligned}
\sum_{t=1}^T \mathbb{E}_{a^t \sim p^t} \left[\left(c_{a^t}^t - c_{\phi(a^t)}^t \right) \right] &= \sum_{t=1}^T \sum_{a \in \mathcal{A}} p_a^t \cdot \left(c_a^t - c_{\phi(a)}^t \right) \\
&= \sum_{a \in \mathcal{A}} \sum_{t=1}^T p_a^t \cdot \left(c_a^t - c_{\phi(a)}^t \right) \\
&= \sum_{a \in \mathcal{A}} \text{Reg}(\pi^T, E_a, \phi(a)) \\
&\leq k\alpha
\end{aligned}$$

We can therefore invoke Theorem 53 to show that Algorithm 32 can be used to obtain low internal (and hence swap) regret:

Theorem 54 When Algorithm 32 is instantiated with the k events \mathcal{E}_{Int} , it guarantees internal regret bounded by:

$$\max_{a \in [k], i \in [k]} \text{Reg}(\pi^T, E_a, i) \leq 4\sqrt{T(2 \ln k)}$$

By Lemma 8.3.1 it therefore also guarantees swap regret bounded by:

$$\max_{\phi: \mathcal{A} \rightarrow \mathcal{A}} \sum_{t=1}^T \mathbb{E}_{a^t \sim p^t} \left[\left(c_{a^t}^t - c_{\phi(a^t)}^t \right) \right] \leq 4k\sqrt{T(2 \ln k)}$$

8.3.2.2 Mixing and Matching Guarantees

Our multi-objective/subsequence regret framework gives us the flexibility to easily mix and match guarantees. For example, given k actions \mathcal{A} and a collection of groups $\mathcal{G} \in 2^{\mathcal{X}}$ if we instantiate Algorithm 32 with the events $\mathcal{E} = \mathcal{E}_{\mathcal{G}} \times \mathcal{E}_{\text{int}} = \{E_G(x_t) \cdot E_a(a^t)\}_{G \in \mathcal{G}, a \in \mathcal{A}}$, then we get an algorithm that has *groupwise* swap regret. Similarly if we instantiate the algorithm with the events $\mathcal{E}_{\text{Adapt}} \times \mathcal{E}_{\text{int}}$ we get adaptive-swap regret. We can similarly get groupwise adaptive swap regret, and a variety of other kinds of guarantees.

8.4 Sequential Adversarial Multicalibration

In this section we give algorithms that can promise mean and quantile multicalibration in the sequential setting, against an adversary. It will be more convenient for us to bound ℓ_∞ calibration error (K_∞) rather than the ℓ_2 calibration error that we have been working with, although getting sequential calibration bounds for ℓ_2 calibration error at the same rates is similar.

For convenience we will also use a slightly modified definition that “buckets” predictions when computing calibration error rather than discretizing them, although there is a simple translation between the two measures that comes from “snapping” all predictions in a bucket to the center-point of that bucket.

8.4.1 A Bucketed Calibration Definition

Recall that calibration asks informally that $\mathbb{E}_{(x,y) \sim \mathcal{D}}[y|f(x) = v] \approx v$ for all v . To avoid problems with the conditioning event, we have thus far restricted our attention to models f that have bounded range $R(f) \subseteq [1/m]$. A different solution is to allow f to have arbitrary range in $[0, 1]$, but not to condition on the event that $f(x) = v$, but rather to condition on the event that $f(x) \approx v$ for an appropriate formalization of \approx . We will do this through bucketing:

Definition 64 *Let m be a bucket coarseness parameter. For $i \in \{1, \dots, m-1\}$ let $B_m(i) = [\frac{i-1}{m}, \frac{i}{m})$ and let $B_m(m) = [\frac{m-1}{m}, 1]$. Collectively, $B_m(i)$ for $i \in [m]$ form a set of m “buckets” of width $1/m$ that partition the unit interval.*

We now give a variant of our ℓ_∞ calibration definition in which informally, the conditioning event $f(x) \approx v$ is taken to mean “ $f(x)$ and v are in the same bucket”.

Definition 65 (Bucketed Multicalibration Error in the Distributional Setting)

Fix a predictor $f : \mathcal{X} \rightarrow [0, 1]$, a collection of groups \mathcal{G} , and a bucket coarseness parameter m . The calibration error of f on a group g with respect to bucketing coarseness m , on distribution \mathcal{D} is defined to be:

$$C_\infty(f, m, g, \mathcal{D}) =$$

$$\max_{i \in [m]} \Pr_{(x,y) \sim \mathcal{D}} [f(x) \in B_m(i) | g(x) = 1] \cdot \left| \mathbb{E}_{(x,y) \sim \mathcal{D}} [f(x) - y | f(x) \in B_m(i), g(x) = 1] \right|$$

We say that f satisfies (α, m) -multicalibration with respect to \mathcal{G} on \mathcal{D} if for every $g \in \mathcal{G}$:

$$C_\infty(f, m, g, \mathcal{D}) \leq \frac{\alpha}{\mu(g)}$$

This is identical to our definition of K_∞ except that the condition that $f(x) = v$ has been replaced with the condition that $f(x) \in B(i)$. We can give a corresponding definition in the sequential setting:

Definition 66 (Bucketed Multicalibration Error in the Sequential Setting)

Fix a collection of groups \mathcal{G} , a transcript $\pi = \{(x_1, p_1, y_1), \dots, (x_T, p_T, y_T)\}$, and a bucket coarseness parameter m . For any $i \in [m]$ and $g \in \mathcal{G}$, let $S(\pi, g, i) = \{t : g(x_t) = 1, p_t \in B_m(i)\}$ and $S(\pi, g) = \{t : g(x_t) = 1\}$. Let $n(\pi, g, i) = |S(\pi, g, i)|$ and let $n(\pi, g) = |S(\pi, g)|$.

The calibration error of π on a group g with respect to bucketing coarseness m is defined to be:

$$C_\infty(\pi, m, g) = \max_{i \in [m]} \frac{n(\pi, g, i)}{n(\pi, g)} \cdot \left| \frac{\sum_{t \in S(\pi, g, i)} (p_t - y_t)}{n(\pi, g, i)} \right|$$

We say that π satisfies (α, m) multicalibration with respect to \mathcal{G} if for every $g \in \mathcal{G}$:

$$C_\infty(\pi, m, g) \leq \frac{\alpha T}{n(\pi, g)}$$

Expanding out the definitions we find that equivalently, π satisfies (α, m) multicalibration error with respect to \mathcal{G} if:

$$\max_{g \in \mathcal{G}, i \in [m]} \left| \sum_{t \in S(\pi, g, i)} (p_t - y_t) \right| \leq \alpha T$$

We can analogously define a “bucketed” definition of quantile multicalibration:

Definition 67 (Bucketed Sequential Quantile Multicalibration Error)

Fix a collection of groups \mathcal{G} , a transcript $\pi = \{(x_1, p_1, y_1), \dots, (x_T, p_T, y_T)\}$, and a bucket coarseness parameter m . The quantile calibration error of π on a group g with respect to bucketing coarseness m and target quantile q is defined to be:

$$Q_\infty(\pi, m, g) = \max_{i \in [m]} \frac{n(\pi, g, i)}{n(\pi, g)} \cdot \left| \frac{\sum_{t \in S(\pi, g, i)} (\mathbb{1}[y_t \leq p_t] - q)}{n(\pi, g, i)} \right|$$

We say that π satisfies (α, m) quantile multicalibration with respect to \mathcal{G} if for every $g \in \mathcal{G}$:

$$Q_\infty(\pi, m, g) \leq \frac{\alpha T}{n(\pi, g)}$$

Expanding out the definitions we find that equivalently, π satisfies (α, m) quantile multicalibration error with respect to \mathcal{G} if:

$$\max_{g \in \mathcal{G}, i \in [m]} \left| \sum_{t \in S(\pi, g, i)} (\mathbb{1}[y_t \leq p_t] - q) \right| \leq \alpha T$$

8.4.2 Bounding Sequential Multicalibration Error

We can encode multicalibration as a multiobjective optimization problem and solve it within our general online multiobjective optimization framework. We will have an objective for each sign $\sigma \in \{-1, 1\}$, prediction p , and group g . For mean multicalibration, we use the following encoding:

Definition 68 (*r*-Encoding of Mean Multicalibration as Multiobjective Optimization)

Given a collection of groups \mathcal{G} and a coarseness parameter m we construct a $2|\mathcal{G}|m$ -dimensional instance of the multiobjective optimization game as follows: At each round t :

1. The strategy space for the Learner is $\mathcal{A}^t = \Delta\{0, 1/r, 2/r, \dots, 1\}$, the set of distributions over a finite $1/r$ -discretization of $[0, 1]$.
2. The strategy space for the Adversary is $\mathcal{B}^t = [0, 1]$.
3. For each $\sigma \in \{-1, 1\}$, $g \in \mathcal{G}$, and $p \in \{0, 1/m, \dots, (m-1)/m\}$ we construct a loss increment function:

$$\ell_{\sigma, g, p}(\mathcal{P}_t, y_t) = \sigma \cdot \mathbb{E}_{p_t \sim \mathcal{P}^t} [\mathbb{1}[p_t \in B(p)] \cdot g(x_t)(p_t - y_t)]$$

Remark 8.4.1 Observe that this construction will allow the algorithm to play distributions over a $1/r$ -discretization of the unit interval, whereas we measure calibration error with respect to a $1/m$ bucketing. In general it will be important that we allow $r \gg m$. r here is a nuisance parameter — a discretization that we need to allow in order for the minimax theorem to apply, and we can take it to be as large as we want without affecting the final bounds. m on the other hand is a key parameter of the problem, effecting both the granularity of our final guarantees and the number of loss functions in our multiobjective optimization problem.

We recall our generic multiobjective optimization algorithm, instantiated for the multicalibration problem:

Algorithm 33 Multicalibration via Multiobjective Optimization

for $t = 1$ to T **do**

 Define the distribution $w^t \in \Delta[2|\mathcal{G}|m]$ as:

$$w_{\sigma, G, s'}^t \propto \exp\left(\frac{\eta}{2} \sum_{t'=1}^{t-1} \sigma \cdot \mathbb{E}_{\hat{s}^{t'} \sim p^{t'}} [\mathbb{1}[\hat{s}^{t'} \in B(s')] \cdot G(t', x^{t'}) (\hat{s}^{t'} - s^{t'})]\right)$$

 Define a zero-sum game in which the minimization player's action are \mathcal{A}^t , the maximization players actions are \mathcal{B}^t , and the utility function is:

$$u^t(p^t, s^t) = \sum_{\sigma, G, s'} w_{\sigma, G, s'}^t \sigma \cdot \mathbb{E}_{\hat{s}^t \sim p^t} [\mathbb{1}[\hat{s}^t \in B(s')] \cdot G(t, x^t) (\hat{s}^t - s^t)]$$

 Compute a minimax equilibrium strategy of this game p^t for the Learner and sample a forecast \hat{s}^t from p^t .

We can read off bounds for our multicalibration error from our general theorem about unbiased prediction (Theorem ??)

Theorem 55 Fix an r -encoding of an (α, m) -multicalibration problem with respect to a collection of events \mathcal{G} (Definition 68). Running Algorithm 33 guarantees against any adversary that the (α, m) -multicalibration error with respect to \mathcal{G} will be bounded by:

$$\mathbb{E}[\alpha] \leq 4\sqrt{T \ln(2 \cdot |\mathcal{G}|m)} + \frac{T}{r}$$

Remark 8.4.2 If we instantiate Theorem 56 with $r \geq \sqrt{T}$, then the multicalibration error we obtain is at most $5\sqrt{T \ln(2 \cdot |\mathcal{G}|m)}$.

We now proceed to specialize the algorithm, taking advantage of the special structure of the 1-dimensional (multi)calibration problem to give a simpler algorithm that will not require that we solve a general minimax problem at every round. In fact, the algorithm we derive will be *almost* deterministic: it will make predictions that randomize between pairs of points that differ by at most $1/r$ from one another. To derive the algorithm, it will be helpful to first rewrite the utility function of the game that we need to solve to implement Algorithm 33. Because the utility function is linear in the learner's distribution p^t , it will be enough to write the utility as a function of the *pure strategies* $\hat{s}^t \in \{0, 1/r, \dots, 1\}$.

Lemma 8.4.1 Fix any $\hat{s}^t \in \{0, 1/r, \dots, 1\}$. Let $\tilde{s}^t \in \{0, 1/m, 2/m, \dots, (m-1)/m\}$ be the unique point such that $\hat{s}^t \in B(\tilde{s}^t)$. Then we have that:

$$u^t(\hat{s}^t, s^t) = \sum_{G \in \mathcal{G}} \underbrace{(w_{G, \tilde{s}^t}^{t+} - w_{G, \tilde{s}^t}^{t-})}_{w_{G, \tilde{s}^t}^t} G(t, x^t)(\hat{s}^t - s^t)$$

where

$$w_{G, s'}^{t+} \propto \exp\left(\frac{\eta}{2} \sum_{t'=1}^{t-1} \mathbb{E}_{\hat{s}^{t'} \sim p^{t'}} [\mathbb{1}[\hat{s}^{t'} \in B(s')] \cdot G(t', x^{t'}) (\hat{s}^{t'} - s^{t'})]\right)$$

$$w_{G, s'}^{t-} \propto \exp\left(\frac{\eta}{2} \sum_{t'=1}^{t-1} \mathbb{E}_{\hat{s}^{t'} \sim p^{t'}} [\mathbb{1}[\hat{s}^{t'} \in B(s')] \cdot G(t', x^{t'}) (s^{t'} - \hat{s}^{t'})]\right)$$

Algorithm 34 A Sequential Multicalibration Algorithm

Given: m, r , and \mathcal{G} .**for** $t = 1$ to T **do**For each $s \in \left\{0, \frac{1}{m}, \dots, \frac{m-1}{m}\right\}$, compute:

$$V_s = \sum_{G \in \mathcal{G}} w_{G,s}^t G(t, x^t)$$

if $V_s \geq 0$ for all s **then****Predict** $\hat{s}^t = 0$.**else if** $V_s \leq 0$ for all s **then****Predict** $\hat{s}^t = 1$.**else****Choose** $s \in \left\{\frac{1}{m}, \dots, \frac{m-1}{m}\right\}$ such that

$$V_{s-1/m} \cdot V_s \leq 0$$

Choose $q \in [0, 1]$ such that

$$q \cdot V_s + (1 - q) \cdot V_{s-1/m} = 0$$

Predict $\hat{s}^t = s - \frac{1}{r}$ with probability $1 - q$ and **Predict** $\hat{s}^t = s$ with probability q .

Theorem 56 Fix an r -encoding of an (α, m) -multicalibration problem with respect to a collection of events \mathcal{G} (Definition 68). Algorithm 34 implements Algorithm 33 and so against any adversary obtains (α, m) -multicalibration error with respect to \mathcal{G} bounded by:

$$\mathbb{E}[\alpha] \leq 4\sqrt{T \ln(2 \cdot |\mathcal{G}|m)} + \frac{T}{r}$$

Proof 75 We need to show that Algorithm 34 plays a strategy p^t that against any opponent action s^t guarantees utility at most $1/r$ in the game with utility function:

$$u^t(\hat{s}^t, s^t) = \sum_{G \in \mathcal{G}} w_{G, \hat{s}^t}^t G(t, x^t) (\hat{s}^t - s^t)$$

We consider the three cases in the algorithm.

1. **Case 1:** $V_s \geq 0$ for all s . In this case, $\hat{s}^t = 0$ and we have:

$$\begin{aligned} u(\hat{s}^t, s^t) &= u(0, s^t) \\ &= (0 - s^t) \cdot \sum_{G \in \mathcal{G}} w_{G,0}^t G(t, x^t) \\ &= -s^t \cdot V_0 \\ &\leq 0 \end{aligned}$$

where the last inequality follows from the fact that $-s^t \leq 0$ and $V_0 \geq 0$.

2. **Case 2:** $V_s \leq 0$ for all s . In this case, $\hat{s}^t = 1$ and we have:

$$\begin{aligned} u(\hat{s}^t, s^t) &= u(1, s^t) \\ &= (1 - s^t) \cdot \sum_{G \in \mathcal{G}} w_{G, \frac{m-1}{m}}^t G(t, x^t) \\ &= (1 - s^t) \cdot V_{\frac{m-1}{m}} \\ &\leq 0 \end{aligned}$$

where the last inequality follows from the fact that $1 - s^t \geq 0$ and $V_{\frac{m-1}{m}} \leq 0$.

3. **Case 3:** $q \cdot V_s + (1-q) \cdot V_{s-1/m} = 0$. In this case, we can compute the expected utility:

$$\begin{aligned} \mathbb{E}[u(\hat{s}^t, s^t)] &= q \cdot u(s, s^t) + (1-q)u(s-1/m, s^t) \\ &= q \cdot (s - s^t)V_s + (1-q)(s-1/r - s^t)V_{s-\frac{1}{m}} \\ &= (s - s^t) \left(qV_s + (1-q)V_{s-\frac{1}{m}} \right) - \frac{(1-q)}{r} V_{s-\frac{1}{m}} \\ &\leq 0 + \frac{1}{r} |V_{s-\frac{1}{m}}| \\ &\leq \frac{1}{r} \end{aligned}$$

which completes the proof in all cases.

Remark 8.4.3 Just as calibration implies that (in the limit) it is a dominant strategy for any downstream agent with utility that is linear and Lipschitz in the state to choose actions using the best response policy, multicalibration with respect to a collection of events \mathcal{G} guarantees that it will be a dominant strategy to use the best response policy not just overall, but on every subsequence defined by an event $G \in \mathcal{G}$.

Bibliographic Notes and Further Reading

The multi-objective optimization framework from Section 8.2 is from Lee et al. [2022], who derive the algorithm from first principles (deriving exponential weights along the way). Haghtalab et al. [2023] give a variant of this reduction that modularly derives multiobjective guarantees from an arbitrary no-regret algorithm, which is the exposition we follow here. Swap regret was first studied by Foster and Vohra [1999]. Blum and Mansour [2007] gave efficient algorithms for swap and subsequence regret that are very similar to the ones we derive here. Algorithms for adaptive regret have been given by Hazan and Seshadhri [2009], Adamskiy et al. [2012]. Blum and Lykouris [2020] gave the first algorithms for obtaining diminishing groupwise regret.

9

Multicalibration for Real Valued Functions: When Does Multicalibration Imply Accuracy?

CONTENTS

9.1	Beyond Groups	165
9.2	Algorithmically Reducing Multicalibration to Regression	169
9.3	Weak Learning, Multicalibration, and Boosting	172
	References and Further Reading	178

In this section we think about how we can view multicalibration as a *boosting algorithm* for regression — i.e. a way to take a regression algorithm that has the capacity to predict in *slightly* non-trivial ways (better than a constant function), and produce an ensemble of regression functions that can predict optimally. Along the way, we will generalize multicalibration over a set of *groups* \mathcal{G} represented by indicator functions $g : \mathcal{X} \rightarrow \{0, 1\}$ to multicalibration over a collection of arbitrary real valued functions \mathcal{H} , where each $h \in \mathcal{H}$ is a function $h : \mathcal{X} \rightarrow \mathbb{R}$. This generalization will also be useful for us for a number of other applications of multicalibration.

9.1 Beyond Groups

Our study of multicalibration thus far has been predicated on *groups* — i.e. subsets of the feature domain \mathcal{X} . We have represented groups by their indicator functions g , such that $g(x) = 1$ if x is a member of the group, and $g(x) = 0$ otherwise. Recall that what we mean by (perfect) multicalibration of a function $f : \mathcal{X} \rightarrow \mathbb{R}$ on a collection of groups \mathcal{G} is that for every $g \in \mathcal{G}$ and $v \in R(f)$:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [(y - f(x)) | f(x) = v, g(x) = 1] = 0$$

Since $g(x)$ is binary, we can equivalently re-write this multicalibration condition as the requirement that for every $g \in \mathcal{G}$ and $v \in R(f)$:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [g(x)(y - f(x)) | f(x) = v] = 0$$

But although this is an equivalent condition to ask for when g is binary (i.e. a group indicator function), it now makes sense to ask for this condition even if g is an arbitrary real valued function $g : \mathcal{X} \rightarrow \mathbb{R}$. We will use this as our more general definition of multicalibration with respect to an arbitrary class of real valued functions. We will have to define approximate versions of this condition, and we will again use an ℓ_2 -error variant:

Definition 69 (Multicalibration With Respect to Real Valued Functions)

Fix a distribution $\mathcal{D} \in \Delta \mathcal{Z}$ and a model $f : \mathcal{X} \rightarrow [0, 1]$. Let \mathcal{H} be an arbitrary collection of real valued functions $h : \mathcal{X} \rightarrow \mathbb{R}$. We say that f is α -approximately multicalibrated with respect to \mathcal{D} and \mathcal{H} if for every $h \in \mathcal{H}$:

$$K_2(f, h, \mathcal{D}) = \sum_{v \in R(f)} \Pr_{(x,y) \sim \mathcal{D}} [f(x) = v] \left(\mathbb{E}_{(x,y) \sim \mathcal{D}} [h(x)(y - v) | f(x) = v] \right)^2 \leq \alpha$$

There is a close connection between a failure of a model f to be multicalibrated with respect to a class of functions \mathcal{H} and the ability to decrease squared error of f using a simple update using a function $h \in \mathcal{H}$. We summarize the connection with two lemmas showing the connection in each direction.

First, suppose \mathcal{H} contains a model h that has lower squared error than the best *constant* prediction on one of the level-sets of a calibrated model f . Then f is not multi-calibrated with respect to \mathcal{H} . Note that if f is calibrated, it is making the best constant prediction on each of its level sets, so the condition that h makes predictions with lower squared error than the best constant predictor on a level-set of f is the same that it makes predictions with better squared error than f on one of its level-sets.

Lemma 9.1.1 Fix a calibrated model $f : \mathcal{X} \rightarrow \mathbb{R}$. Suppose for some $v \in R(f)$ there is an $h \in \mathcal{H}$ such that:

$$\mathbb{E}[(f(x) - y)^2 - (h(x) - y)^2 | f(x) = v] \geq \alpha$$

Then it must be that:

$$\mathbb{E}[h(x)(y - v) | f(x) = v] \geq \frac{\alpha}{2}$$

Proof 76 We calculate:

$$\begin{aligned}
 & \mathbb{E}_{(x,y) \sim \mathcal{D}} [h(x)(y-v) | f(x) = v] \\
 = & \mathbb{E}_{(x,y) \sim \mathcal{D}} [h(x)y | f(x) = v] - v \mathbb{E}_{(x,y) \sim \mathcal{D}} [h(x) | f(x) = v] \\
 = & \frac{1}{2} \left(2 \mathbb{E}_{(x,y) \sim \mathcal{D}} [h(x)y | f(x) = v] - 2v \mathbb{E}_{(x,y) \sim \mathcal{D}} [h(x) | f(x) = v] \right) \\
 \geq & \frac{1}{2} \left(2 \mathbb{E}_{(x,y) \sim \mathcal{D}} [h(x)y | f(x) = v] - 2v \mathbb{E}_{(x,y) \sim \mathcal{D}} [h(x) | f(x) = v] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [(h(x)-v)^2 | f(x) = v] \right) \\
 = & \frac{1}{2} \left(\mathbb{E}_{(x,y) \sim \mathcal{D}} [2h(x)y - h(x)^2 - v^2 | f(x) = v] \right) \\
 = & \frac{1}{2} \left(\mathbb{E}_{(x,y) \sim \mathcal{D}} [2h(x)y - h(x)^2 - 2vy + v^2 | f(x) = v] \right) \\
 = & \frac{1}{2} \left(\mathbb{E}_{(x,y) \sim \mathcal{D}} [(v-y)^2 - (h(x)-y)^2 | f(x) = v] \right) \\
 \geq & \frac{\alpha}{2}
 \end{aligned}$$

where the 3rd to last line follows from adding and subtracting v^2 and the fact that because f is calibrated, $v \mathbb{E}[y | f(x) = v] = v^2$.

In the reverse direction, we show that if a model f fails to be multicalibrated with respect to a class of functions \mathcal{H} , then it is possible to perform a simple update on one of the level-sets of f using a function $h \in \mathcal{H}$ that witnesses a failure of multicalibration that decreases squared error on that level-set

Lemma 9.1.2 Fix a model $f : \mathcal{X} \rightarrow \mathbb{R}$. Suppose for some $v \in R(f)$ there is an $h \in \mathcal{H}$ such that:

$$\mathbb{E}[h(x)(y-v) | f(x) = v] \geq \alpha$$

Let $h' = v + \eta h(x)$ for $\eta = \frac{\alpha}{\mathbb{E}[h(x)^2 | f(x) = v]}$. Then:

$$\mathbb{E}[(f(x) - y)^2 - (h'(x) - y)^2 | f(x) = v] \geq \frac{\alpha^2}{\mathbb{E}[h(x)^2 | f(x) = v]}$$

Proof 77 We calculate:

$$\begin{aligned}
 & \mathbb{E}[(f(x) - y)^2 - (h'(x) - y)^2 | f(x) = v] \\
 = & \mathbb{E}[(v - y)^2 - (v + \eta h(x) - y)^2 | f(x) = v] \\
 = & \mathbb{E}[v^2 - 2vy + y^2 - (v + \eta h(x))^2 + 2y(v + \eta h(x)) - y^2 | f(x) = v] \\
 = & \mathbb{E}[2y\eta h(x) - 2v\eta h(x) - \eta^2 h(x)^2 | f(x) = v] \\
 = & \mathbb{E}[2\eta h(x)(y - v) - \eta^2 h(x)^2 | f(x) = v] \\
 \geq & 2\eta\alpha - \eta^2 \mathbb{E}[h(x)^2 | f(x) = v] \\
 = & \frac{\alpha^2}{\mathbb{E}[h(x)^2 | f(x) = v]}
 \end{aligned}$$

Where the last line follows from the definition of η .

Note that there is an asymmetry in Lemma 9.1.1 and Lemma 9.1.2. Lemma 9.1.1 implies that if h has improved squared error compared to f on one of its level-sets, then h itself fails the multi-calibration condition on this levelset. On the other hand, Lemma 9.1.2 says that if h fails the multicalibration condition on some levelset v of f , then there is a function $h' = v + \eta h(x)$ that improves on the squared error of f on level-set v . We can remove this asymmetry by assuming that \mathcal{H} is closed under affine transformations

Definition 70 A class of functions \mathcal{H} is closed under affine transformations if for every $a, b \in \mathbb{R}$, if $h(x) \in \mathcal{H}$ then:

$$h'(x) \equiv ah(x) + b \in \mathcal{H}$$

Most natural classes of regression functions are closed under affine transformations: linear functions, polynomials of any fixed degree d , regression trees, etc.

For classes of functions \mathcal{H} that are closed under affine transformation, the relationship becomes symmetric:

Lemma 9.1.3 Suppose \mathcal{H} is closed under affine transformation. Fix a model $f: \mathcal{X} \rightarrow \mathbb{R}$ and a levelset $v \in R(f)$ of f . Then:

1. If f is calibrated and there exists an $h \in \mathcal{H}$ such that

$$\mathbb{E}[(f(x) - y)^2 - (h(x) - y)^2 | f(x) = v] \geq \alpha$$

then there exists an $h' \in \mathcal{H}$ such that:

$$\mathbb{E}[h'(x)(y - v) | f(x) = v] \geq \frac{\alpha}{2}$$

2. If there exists an $h \in \mathcal{H}$ such that:

$$\mathbb{E}[h(x)(y - v) | f(x) = v] \geq \alpha$$

then there exists an $h' \in \mathcal{H}$ such that:

$$\mathbb{E}[(f(x) - y)^2 - (h'(x) - y)^2 | f(x) = v] \geq \frac{\alpha^2}{\mathbb{E}[h(x)^2 | f(x) = v]}$$

Proof 78 The first part follows from Lemma 9.1.1 using $h' = h$. The second part follows from Lemma 9.1.2 using $h' = v + \eta h(x)$, where $h' \in \mathcal{H}$ by the assumption that \mathcal{G} is closed under affine transformations.

The equivalence between a failure of multicalibration with respect to \mathcal{H} and the ability for some $h \in \mathcal{H}$ to improve on the squared error of f on one of f 's level sets is useful for several reasons. First, it means that we can reduce the

problem of finding a model f that is multicalibrated over \mathcal{H} to the standard regression problem of finding models $h \in \mathcal{H}$ that minimize squared error over subsets of the distribution \mathcal{D} , which is a well studied problem for which we have very good algorithms for many classes \mathcal{H} . The second is that it will allow us to give a simple, intuitive characterization of what properties \mathcal{H} must have relative to a data distribution \mathcal{D} such that multicalibration with respect to \mathcal{H} implies Bayes optimal prediction with respect to \mathcal{H} . Importantly, since we only have to solve regression problems on subsets of the distribution \mathcal{D} — for which there is a fixed Bayes optimal predictor — this will make it easy for us to enunciate conditions under which multicalibration implies accuracy; this would be more difficult if we needed to solve regression problems on distributions with different conditional label distributions.

9.2 Algorithmically Reducing Multicalibration to Regression

In this section we give an algorithm for computing predictors that are multicalibrated with respect to a real-valued class of functions \mathcal{H} . We will be interested in infinite classes \mathcal{H} , so we will need to think about what kind of access we have to this class of functions. What we will assume is that we have access to an algorithm $A_{\mathcal{H}}$ that given access to a distribution \mathcal{D} solves the squared error regression problem on \mathcal{D} over \mathcal{H} .

Definition 71 $A_{\mathcal{H}}$ is a squared error regression oracle for a class of real valued functions \mathcal{H} if for every $\mathcal{D} \in \Delta\mathcal{Z}$, $A_{\mathcal{H}}(\mathcal{D})$ outputs a function $h \in \mathcal{H}$ such that:

$$h \in \arg \min_{h' \in \mathcal{H}} \mathbb{E}_{(x,y) \sim \mathcal{D}} [(h'(x) - y)^2]$$

A squared error regression oracle $A_{\mathcal{H}}$ is a very natural object: for example, if \mathcal{H} is the class of linear functions, then $A_{\mathcal{H}}$ simply solves a linear regression problem (which has a solution in closed form). Polynomial squared error regression problems can also be solved in closed form. Even for model classes (e.g. regression trees and neural networks) such that the corresponding squared error regression problem is not convex, we have very good heuristics for solving the problem. So assuming that we have a squared error regression oracle for a class \mathcal{H} is a very reasonable assumption. We now ask: if we have such an oracle, can we leverage it to learn a multi-calibrated predictor over \mathcal{H} ?

Algorithm 35 RegressionMulticalibrate($f, \alpha, A_{\mathcal{H}}, \mathcal{D}, B$)

Let $m = \frac{2B}{\alpha}$.
Let $f_0 = \text{Round}(f; m)$, $\text{err}_0 = \mathbb{E}_{(x,y) \sim \mathcal{D}}[(f_0(x) - y)^2]$, $\text{err}_{-1} = \infty$ and $t = 0$.
while $(\text{err}_{t-1} - \text{err}_t) \geq \frac{\alpha}{2B}$ **do**
 for each $v \in [1/m]$ **do**
 Let $\mathcal{D}_v^{t+1} = \mathcal{D} | f_t(x) = v$.
 Let $h_v^{t+1} = A_{\mathcal{H}}(\mathcal{D}_v^{t+1})$.
 Let:

$$\tilde{f}_{t+1}(x) = \sum_{v \in [1/m]} \mathbb{1}[f_t(x) = v] \cdot h_v^{t+1}(x) \quad f_{t+1} = \text{Round}(f_{t+1}, m)$$

Let $\text{err}_{t+1} = \mathbb{E}_{(x,y) \sim \mathcal{D}}[(f_{t+1}(x) - y)^2]$ and $t = t + 1$.
Output f_{t-1} .

Just as in our algorithm for multicalibration over groups \mathcal{G} (Algorithm 18), Algorithm 35 rounds its output to the discrete range $[1/m] = \{0, \frac{1}{m}, \dots, \frac{m-1}{m}, 1\}$. We recall that $\text{Round}(h, m)$ outputs the function:

$$\tilde{h}(x) = \min_{v \in [1/m]} |h(x) - v|$$

— i.e. the function that outputs the closest grid-point in $[1/m]$ to the function value $h_t(x)$.

Theorem 57 Fix any distribution $\mathcal{D} \in \Delta \mathcal{Z}$, any model $f : \mathcal{X} \rightarrow [0, 1]$, any $\alpha < 1$, any class of real valued functions \mathcal{H} that is closed under affine transformations, and a squared error regression oracle $A_{\mathcal{H}}$ for \mathcal{H} . For any bound $B > 0$ let:

$$\mathcal{H}_B = \{h \in \mathcal{H} : h(x)^2 \leq B\}$$

be the set of functions in \mathcal{H} with squared magnitude bounded by B . Then $\text{RegressionMulticalibrate}(f, \alpha, A_{\mathcal{H}}, \mathcal{D}, B)$ (Algorithm 35) halts after at most $T \leq \frac{2B}{\alpha}$ many iterations and outputs a model f_{T-1} such that f_{T-1} is α -approximately multicalibrated with respect to \mathcal{D} and \mathcal{H}_B .

Remark 9.2.1 Note the form of this theorem — we do not promise multicalibration at approximation parameter α for all of \mathcal{H} , but only for \mathcal{H}_B — i.e. those functions in \mathcal{H} satisfying a bound on their squared value. This is necessary, since \mathcal{H} is closed under affine transformations. To see this, note that if $\mathbb{E}[h(x)(y - v)] \geq \alpha$, then it must be that $\mathbb{E}[c \cdot h(x)(y - v)] \geq c \cdot \alpha$. Since $h'(x) = ch(x)$ is also in \mathcal{H} by assumption, approximate multicalibration bounds must always also be paired with a bound on the norm of the functions for which we promise those bounds.

Remark 9.2.2 The algorithm runs for at most $\frac{2B}{\alpha}$ iterations, and at each iteration needs to make $m + 1 = \frac{2B}{\alpha} + 1$ many calls to the squared error

regression oracle $A_{\mathcal{H}}$. Thus to obtain α -approximate multi-calibration with respect to \mathcal{H}_B , it suffices to make $\frac{4B^2}{\alpha^2} + \frac{2B}{\alpha}$ many oracle calls to a regression oracle for \mathcal{H} .

Proof 79 Since f_0 takes values in $[0, 1]$ and $y \in [0, 1]$, we have $\text{err}_0 \leq 1$, and by definition $\text{err}_T \geq 0$ for all T . By construction, if the algorithm has not halted at round t it must be that $\text{err}_t \leq \text{err}_{t-1} - \frac{\alpha}{2B}$, and so the algorithm must halt after at most $T \leq \frac{2B}{\alpha}$ many iterations to avoid a contradiction.

It remains to show that when the algorithm halts at round T , the model f_{T-1} that it outputs is α -approximately multi-calibrated with respect to \mathcal{D} and \mathcal{H}_B . We will show that if this is not the case, then $\text{err}_{T-1} - \text{err}_T > \frac{\alpha}{2B}$, which will be a contradiction to the halting criterion of the algorithm.

Suppose that f_{T-1} is not α -approximately multicalibrated with respect to \mathcal{D} and \mathcal{H}_B . This means there must be some $h \in \mathcal{H}_B$ such that:

$$\sum_{v \in [1/m]} \Pr_{(x,y) \sim \mathcal{D}} [f_{T-1}(x) = v] \left(\mathbb{E}_{(x,y) \sim \mathcal{D}} [h(x)(y-v) | f_{T-1}(x) = v] \right)^2 > \alpha$$

For each $v \in [1/m]$ define

$$\alpha_v = \Pr_{(x,y) \sim \mathcal{D}} [f_{T-1}(x) = v] \left(\mathbb{E}_{(x,y) \sim \mathcal{D}} [h(x)(y-v) | f_{T-1}(x) = v] \right)^2$$

So we have $\sum_{v \in [1/m]} \alpha_v > \alpha$.

Applying the 2nd part of Lemma 9.1.3 we learn that for each v , there must be some $h_v \in \mathcal{H}$ such that:

$$\begin{aligned} \mathbb{E}[(f_{T-1}(x) - y)^2 - (h_v(x) - y)^2 | f_{T-1}(x) = v] &> \frac{1}{\mathbb{E}[h(x)^2 | f_{T-1}(x) = v]} \cdot \frac{\alpha_v}{\Pr_{(x,y) \sim \mathcal{D}} [f_{T-1}(x) = v]} \\ &\geq \frac{1}{B} \frac{\alpha_v}{\Pr_{(x,y) \sim \mathcal{D}} [f_{T-1}(x) = v]} \end{aligned}$$

where the last inequality follows from the fact that $h \in \mathcal{H}_B$. Now we can compute:

$$\begin{aligned} &\mathbb{E}_{(x,y) \sim \mathcal{D}} [(f_{T-1}(x) - y)^2 - (\tilde{f}_T(x) - y)^2] \\ &= \sum_{v \in [1/m]} \Pr_{(x,y) \sim \mathcal{D}} [f_{T-1}(x) = v] \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f_{T-1}(x) - y)^2 - (\tilde{f}_T(x) - y)^2 | f_{T-1}(x) = v] \\ &= \sum_{v \in [1/m]} \Pr_{(x,y) \sim \mathcal{D}} [f_{T-1}(x) = v] \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f_{T-1}(x) - y)^2 - (h_v^T(x) - y)^2 | f_{T-1}(x) = v] \\ &\geq \sum_{v \in [1/m]} \Pr_{(x,y) \sim \mathcal{D}} [f_{T-1}(x) = v] \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f_{T-1}(x) - y)^2 - (h_v(x) - y)^2 | f_{T-1}(x) = v] \\ &\geq \sum_{v \in [1/m]} \frac{\alpha_v}{B} \\ &> \frac{\alpha}{B} \end{aligned}$$

Here the third line follows from the definition of \tilde{f}_T and the fourth line follows from the fact $h_v \in \mathcal{H}$ and that h_v^T minimizes squared error on \mathcal{D}_v^T amongst all $h \in \mathcal{H}$.

Finally we calculate:

$$\begin{aligned}
& err_{T-1} - err_T \\
&= \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f_{T-1}(x) - y)^2 - (f_T(x) - y)^2] \\
&= \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f_{T-1}(x) - y)^2 - (\tilde{f}_T(x) - y)^2] + \mathbb{E}_{(x,y) \sim \mathcal{D}} [(\tilde{f}_T(x) - y)^2 - (f_T(x) - y)^2] \\
&> \frac{\alpha}{B} + \mathbb{E}_{(x,y) \sim \mathcal{D}} [(\tilde{f}_T(x) - y)^2 - (f_T(x) - y)^2] \\
&> \frac{\alpha}{B} - \frac{1}{m} \\
&\geq \frac{\alpha}{2B}
\end{aligned}$$

where the last equality follows from the fact that $m \geq \frac{2B}{\alpha}$.

The 2nd inequality follows from the fact that for every pair (x, y) :

$$(\tilde{f}_T(x) - y)^2 - (f_T(x) - y)^2 \geq -\frac{1}{m}$$

To see this we consider two cases. Since $y \in [0, 1]$, if $\tilde{f}_T(x) > 1$ or $\tilde{f}_T(x) < 0$ then the Round operation decreases squared error and we have $(\tilde{f}_T(x) - y)^2 - (f_T(x) - y)^2 \geq 0$. In the remaining case we have $f_T(x) \in [0, 1]$ and $\Delta = \tilde{f}_T(x) - f_T(x)$ is such that $|\Delta| \leq \frac{1}{2m}$. In this case we can compute:

$$\begin{aligned}
(\tilde{f}_T(x) - y)^2 - (f_T(x) - y)^2 &= (f_T(x) + \Delta - y)^2 - (f_T(x) - y)^2 \\
&= 2\Delta(f_T(x) - y) + \Delta^2 \\
&\geq -2|\Delta| + \Delta^2 \\
&\geq -\frac{1}{m}
\end{aligned}$$

9.3 Weak Learning, Multicalibration, and Boosting

We now turn from multicalibration to “Boosting”. Our analysis of multicalibration algorithms has used squared error as a potential function — so we know that post-processing a model to make it multicalibrated does not harm accuracy (as measured by squared error). But when must multicalibration improve accuracy meaningfully? Can we find conditions on the class \mathcal{H} with respect to which we are multicalibrated such that multicalibration must imply *Bayes optimality*? That is what we’ll do now.

Definition 72 Fix a distribution $\mathcal{D} \in \Delta\mathcal{Z}$ and a class of functions \mathcal{H} . Let $f^*(x) = \mathbb{E}_{y \sim \mathcal{D}(x)}[y]$ denote the true conditional label expectation conditional on x . We say that \mathcal{H} satisfies the weak learner condition relative to \mathcal{D} if for every $S \subset \mathcal{X}$ with $\Pr_{x \sim \mathcal{D}_X}[x \in S] > 0$, if:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [(f^*(x) - y)^2 | x \in S] < \min_{c \in \mathbb{R}} \mathbb{E}_{(x,y) \sim \mathcal{D}} [(c - y)^2 | x \in S]$$

then there exists an $h \in \mathcal{H}$ such that:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [(h(x) - y)^2 | x \in S] < \min_{c \in \mathbb{R}} \mathbb{E}_{(x,y) \sim \mathcal{D}} [(c - y)^2 | x \in S]$$

First lets pause to interpret this condition and explain why it is “weak”. It is helpful to recall that $f^*(x)$ is the Bayes optimal predictor for squared error — it minimizes squared error over \mathcal{D} over the set of all possible functions (we proved this in Lemma 5.1.2.) The weak learning condition requires that for every restriction of \mathcal{D} to some subset $S \subset \mathcal{X}$ of its domain, if the Bayes optimal predictor performs better than a constant predictor in terms of squared error, then there must be some $h \in \mathcal{H}$ that also performs better than a constant predictor. This is a *weak* learning assumption because it might be that $f^*(x)$ performs *much* better than a constant predictor, but that the best $h \in \mathcal{H}$ performs only a little bit better than a constant predictor on S — this situation is still consistent with our assumption.

Nevertheless, we will show that the weak learning assumption is enough (together with our Algorithm 35 for multicalibration with respect to real valued functions \mathcal{H}) to boost the *weak learners* in \mathcal{H} to a *strong learner* f — i.e. a model f that performs as well as the optimal model f^* with respect to squared error. In fact, the weak learning condition on \mathcal{H} is both necessary and sufficient for multicalibration of f with respect to \mathcal{H} to imply Bayes optimality of f . Our “boosting algorithm” will simply be our multicalibration algorithm!

First we define what we mean when we say that multicalibration with respect to \mathcal{H} implies Bayes optimality. Note that $f^*(x)$ is multicalibrated with respect to any set of functions, so it is not enough to require that there *exist* Bayes optimal functions f that are multicalibrated with respect to \mathcal{H} . Instead, we have to require that *every* function that is multicalibrated with respect to \mathcal{H} is Bayes optimal:

Definition 73 Fix a distribution $\mathcal{D} \in \Delta\mathcal{Z}$. We say that multicalibration with respect to \mathcal{H} implies Bayes optimality over \mathcal{D} if for every $f : \mathcal{X} \rightarrow \mathbb{R}$ that is multicalibrated with respect to \mathcal{D} and \mathcal{H} , we have:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x) - y)^2] = \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f^*(x) - y)^2]$$

Where $f^*(x) = \mathbb{E}_{y \sim \mathcal{D}(x)}[y]$ is the function that has minimum squared error over the set of all functions.

Theorem 58 Fix a distribution $\mathcal{D} \in \Delta\mathcal{Z}$. Let \mathcal{H} be a class of functions that is

closed under affine transformation. Multicalibration with respect to \mathcal{H} implies Bayes optimality over \mathcal{D} if and only if \mathcal{H} satisfies the weak learner condition relative to \mathcal{D} .

Proof 80 To avoid measurability issues we assume that models f have a countable range (which is true in particular whenever \mathcal{X} is countable) — but this assumption can be avoided with more care.

First we show that if \mathcal{H} satisfies the weak learner condition relative to \mathcal{D} , then multicalibration with respect to \mathcal{H} implies Bayes optimality over \mathcal{D} . Suppose not. Then there exists a function f that is multicalibrated with respect to \mathcal{D} and \mathcal{H} , but is such that:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x) - y)^2] > \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f^*(x) - y)^2]$$

By linearity of expectation we have:

$$\sum_{v \in R(f)} \Pr[f(x) = v] \cdot \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x) - y)^2 - (f^*(x) - y)^2 | f(x) = v] > 0$$

In particular there must be some $v \in R(f)$ with $\Pr_{x \sim \mathcal{D}_x}[f(x) = v] > 0$ such that:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x) - y)^2 | f(x) = v] > \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f^*(x) - y)^2 | f(x) = v]$$

Let $S = \{x : f(x) = v\}$. Since f is calibrated, we know that:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [(v - y)^2 | x \in S] = \min_{c \in \mathbb{R}} \mathbb{E}_{(x,y) \sim \mathcal{D}} [(c - y)^2 | x \in S]$$

Thus by the weak learning assumption there must exist some $h \in \mathcal{H}$ such that:

$$\mathbb{E}[(v - y)^2 - (h(x) - y)^2 | x \in S] = \mathbb{E}[(f(x) - y)^2 - (h(x) - y)^2 | f(x) = v] > 0$$

By Lemma 9.1.3, there must therefore exist some $h' \in \mathcal{H}$ such that:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [h'(x)(y - v) | f(x) = v] > 0$$

implying that f is not multicalibrated with respect to \mathcal{D} and \mathcal{H} , a contradiction.

In the reverse direction, we show that for any \mathcal{H} that does not satisfy the weak learning condition with respect to \mathcal{D} , then multicalibration with respect to \mathcal{H} and \mathcal{D} does not imply Bayes optimality over \mathcal{D} . In particular, we exhibit a function f such that f is multicalibrated with respect to \mathcal{H} and \mathcal{D} , but such that:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x) - y)^2] > \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f^*(x) - y)^2]$$

Since \mathcal{H} does not satisfy the weak learning assumption over \mathcal{D} , there must exist some set $S \subseteq \mathcal{X}$ with $\Pr[x \in S] > 0$ such that

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [(f^*(x) - y)^2 | x \in S] < \min_{c \in \mathbb{R}} \mathbb{E}_{(x,y) \sim \mathcal{D}} [(c - y)^2 | x \in S]$$

but for every $h \in \mathcal{H}$:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [(h(x) - y)^2 | x \in S] \geq \min_{c \in \mathbb{R}} \mathbb{E}_{(x,y) \sim \mathcal{D}} [(c - y)^2 | x \in S]$$

Let $c(S) = \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | x \in S]$. We define $f(x)$ as follows:

$$f(x) = \begin{cases} f^*(x) & x \notin S \\ c(S) & x \in S \end{cases}$$

We can calculate that:

$$\begin{aligned} & \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x) - y)^2] \\ = & \Pr_{(x,y) \sim \mathcal{D}} [x \in S] \mathbb{E}_{(x,y) \sim \mathcal{D}} [(c(S) - y)^2 | x \in S] + \Pr_{(x,y) \sim \mathcal{D}} [x \notin S] \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f^*(x) - y)^2 | x \notin S] \\ > & \Pr_{(x,y) \sim \mathcal{D}} [x \in S] \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f^*(x) - y)^2 | x \in S] + \Pr_{(x,y) \sim \mathcal{D}} [x \notin S] \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f^*(x) - y)^2 | x \notin S] \\ = & \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f^*(x) - y)^2] \end{aligned}$$

In other words, f is not Bayes optimal. So if we can demonstrate that f is multicalibrated with respect to \mathcal{H} and \mathcal{D} we are done. Suppose otherwise. Then there exists some $h \in \mathcal{H}$ and some $v \in R(f)$ such that

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [h(x)(y - v) | f(x) = v] > 0$$

By Lemma 9.1.3, there exists some $h' \in \mathcal{H}$ such that:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [(h'(x) - y)^2 | f(x) = v] < \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x) - y)^2 | f(x) = v]$$

We first observe that it must be that $v = c(S)$. If this were not the case, by definition of f we would have that:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [(h'(x) - y)^2 | f(x) = v] < \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f^*(x) - y)^2 | f(x) = v]$$

which would contradict the Bayes optimality of f^* . Having established that $v = c(S)$ we can calculate:

$$\begin{aligned} & \mathbb{E}_{(x,y) \sim \mathcal{D}} [(h'(x) - y)^2 | f(x) = c(S)] \\ = & \Pr_{(x,y) \sim \mathcal{D}} [x \in S] \mathbb{E}_{(x,y) \sim \mathcal{D}} [(h'(x) - y)^2 | x \in S] + \\ & \Pr_{(x,y) \sim \mathcal{D}} [x \notin S, f(x) = c(S)] \mathbb{E}_{(x,y) \sim \mathcal{D}} [(h'(x) - y)^2 | x \notin S, f(x) = c(S)] \\ \geq & \Pr_{(x,y) \sim \mathcal{D}} [x \in S] \mathbb{E}_{(x,y) \sim \mathcal{D}} [(h'(x) - y)^2 | x \in S] + \\ & \Pr_{(x,y) \sim \mathcal{D}} [x \notin S, f(x) = c(S)] \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x) - y)^2 | x \notin S, f(x) = c(S)] \end{aligned}$$

where in the last inequality we have used the fact that by definition, $f(x) = f^*(x)$ for all $x \notin S$, and so is pointwise Bayes optimal for all $x \notin S$.

Hence the only way we can have $\mathbb{E}_{(x,y) \sim \mathcal{D}}[(h'(x) - y)^2 | f(x) = c(S)] < \mathbb{E}_{(x,y) \sim \mathcal{D}}[(f(x) - y)^2 | f(x) = c(S)]$ is if:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [(h'(x) - y)^2 | x \in S] < \mathbb{E}_{(x,y) \sim \mathcal{D}} [(c(S) - y)^2 | x \in S]$$

But this contradicts our assumption that \mathcal{H} violates the weak learning condition on S , which completes the proof.

Theorem 58 characterizes when *exact* multicalibration with respect to \mathcal{H} implies *exact* Bayes optimality and vice versa. But our algorithm 35 only converges to approximate multi-calibration over a set of functions \mathcal{H} . What can we say about its convergence to approximate Bayes optimality when \mathcal{H} satisfies the weak learning condition? To answer this question we'll need a quantitative version of our weak learning condition.

Definition 74 Fix a distribution $\mathcal{D} \in \Delta \mathcal{Z}$ and a class of functions \mathcal{H} . Let $f^*(x) = \mathbb{E}_{y \sim \mathcal{D}(x)}[y]$ denote the true conditional label expectation conditional on x . We say that \mathcal{H} satisfies the γ -weak learner condition relative to \mathcal{D} if for every $S \subset \mathcal{X}$ with $\Pr_{x \sim \mathcal{D}_x}[x \in S] > 0$, if:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [(f^*(x) - y)^2 | x \in S] < \min_{c \in \mathbb{R}} \mathbb{E}_{(x,y) \sim \mathcal{D}} [(c - y)^2 | x \in S] - \gamma$$

then there exists an $h \in \mathcal{H}$ such that:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [(h(x) - y)^2 | x \in S] < \min_{c \in \mathbb{R}} \mathbb{E}_{(x,y) \sim \mathcal{D}} [(c - y)^2 | x \in S] - \gamma$$

Definition 74 approaches Definition 72 as $\gamma \rightarrow 0$. It says that when the Bayes optimal predictor improves over a constant predictor on set S by at least some margin γ , then there is some $h \in \mathcal{H}$ that does so as well. On the one hand, it is weaker than Definition 72 in that it does not require anything of \mathcal{H} if the Bayes optimal predictor improves over a constant prediction by less than γ . On the other hand, it is stronger, in that it requires that some $h \in \mathcal{H}$ improve over a constant predictor on \mathcal{H} by margin γ (rather than just infinitesimally) whenever doing so is possible.

Since the γ -weak learning condition does not make any requirements on \mathcal{H} on sets for which $f^*(x)$ improves over a constant predictor by less than γ , the best we can hope to prove under this assumption is γ -approximate Bayes optimality, which is what we do next.

Theorem 59 Fix any distribution $\mathcal{D} \in \Delta \mathcal{Z}$, any model $f : \mathcal{X} \rightarrow [0, 1]$, any $\gamma > 0$, any class of real valued functions \mathcal{H} that satisfies the γ -weak learner condition relative to \mathcal{D} , and a squared error regression oracle $A_{\mathcal{H}}$ for \mathcal{H} . Let $\alpha = \gamma$ and $B = 1/\gamma$ (or any pair such that $\alpha/B = \gamma^2$). Then

$\text{RegressionMulticalibrate}(f, \alpha, A_{\mathcal{H}}, \mathcal{D}, B)$ halts after at most $T \leq \frac{2}{\gamma^2}$ many iterations and outputs a model f_{T-1} such that f_{T-1} is 2γ -approximately Bayes optimal over \mathcal{D} :

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [(f_{T-1}(x) - y)^2] \leq \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f^*(x) - y)^2] + 2\gamma$$

where $f^*(x) = \mathbb{E}_{(x,y) \sim \mathcal{D}}[y]$ is the function that minimizes squared error over \mathcal{D} .

Proof 81 At each round t before the algorithm halts, we have by construction that $\text{err}_t \leq \text{err}_{t-1} - \frac{\alpha}{2B}$, and since the squared error of f_0 is at most 1, and squared error is non-negative, we must have $T \leq \frac{2B}{\alpha} = \frac{2}{\gamma^2}$.

Now suppose the algorithm halts at round T and outputs f_{T-1} . It must be that $\text{err}_T > \text{err}_{T-1} - \frac{\gamma^2}{2}$. Suppose also that f_{T-1} is not 2γ -approximately Bayes optimal:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [(f_{T-1}(x) - y)^2 - (f^*(x) - y)^2] > 2\gamma$$

We can write this condition as:

$$\sum_{v \in [1/m]} \Pr[f_{T-1}(x) = v] \cdot \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f_{T-1}(x) - y)^2 - (f^*(x) - y)^2 | f_{T-1}(x) = v] > 2\gamma$$

Define the set:

$$S = \{v \in [1/m] : \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f_{T-1}(x) - y)^2 - (f^*(x) - y)^2 | f_{T-1}(x) = v] \geq \gamma\}$$

to denote the set of values v in the range of f_{T-1} such that conditional on $f_{T-1}(x) = v$, f_{T-1} is at least γ -sub-optimal. Since we have both $y \in [0, 1]$ and $f_{T-1}(x) \in [0, 1]$, for every v we must have that $\mathbb{E}[(f_{T-1}(x) - y)^2 - (f^*(x) - y)^2 | f_{T-1}(x) = v] \leq 1$. Therefore we can bound:

$$\begin{aligned} 2\gamma &< \sum_{v \in [1/m]} \Pr[f_{T-1}(x) = v] \cdot \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f_{T-1}(x) - y)^2 - (f^*(x) - y)^2 | f_{T-1}(x) = v] \\ &\leq \Pr_{(x,y) \sim \mathcal{D}} [x \in S] + (1 - \Pr_{(x,y) \sim \mathcal{D}} [x \in S])\gamma \end{aligned}$$

Solving we learn that:

$$\Pr_{(x,y) \sim \mathcal{D}} [x \in S] \geq \frac{2\gamma - \gamma}{(1 - \gamma)} \geq 2\gamma - \gamma = \gamma$$

Now observe that by the fact that \mathcal{H} is assumed to satisfy the γ -weak learning assumption with respect to \mathcal{D} , at the final round T of the algorithm, for every $v \in S$ we have that h_v^T satisfies:

$$\mathbb{E}_{(x,y) \sim \mathcal{D}} [(f_{T-1}(x) - y)^2 - (h_v^T(x) - y)^2 | f_{T-1}(x) = v] \geq \gamma$$

Let $e\tilde{r}r_T = \mathbb{E}_{(x,y) \sim \mathcal{D}}[(\tilde{f}_T(x) - y)^2]$ Therefore we have:

$$\begin{aligned} err_{T-1} - e\tilde{r}r_T &= \sum_{v \in [1/m]} \Pr_{(x,y) \sim \mathcal{D}}[f_{T-1}(x) = v] \mathbb{E}_{(x,y) \sim \mathcal{D}}[(f_{T-1}(x) - y)^2 - (h_v^T(x) - y)^2 | f_{T-1}(x) = v] \\ &\geq \Pr_{(x,y) \sim \mathcal{D}}[f_{T-1}(x) \in S] \gamma \\ &\geq \gamma^2 \end{aligned}$$

We recall that $|e\tilde{r}r_T - err_T| \leq 1/m = \frac{\gamma^2}{2}$ and so we can conclude that

$$err_{T-1} - err_T \geq \frac{\gamma^2}{2}$$

which contradicts the fact that the algorithm halted at round T , completing the proof.

References and Further Reading

Kim et al. [2019] first studied *multi-accuracy* (what we call group conditional mean consistency in this book) for real valued functions, and gave a boosting like algorithm for obtaining it. Multicalibration with respect to real valued functions was first studied in Gopalan et al. [2022b] who gave an algorithm based on “split” and “merge” operations, related to boosting-by-branching-programs algorithms from the learning theory literature. Burhanpurkar et al. [2021] first ask the question what properties of a set of groups \mathcal{G} are sufficient to guarantee that multicalibration with respect to \mathcal{G} implies Bayes optimality — the answer they give (which is sufficient but not necessary) is that \mathcal{G} contains refinements of the levelsets of the optimal predictor f^* . This can be viewed as a “strong learning” assumption in comparison to our “weak learning” assumption. The main results from this chapter, including the multicalibration algorithm that operates as a reduction to squared error regression, and the characterization that multicalibration implies Bayes Optimality if and only if \mathcal{H} satisfies the weak learning condition comes from Globus-Harris et al. [2023].

10

Conformal Prediction

CONTENTS

10.1 Prediction Sets and Nonconformity Scores	179
---	-----

Thus far we have restricted our attention to regression problems (in which the label domain $\mathcal{Y} = \mathbb{R}$), and have focused on estimating distributional quantities of conditional label distributions, like means and quantiles. In this chapter, we introduce a much more general framework for uncertainty quantification that reduces a very general uncertainty quantification problem to the problem of one dimensional quantile estimation. As a result, we will be able to draw on our development of powerful quantile estimation techniques to give an analogously powerful set of results for a much more general problem.

10.1 Prediction Sets and Nonconformity Scores

Suppose we have a distribution $\mathcal{D} \in \Delta\mathcal{Z}$ (although we will also consider the sequential prediction setting in which there need not be any distribution). Our goal is to be able to produce *prediction sets* as a function of observed features x that are likely to contain the corresponding label y . More specifically, we want to be able to find a function $\mathcal{T} : \mathcal{X} \rightarrow 2^{\mathcal{Y}}$ mapping unlabelled examples x to *subsets* of labels $\mathcal{T}(x)$ that have the property that the true label is contained within $\mathcal{T}(x)$ with some specified level of confidence $1 - \delta$:

$$\Pr[y \in \mathcal{T}(x)] \approx 1 - \delta$$

We leave unspecified for now what distribution this probability is taken over, because we will consider a spectrum of guarantees of increasing strength, mirroring our treatment of mean and quantile estimation. For example, we can ask for marginal guarantees, group conditional guarantees, calibrated guarantees, or ask for guarantees that hold empirically on adversarially chosen sequences. Prediction sets can take different forms: when we are facing a regression problem ($\mathcal{Y} = \mathbb{R}$) it is natural (but not necessary) for a prediction set to take the form of an *interval*: $\mathcal{T}(x) = [a, b]$ for some $a < b \in \mathbb{R}$. On the

**FIGURE 10.1**

Images x about which we might have uncertainty about their labels y .

other hand, for a multiclass classification problem (when \mathcal{Y} is some unordered discrete set), prediction sets correspond to subsets of labels — e.g. we might have $\mathcal{T}(x) = \{\text{Blueberry Muffin, Chihuahua}\}$ for x representing images from Figure 10.1.

Prediction sets are a very attractive way to quantify uncertainty: their size represents a quantitative *degree* of uncertainty. For example, if $\mathcal{T}(x)$ is a singleton, this represents certainty at the specified $1 - \delta$ level in a particular point prediction. But the contents of the set also provides insight into *where* the uncertainty lies. For example in a classification problem, there might be a high degree of uncertainty in the specific label, but a well crafted prediction set might nevertheless tell us that our uncertainty is concentrated in a region that corresponds to the same downstream action. Say, in a computer vision setting, we might be unsure of the breed of dog in front of us—so $\mathcal{T}(x)$ contains half a dozen different labels, corresponding to different breeds—but despite this uncertainty in the specifics, this prediction set gives us a high degree of confidence in what action to take—apply the breaks.

11

Distribution Shift

CONTENTS

11.1	Likelihood Ratio Reweighting	181
11.2	Multicalibration under Distribution Shift	184
11.3	Why Calibration Under Distribution Shift is Useful	186
	References and Further Reading	190

Thus far we have studied prediction in two very different models:

1. In the *batch* or *distributional* setting we assume that we have sample access to a distribution \mathcal{D} which we can use to train a model, that we can then deploy; it has guarantees on new data drawn from the *same* distribution.
2. In the *sequential adversarial* setting we assume data arrives sequentially and can be worst-case/generated by an adversary. But in order to make progress we assume that we learn the true label after each prediction.

But what if we want the best of both worlds — to be able to train a model on data drawn from some distribution \mathcal{D} , but then deploy it *without test time labels* on new data drawn from some other process, and still have guarantees about our predictions?

Of course this is impossible in general, but we can say something about it if we make assumptions about how the data distribution might shift. Suppose that we get training data from some source distribution \mathcal{D}^s , and then evaluate our model on a test distribution \mathcal{D}^t . Can we give guarantees if we assume something about how \mathcal{D}^s and \mathcal{D}^t relate to one another?

11.1 Likelihood Ratio Reweighting

Our goal is to learn to make predictions about labels y from examples x . If we are going to learn about the relationship between x and y on \mathcal{D}^s and then

hope to do well on \mathcal{D}^t , then this relationship had better be similar on both distributions — in this chapter we will assume that it is the same.

Definition 75 *Two distributions $\mathcal{D}^s, \mathcal{D}^t \in \Delta\mathcal{Z}$ are said to have the same conditional label distributions if for every $x \in \mathcal{X}$, $\mathcal{D}_y^s(x) = \mathcal{D}_y^t(x)$. In other words the distributions differ only in their marginal distributions on features \mathcal{D}_x^s and \mathcal{D}_x^t .*

So, two distributions that have the same conditional label distributions differ in the relative frequency with which different feature vectors x appear, but agree on how labels are distributed conditional on features — so there is some fixed “truth” that we can hope to learn.

Definition 76 (Likelihood Ratios) *For each $x \in \mathcal{X}$ let $p^s(x) = \Pr_{\mathcal{D}_x^s}[x]$ and let $p^t(x) = \Pr_{\mathcal{D}_x^t}[x]$ denote the probability mass/density that the feature distributions \mathcal{D}_x^s and \mathcal{D}_x^t respectively put on x . The $s \rightarrow t$ likelihood ratio for a point x is:*

$$w_{s \rightarrow t}(x) = \frac{p^t(x)}{p^s(x)}$$

Remark 11.1.1 *Observe that:*

$$\frac{1}{w_{s \rightarrow t}(x)} = \frac{p^s(x)}{p^t(x)} = w_{t \rightarrow s}(x)$$

$s \rightarrow t$ likelihood ratios are useful because they allow us to relate expectations taken over \mathcal{D}^s to expectations taken over \mathcal{D}^t .

Lemma 11.1.1 *Suppose \mathcal{D}^s and \mathcal{D}^t have the same conditional label distribution $\mathcal{D}_y^s(x) = \mathcal{D}_y^t(x) = \mathcal{D}_y(x)$. Fix any $S \subseteq \mathcal{X}$. For any function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, we have:*

$$\Pr_{(x,y) \sim \mathcal{D}^s}[x \in S] \mathbb{E}_{(x,y) \sim \mathcal{D}^s}[w_{s \rightarrow t}(x) \cdot F(x, y) | x \in S] = \Pr_{(x,y) \sim \mathcal{D}^t}[x \in S] \mathbb{E}_{(x,y) \sim \mathcal{D}^t}[F(x, y) | x \in S]$$

Proof 82 *For simplicity assume the distribution over \mathcal{X} is discrete (otherwise repeat the derivation below with sums replaced by integrals). We have:*

$$\begin{aligned} & \Pr_{(x,y) \sim \mathcal{D}^s}[x \in S] \mathbb{E}_{(x,y) \sim \mathcal{D}^s}[w_{s \rightarrow t}(x) \cdot F(x, y) | x \in S] \\ &= \sum_{x \in S} p^s(x) \cdot w_{s \rightarrow t}(x) \cdot \mathbb{E}_{y \sim \mathcal{D}_y(x)}[F(x, y)] \\ &= \sum_{x \in S} p^s(x) \frac{p^t(x)}{p^s(x)} \cdot \mathbb{E}_{y \sim \mathcal{D}_y(x)}[F(x, y)] \\ &= \sum_{x \in S} p^t(x) \cdot \mathbb{E}_{y \sim \mathcal{D}_y(x)}[F(x, y)] \\ &= \Pr_{(x,y) \sim \mathcal{D}^t}[x \in S] \mathbb{E}_{(x,y) \sim \mathcal{D}^t}[F(x, y) | x \in S] \end{aligned}$$

Of course, even if we are explicitly given samples from \mathcal{D}^s and \mathcal{D}^t , we will not generally know the likelihood ratios $w_{s \rightarrow t}(x)$. A common approach is to attempt to learn a function h from some class \mathcal{H} that approximates them well. Since they are a function only of x , this can be done using only unlabelled examples from $\mathcal{D}_{\mathcal{X}}^t$. Suppose we attempt to approximate $w_{s \rightarrow t}(x)$ using a function h . How should we evaluate our approximation error?

Definition 77 Suppose \mathcal{D}^s and \mathcal{D}^t have the same conditional label distribution. For a function $h : \mathcal{X} \rightarrow \mathbb{R}$, we write:

$$e(h, w_{s \rightarrow t}) = \mathbb{E}_{x \sim \mathcal{D}^s} [|h(x) - w_{s \rightarrow t}(x)|]$$

Similarly, for any subset $S \subseteq \mathcal{X}$ of the feature space, we write:

$$e(h, w_{s \rightarrow t}, S) = \mathbb{E}_{x \sim \mathcal{D}^s} [|h(x) - w_{s \rightarrow t}(x)| \mid x \in S]$$

Remark 11.1.2 Observe that by the law of total probability, for any collection of sets $\{S_1, \dots, S_k\}$ that partition \mathcal{X} , we have that:

$$\sum_{i=1}^k \Pr_{(x,y) \sim \mathcal{D}^s} [x \in S_i] e(h, w_{s \rightarrow t}, S_i) = e(h, w_{s \rightarrow t})$$

The next lemma shows that if we can estimate $w_{s \rightarrow t}$ closely in total variation distance (as measured in expectation over the source distribution \mathcal{D}^s), then we can closely approximate expectations over \mathcal{D}^t .

Lemma 11.1.2 Suppose \mathcal{D}^s and \mathcal{D}^t have the same conditional label distribution. Fix any $S \subseteq \mathcal{X}$. For any function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$, and any function $h : \mathcal{X} \rightarrow \mathbb{R}$, we have:

$$\begin{aligned} & \left| \Pr_{(x,y) \sim \mathcal{D}^s} [x \in S] \mathbb{E}_{(x,y) \sim \mathcal{D}^s} [h(x) \cdot F(x,y) \mid x \in S] - \Pr_{(x,y) \sim \mathcal{D}^t} [x \in S] \mathbb{E}_{(x,y) \sim \mathcal{D}^t} [F(x,y) \mid x \in S] \right| \\ & \leq \Pr_{(x,y) \sim \mathcal{D}^s} [x \in S] \cdot \max_{(x,y) \in \mathcal{Z}} |F(x,y)| \cdot e(h, w_{s \rightarrow t}, S) \end{aligned}$$

Proof 83 we know from Lemma 11.1.1 that we can write:

$$\Pr_{(x,y) \sim \mathcal{D}^t} [x \in S] \mathbb{E}_{(x,y) \sim \mathcal{D}^t} [F(x,y) \mid x \in S] = \Pr_{(x,y) \sim \mathcal{D}^s} [x \in S] \mathbb{E}_{(x,y) \sim \mathcal{D}^s} [w_{s \rightarrow t}(x) \cdot F(x,y) \mid x \in S]$$

So, we can calculate:

$$\begin{aligned} & \left| \Pr_{(x,y) \sim \mathcal{D}^s} [x \in S] \mathbb{E}_{(x,y) \sim \mathcal{D}^s} [h(x) \cdot F(x,y) \mid x \in S] - \Pr_{(x,y) \sim \mathcal{D}^t} [x \in S] \mathbb{E}_{(x,y) \sim \mathcal{D}^t} [F(x,y) \mid x \in S] \right| \\ & = \left| \Pr_{(x,y) \sim \mathcal{D}^s} [x \in S] \mathbb{E}_{(x,y) \sim \mathcal{D}^s} [h(x) \cdot F(x,y) \mid x \in S] - \Pr_{(x,y) \sim \mathcal{D}^s} [x \in S] \mathbb{E}_{(x,y) \sim \mathcal{D}^s} [w_{s \rightarrow t}(x) F(x,y) \mid x \in S] \right| \\ & = \Pr_{(x,y) \sim \mathcal{D}^s} [x \in S] \left| \mathbb{E}_{(x,y) \sim \mathcal{D}^s} [F(x,y) \cdot (h(x) - w_{s \rightarrow t}(x)) \mid x \in S] \right| \\ & \leq \Pr_{(x,y) \sim \mathcal{D}^s} [x \in S] \cdot \max_{(x,y) \in \mathcal{Z}} |F(x,y)| \mathbb{E}_{(x,y) \sim \mathcal{D}^s} [|h(x) - w_{s \rightarrow t}(x)| \mid x \in S] \\ & = \Pr_{(x,y) \sim \mathcal{D}^s} [x \in S] \cdot \max_{(x,y) \in \mathcal{Z}} |F(x,y)| \cdot e(h, w_{s \rightarrow t}, S) \end{aligned}$$

11.2 Multicalibration under Distribution Shift

We'll now study how multi-calibration guarantees change under distribution shift, and how the relationship between the class of functions \mathcal{H} we are multicalibrated with respect to interacts with the likelihood ratios $w_{s \rightarrow t}(x)$ defined by the shift. It will be more convenient for us to work with an ℓ_1 notion of multicalibration (compared to the ℓ_2 notion we gave in Definition 69).

Definition 78 (L_1 Multicalibration For Real Valued Functions) Fix a distribution $\mathcal{D} \in \Delta \mathcal{Z}$ and a model $f : \mathcal{X} \rightarrow [0, 1]$. Let \mathcal{H} be an arbitrary collection of real valued functions $h : \mathcal{X} \rightarrow \mathbb{R}$. We say that f is α -approximately L_1 -multicalibrated with respect to \mathcal{D} and \mathcal{H} if for every $h \in \mathcal{H}$:

$$K_1(f, h, \mathcal{D}) = \sum_{v \in R(f)} \Pr_{(x, y) \sim \mathcal{D}} [f(x) = v] \left| \mathbb{E}_{(x, y) \sim \mathcal{D}} [h(x)(y - v) | f(x) = v] \right| \leq \alpha$$

We say that f is α -approximately L_1 -multicalibrated with respect to \mathcal{D} if:

$$K_1(f, \mathcal{D}) = \sum_{v \in R(f)} \Pr_{(x, y) \sim \mathcal{D}} [f(x) = v] \left| \mathbb{E}_{(x, y) \sim \mathcal{D}} [(y - v) | f(x) = v] \right| \leq \alpha$$

Recall that we know from Lemma 5.1.1 that $K_1(f, h, \mathcal{D}) \leq \sqrt{K_2(f, h, \mathcal{D})}$. Thus, we can use algorithm 35 — which guarantees that $K_2(f, h, \mathcal{D}) \leq \alpha'$ for all $h \in \mathcal{H}$ — to obtain α -approximate L_1 multicalibration by setting $\alpha' = \alpha^2$.

Theorem 60 Suppose \mathcal{D}^s and \mathcal{D}^t have the same conditional label distribution, and suppose f is α -approximately L_1 -multicalibrated with respect to \mathcal{D}^s and \mathcal{H} . Then f is also α -approximately L_1 -multicalibrated with respect to \mathcal{D}^t and $\mathcal{H}_{s \rightarrow t}$ where:

$$\mathcal{H}_{s \rightarrow t} = \left\{ \frac{h(x)}{w_{s \rightarrow t}(x)} : h(x) \in \mathcal{H} \right\}$$

Proof 84 Since f is α -approximately L_1 -multicalibrated with respect to \mathcal{D}^s

and \mathcal{H} , we have that for every $h \in \mathcal{H}$:

$$\begin{aligned}
\alpha &\geq K_1(f, h, \mathcal{D}^s) \\
&= \sum_{v \in R(f)} \Pr_{(x,y) \sim \mathcal{D}^s} [f(x) = v] \left| \mathbb{E}_{(x,y) \sim \mathcal{D}^s} [h(x)(y-v) | f(x) = v] \right| \\
&= \sum_{v \in R(f)} \left| \Pr_{(x,y) \sim \mathcal{D}^s} [f(x) = v] \mathbb{E}_{(x,y) \sim \mathcal{D}^s} [h(x)(y-v) | f(x) = v] \right| \\
&= \sum_{v \in R(f)} \left| \Pr_{(x,y) \sim \mathcal{D}^s} [f(x) = v] \mathbb{E}_{(x,y) \sim \mathcal{D}^s} \left[w_{s \rightarrow t}(x) \cdot \left(\frac{h(x)}{w_{s \rightarrow t}(x)} (y-v) \right) | f(x) = v \right] \right| \\
&= \sum_{v \in R(f)} \left| \Pr_{(x,y) \sim \mathcal{D}^t} [f(x) = v] \mathbb{E}_{(x,y) \sim \mathcal{D}^t} \left[\frac{h(x)}{w_{s \rightarrow t}(x)} (y-v) | f(x) = v \right] \right| \\
&= K_1 \left(f, \frac{h}{w_{s \rightarrow t}}, \mathcal{D}^t \right)
\end{aligned}$$

Here the second to last equality follows from applying Lemma 11.1.1 to each term:

$$\Pr_{(x,y) \sim \mathcal{D}^s} [f(x) = v] \mathbb{E}_{(x,y) \sim \mathcal{D}^s} \left[w_{s \rightarrow t}(x) \cdot \left(\frac{h(x)}{w_{s \rightarrow t}(x)} (y-v) \right) | f(x) = v \right]$$

using $S = \{x : f(x) = v\}$ and $F(x, y) = \frac{h(x)}{w_{s \rightarrow t}(x)} (y-v)$.

Corollary 11.2.1 Suppose \mathcal{D}^s and \mathcal{D}^t have the same conditional label distribution, and suppose f is α -approximately L_1 -multicalibrated with respect to \mathcal{D}^s and \mathcal{H} . Then if $w_{s \rightarrow t} \in \mathcal{H}$, f has at most α L_1 -calibration error on \mathcal{D}^t :

$$K_1(f, \mathcal{D}^t) \leq \alpha$$

Proof 85 We apply Theorem 60. Since by assumption $w_{s \rightarrow t}(x) \in \mathcal{H}$, we can choose $h = w_{s \rightarrow t}$ and find that:

$$\begin{aligned}
\alpha &\geq K_1 \left(f, \frac{h}{w_{s \rightarrow t}}, \mathcal{D}^t \right) \\
&= \sum_{v \in R(f)} \left| \Pr_{(x,y) \sim \mathcal{D}^t} [f(x) = v] \mathbb{E}_{(x,y) \sim \mathcal{D}^t} \left[\frac{h(x)}{w_{s \rightarrow t}(x)} (y-v) | f(x) = v \right] \right| \\
&= \sum_{v \in R(f)} \Pr_{(x,y) \sim \mathcal{D}^t} [f(x) = v] \left| \mathbb{E}_{(x,y) \sim \mathcal{D}^t} [(y-v) | f(x) = v] \right| \\
&= K_1(f, \mathcal{D}^t)
\end{aligned}$$

Similarly, if we are approximately multicalibrated on \mathcal{D}^s with respect to a class \mathcal{H} that contains a function h that is close in total variation distance to $w_{s \rightarrow t}$ on \mathcal{D}^s , then we remain approximately calibrated on \mathcal{D}^t .

Lemma 11.2.1 *Suppose \mathcal{D}^s and \mathcal{D}^t have the same conditional label distribution, and suppose $f : \mathcal{X} \rightarrow [0, 1]$ is α -approximately L_1 -multicalibrated with respect to \mathcal{D}^s and \mathcal{H} . Then f has L_1 -calibration error on \mathcal{D}^t at most:*

$$K_1(f, \mathcal{D}^t) \leq \alpha + \min_{h \in \mathcal{H}} e(h, w_{s \rightarrow t})$$

Proof 86 *Let $h^* = \arg \min_{h \in \mathcal{H}} e(h, w_{s \rightarrow t})$. Since f is α -approximately L_1 multicalibrated with respect to \mathcal{D}^s and \mathcal{H} we have:*

$$\begin{aligned} \alpha &\geq K_1(f, h^*, \mathcal{D}^s) \\ &= \sum_{v \in R(f)} \left| \Pr_{(x,y) \sim \mathcal{D}^s} [f(x) = v] \mathbb{E}_{(x,y) \sim \mathcal{D}^s} [h^*(x)(y - v) | f(x) = v] \right| \\ &\geq \sum_{v \in R(f)} \left| \Pr_{(x,y) \sim \mathcal{D}^t} [f(x) = v] \mathbb{E}_{(x,y) \sim \mathcal{D}^t} [(y - v) | f(x) = v] \right| \\ &\quad - \sum_{v \in R(f)} \Pr_{(x,y) \sim \mathcal{D}^s} [f(x) = v] \cdot e(h^*, w_{s \rightarrow t}, \{x : f(x) = v\}) \\ &= K_1(f, \mathcal{D}^t) - e(h^*, w_{s \rightarrow t}) \end{aligned}$$

where the inequality follows from Lemma 11.1.2 applied to each term:

$$\Pr_{(x,y) \sim \mathcal{D}^t} [f(x) = v] \mathbb{E}_{(x,y) \sim \mathcal{D}^t} [(y - v) | f(x) = v]$$

choosing $S_v = \{x : f(x) = v\}$, $F(x, y) = (y - v)$, and the fact that since $y, v \in [0, 1]$, $\max_y |y - v| \leq 1$. The final line follows from the observation that the collection $\{S_v\}_{v \in R(f)}$ forms a partition of \mathcal{X} .

11.3 Why Calibration Under Distribution Shift is Useful

On our training distribution we generally have samples of labeled data $(x, y) \sim \mathcal{D}^s$, and so we can empirically evaluate various quantities of interest. When it comes time to deploy a model, we may have unlabelled examples $x \sim \mathcal{D}_{\mathcal{X}}^t$ from the target distribution, but we may not have labelled examples. But if f is calibrated on \mathcal{D}^t , then there are certain things we can do with only unlabelled examples.

One very simple thing we can do is estimate the average value of the label.

Lemma 11.3.1 *Suppose f satisfies α -approximate L_1 calibration on \mathcal{D}^t : $K_1(f, \mathcal{D}^t) \leq \alpha$. Then:*

$$\left| \mathbb{E}_{x \sim \mathcal{D}_{\mathcal{X}}^t} [f(x)] - \mathbb{E}_{(x,y) \sim \mathcal{D}^t} [y] \right| \leq \alpha$$

Proof 87 Expanding the definition of K_1 we can write:

$$\begin{aligned}
\alpha &\geq K_1(f, \mathcal{D}^t) \\
&= \sum_{v \in R(f)} \Pr_{x \sim \mathcal{D}_{\mathcal{X}}^t} [f(x) = v] \left| \mathbb{E}_{(x,y) \sim \mathcal{D}^t} [(y - v) | f(x) = v] \right| \\
&= \sum_{v \in R(f)} \left| \Pr_{x \sim \mathcal{D}_{\mathcal{X}}^t} [f(x) = v] \mathbb{E}_{(x,y) \sim \mathcal{D}^t} [y | f(x) = v] - \Pr_{x \sim \mathcal{D}_{\mathcal{X}}^t} [f(x) = v] v \right| \\
&\geq \left| \sum_{v \in R(f)} \left(\Pr_{x \sim \mathcal{D}_{\mathcal{X}}^t} [f(x) = v] \mathbb{E}_{(x,y) \sim \mathcal{D}^t} [y | f(x) = v] - \Pr_{x \sim \mathcal{D}_{\mathcal{X}}^t} [f(x) = v] v \right) \right| \\
&= \left| \mathbb{E}_{(x,y) \sim \mathcal{D}^t} [y] - \mathbb{E}_{x \sim \mathcal{D}_{\mathcal{X}}^t} [f(x)] \right|
\end{aligned}$$

If our label space is binary $\mathcal{Y} = \{0, 1\}$, then we can go beyond this, and estimate the cost of *acting on any policy depending on the predictions of f* .

Definition 79 Fix an action space \mathcal{A} and a model $f : \mathcal{X} \rightarrow [0, 1]$. A policy of f is any mapping $\rho : [0, 1] \rightarrow \mathcal{A}$ that chooses an action $\rho(f(x)) \in \mathcal{A}$ as a function of the prediction $f(x)$.

We can evaluate the *cost* of a policy using a loss function:

Definition 80 Fixing an action space \mathcal{A} , a loss function $\ell : \mathcal{A} \times \{0, 1\} \rightarrow \mathbb{R}$ maps action/label pairs to a real valued loss. Given a distribution \mathcal{D} and a predictor $f : \mathcal{X} \rightarrow [0, 1]$, the expected cost of a policy ρ is:

$$\ell(\rho, f, \mathcal{D}) = \mathbb{E}_{(x,y) \sim \mathcal{D}} [\ell(\rho(f(x)), y)]$$

We can estimate the cost of any policy ρ if we have sample access to \mathcal{D} — but this requires samples both of x (to compute $\rho(f(x))$) and y (to plug into the second argument of $\ell(\cdot, \cdot)$). What if we only have sample access to unlabelled examples from $\mathcal{D}_{\mathcal{X}}$? Recall that $f(x)$ *purports* to estimate $\mathbb{E}_{\mathcal{D}_{\mathcal{Y}(x)}}[y]$ — i.e. the probability that $y = 1$ conditional on x . So we can attempt to estimate $\ell(\rho, f, \mathcal{D})$ taking this as a given, using only unlabelled examples:

Definition 81 Given an action space \mathcal{A} , a loss function $\ell : \mathcal{A} \times \mathcal{Y} \rightarrow \mathbb{R}$, a policy ρ , a predictor $f : \mathcal{X} \rightarrow \mathbb{R}$, and a feature distribution $\mathcal{D}_{\mathcal{X}}$, the f -estimated cost of ρ is:

$$\tilde{\ell}(\rho, f, \mathcal{D}_{\mathcal{X}}) = \mathbb{E}_{x \sim \mathcal{D}_{\mathcal{X}}} [f(x)\ell(\rho(f(x)), 1) + (1 - f(x))\ell(\rho(f(x)), 0)]$$

Observe that for the Bayes optimal predictor — $f^*(x) = \mathbb{E}_{\mathcal{D}_{\mathcal{Y}(x)}}[y]$ — that the f^* -estimated cost of ρ : $\tilde{\ell}(\rho, f^*, \mathcal{D}_{\mathcal{X}})$ is equal to its true expected cost: $\ell(\rho, f^*, \mathcal{D})$.

We now show that the same is true if f is not Bayes optimal, but merely calibrated.

Theorem 61 Fix an action space \mathcal{A} , a loss function $\ell : \mathcal{A} \times \{0, 1\} \rightarrow \mathbb{R}$, a policy ρ , a distribution \mathcal{D} , and a predictor $f : \mathcal{X} \rightarrow \mathbb{R}$, and a distribution \mathcal{D} . Let:

$$C = \max_{a \in \mathcal{A}} (\ell(a, 0) + \ell(a, 1))$$

If f is α -approximately L_1 -calibrated with respect to \mathcal{D} , then:

$$\left| \ell(\rho, f, \mathcal{D}) - \tilde{\ell}(\rho, f, \mathcal{D}_{\mathcal{X}}) \right| \leq C \cdot \alpha$$

Proof 88 Let:

$$k_v := \Pr_{x \sim \mathcal{D}_{\mathcal{X}}} [f(x) = v] \cdot \left| \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f(x) = v] - v \right|$$

Since f satisfies α -approximate L_1 calibration with respect to \mathcal{D} , we know that:

$$K_1(f, \mathcal{D}) = \sum_{v \in R(f)} k_v \leq \alpha$$

We can now calculate:

$$\begin{aligned} & \ell(\rho, f, \mathcal{D}) \\ &= \mathbb{E}_{(x,y) \sim \mathcal{D}} [\ell(\rho(f(x)), y)] \\ &= \sum_{v \in R(f)} \Pr_{x \sim \mathcal{D}_{\mathcal{X}}} [f(x) = v] \mathbb{E}_{(x,y) \sim \mathcal{D}} [\ell(\rho(f(x)), y) | f(x) = v] \\ &= \sum_{v \in R(f)} \Pr_{x \sim \mathcal{D}_{\mathcal{X}}} [f(x) = v] \left(\ell(\rho(v), 1) \Pr_{(x,y) \sim \mathcal{D}} [y = 1 | f(x) = v] + \ell(\rho(v), 0) \Pr_{(x,y) \sim \mathcal{D}} [y = 0 | f(x) = v] \right) \\ &= \sum_{a \in \mathcal{A}} \ell(a, 1) \sum_{v: \rho(v)=a} \Pr_{x \sim \mathcal{D}_{\mathcal{X}}} [f(x) = v] \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f(x) = v] + \\ & \quad \sum_{a \in \mathcal{A}} \ell(a, 0) \sum_{v: \rho(v)=a} \Pr_{x \sim \mathcal{D}_{\mathcal{X}}} [f(x) = v] (1 - \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | f(x) = v]) \\ &\leq \sum_{a \in \mathcal{A}} \ell(a, 1) \sum_{v: \rho(v)=a} \left(\Pr_{x \sim \mathcal{D}_{\mathcal{X}}} [f(x) = v] v + k_v \right) + \\ & \quad \sum_{a \in \mathcal{A}} \ell(a, 0) \sum_{v: \rho(v)=a} \left(\Pr_{x \sim \mathcal{D}_{\mathcal{X}}} [f(x) = v] (1 - v) + k_v \right) \\ &= \sum_{v \in R(f)} \Pr_{x \sim \mathcal{D}_{\mathcal{X}}} [f(x) = v] (v \ell(\rho(v), 1) + (1 - v) \ell(\rho(v), 0)) + \sum_{v \in R(f)} k_v (\ell(\rho(v), 1) + \ell(\rho(v), 0)) \\ &= \tilde{\ell}(\rho, f, \mathcal{D}_{\mathcal{X}}) + \sum_{v \in R(f)} k_v (\ell(\rho(v), 1) + \ell(\rho(v), 0)) \\ &\leq \tilde{\ell}(\rho, f, \mathcal{D}_{\mathcal{X}}) + C \cdot \alpha \end{aligned}$$

The other direction is identical.

A simple example of a policy and loss function arises in binary classification. Here, $\mathcal{Y} = \{0, 1\}$ and $\mathcal{A} = \{0, 1\}$: our goal is for each example x observed, to predict the true label by selecting $a \in \mathcal{A}$ such that $a = y$. The 0/1 loss function defined as $\ell^{0/1}(a, y) = \mathbb{1}[a \neq y]$ measures the frequency with which a given policy makes prediction mistakes.

For any calibrated predictor f (including the true conditional label expectation $f = f^* = \mathbb{E}_{y \sim \mathcal{D}_{\mathcal{Y}}(x)}[y]$), the following policy minimizes 0/1 loss among all policies defined as a function of f :

$$\rho^*(v) = \begin{cases} 1 & v \geq \frac{1}{2} \\ 0 & v < \frac{1}{2} \end{cases}$$

Lemma 11.3.2 Fix any distribution \mathcal{D} and any predictor $f : \mathcal{X} \rightarrow [0, 1]$ such that $K_1(f, \mathcal{D}) \leq \alpha$. Consider the policy ρ^* defined above. For any other policy $\rho : [0, 1] \rightarrow \{0, 1\}$, we have:

$$\ell^{0,1}(\rho^*, f, \mathcal{D}) \leq \ell^{0,1}(\rho, f, \mathcal{D}) + 2\alpha$$

Proof 89 Using Theorem 61, the fact that f satisfies α -approximate L_1 -calibration, and the fact that for all y $\ell^{0/1}(0, y) + \ell^{0/1}(1, y) = 1$, we can calculate:

$$\begin{aligned} \ell^{0,1}(\rho, f, \mathcal{D}) &\geq \tilde{\ell}^{0,1}(\rho, f, \mathcal{D}_{\mathcal{X}}) - \alpha \\ &= \sum_{v \in R(f)} \Pr_{x \sim \mathcal{D}_{\mathcal{X}}} [f(x) = v] (v \cdot \ell^{0,1}(\rho(v), 1) + (1-v)\ell^{0,1}(\rho(v), 0)) - \alpha \\ &\geq \sum_{v \in R(f)} \Pr_{x \sim \mathcal{D}_{\mathcal{X}}} [f(x) = v] (v \cdot \ell^{0,1}(\rho^*(v), 1) + (1-v)\ell^{0,1}(\rho^*(v), 0)) - \alpha \\ &= \tilde{\ell}^{0,1}(\rho^*, f, \mathcal{D}_{\mathcal{X}}) - \alpha \\ &\geq \ell^{0,1}(\rho^*, f, \mathcal{D}) - 2\alpha \end{aligned}$$

Here the first and last inequalities follow from Theorem 61. The middle inequality follows from the fact that pointwise (for each value v):

$$v \cdot \ell^{0,1}(\rho(v), 1) + (1-v)\ell^{0,1}(\rho(v), 0) = \begin{cases} 1-v & \rho(v) = 1 \\ v & \rho(v) = 0 \end{cases}$$

is minimized by setting $\rho(v) = 1$ when $v \geq \frac{1}{2}$ and $\rho(v) = 0$ when $v < \frac{1}{2}$, which is what the policy $\rho^*(v)$ does.

So if f is calibrated on \mathcal{D} , the not only is ρ^* the optimal post-processing of f to minimize classification error on \mathcal{D} , but we can estimate the classification error of $\rho^*(f(x))$ on \mathcal{D} without the need for any labelled examples from \mathcal{D} — since we can estimate $\tilde{\ell}^{0,1}(\rho, f, \mathcal{D}_{\mathcal{X}})$ using only samples from $\mathcal{D}_{\mathcal{X}}$.

Thus if we have trained a model that is multicalibrated on some distribution \mathcal{D}^s with respect to some class of functions \mathcal{H} , then for any distribution

\mathcal{D}^t such that $w_{s \rightarrow t} \in \mathcal{H}$ (or is close in statistical distance to some $h \in \mathcal{H}$), we can correctly estimate the performance of our model on \mathcal{D}^t given access only to unlabelled data from \mathcal{D}^t , which is much more commonly available.

References and Further Reading

Multicalibration under distribution shift was studied by Kim et al. [2022], from which this chapter primarily draws — they call the phenomenon “universal adaptability”. Baek et al. [2022] suggest in a different context that predictors that are calibrated out of distribution can be used to evaluate the out of distribution performance using only unlabelled data.

12

Sufficient Statistics for Optimization

CONTENTS

12.1	Omnipredictors: Sufficient Statistics for Unconstrained Optimization	191
12.2	Sufficient Statistics for Constrained Optimization	197
12.2.1	Convex Optimization	198
12.2.2	f -estimated Optimization	199
12.2.3	Solving Optimization Problems Without Labelled Data	200
	References and Further Reading	202

In Chapter 11 we started studying the problem of choosing *actions* $a \in \mathcal{A}$ to minimize the expectation of a loss function $\ell : \mathcal{A} \times \mathcal{Y} \rightarrow \mathbb{R}$ using some policy $\rho : [0, 1] \rightarrow \mathcal{A}$ that we evaluate as a function of a predictor $f : \mathcal{X} \rightarrow [0, 1]$. In the case of binary labels $\mathcal{Y} = \{0, 1\}$, we saw that if f is calibrated on \mathcal{D} , then for any such policy, we can accurately estimate the loss of a policy $\rho(f(x))$ using only unlabeled data:

$$\ell(\rho, f, \mathcal{D}) \approx \tilde{\ell}(\rho, f, \mathcal{D}_{\mathcal{X}})$$

by “pretending” that $\Pr[y|f(x) = v] = v$ — so we can choose the optimal policy $\rho^*(f(x))$ under this fiction, and know that it performs as well as any other policy that is a function only of $f(x)$.

How strong is this guarantee? It depends. If $f(x) = f^*(x) = \mathbb{E}_{(x,y) \sim \mathcal{D}}[y|x]$ is the true conditional label expectation, then this guarantee means that the optimal policy ρ^* that is a function of $f(x)$ is as good as any other policy, regardless of what information about x it uses. On the other hand, if $f(x) = \mathbb{E}_{(x,y) \sim \mathcal{D}}[y]$ is simply the (calibrated) constant function, then policies $\rho(f(x))$ must also be constant functions, and so have necessarily very weak performance guarantees. In this chapter we ask under what conditions on f we can compare the performance of policies $\rho(f(x))$ to the performance of policies $h(x)$ that can depend in other ways on the features x . We assume throughout this chapter that the label space is binary: $\mathcal{Y} = \{0, 1\}$.

12.1 Omnipredictors: Sufficient Statistics for Unconstrained Optimization

We recall several important definitions from Chapter 11.

Definition 82 Fix an action space \mathcal{A} and a model $f : \mathcal{X} \rightarrow [0, 1]$. A policy of f is any mapping $\rho : [0, 1] \rightarrow \mathcal{A}$ that chooses an action $\rho(f(x)) \in \mathcal{A}$ as a function of the prediction $f(x)$.

We can evaluate the *cost* of a policy using a loss function:

Definition 83 Fixing an action space \mathcal{A} , a loss function $\ell : \mathcal{A} \times \{0, 1\} \rightarrow \mathbb{R}$ maps action/label pairs to a real valued loss. Given a distribution \mathcal{D} and a predictor $f : \mathcal{X} \rightarrow [0, 1]$, the expected cost of a policy ρ is:

$$\ell(\rho, f, \mathcal{D}) = \mathbb{E}_{(x,y) \sim \mathcal{D}} [\ell(\rho(f(x)), y)]$$

To compute the loss ℓ of a policy we need access to labeled examples $(x, y) \sim \mathcal{D}$. But we can estimate the loss of a policy using only unlabelled examples together with a model f if we “pretend” that $f(x)$ actually encodes the true conditional label expectation $\mathbb{E}[y|x]$:

Definition 84 Given an action space \mathcal{A} , a loss function $\ell : \mathcal{A} \times \mathcal{Y} \rightarrow \mathbb{R}$, a policy ρ , a predictor $f : \mathcal{X} \rightarrow \mathbb{R}$, and a feature distribution \mathcal{D}_X , the f -estimated cost of ρ is:

$$\tilde{\ell}(\rho, f, \mathcal{D}_X) = \mathbb{E}_{x \sim \mathcal{D}_X} [f(x)\ell(\rho(f(x)), 1) + (1 - f(x))\ell(\rho(f(x)), 0)]$$

Another nice property of the f -estimated loss is that we can find the policy that optimizes it without needing to know anything about the underlying distribution. Specifically, if we have a loss function ℓ in mind, we can choose the policy ρ_ℓ^* that pointwise optimizes the f -estimated cost $\tilde{\ell}$:

$$\rho_\ell^*(v) = \arg \min_{a \in \mathcal{A}} (v\ell(a, 1) + (1 - v)\ell(a, 0))$$

If f is calibrated, then the policy ρ_ℓ^* has the smallest expected loss (as measured by ℓ) of any policy that is a function of f . This statement generalizes what we proved in Lemma 11.3.2 in the special case of 0/1 loss and has essentially the same proof.

Lemma 12.1.1 Fix any distribution \mathcal{D} , loss function $\ell : \mathcal{A} \times \{0, 1\} \rightarrow \mathbb{R}$, and any predictor $f : \mathcal{X} \rightarrow [0, 1]$ such that $K_1(f, \mathcal{D}) \leq \alpha$. Let:

$$C = \max_{a \in \mathcal{A}} (\ell(a, 0) + \ell(a, 1))$$

Consider the policy ρ_ℓ^* defined above. For any other policy $\rho : [0, 1] \rightarrow \mathcal{A}$, we have:

$$\ell(\rho_\ell^*, f, \mathcal{D}) \leq \ell(\rho, f, \mathcal{D}) + 2C\alpha$$

Proof 90 Using Theorem 61 and the fact that f satisfies α -approximate L_1 -calibration, we can calculate:

$$\begin{aligned}
\ell(\rho, f, \mathcal{D}) &\geq \tilde{\ell}(\rho, f, \mathcal{D}_{\mathcal{X}}) - C\alpha \\
&= \sum_{v \in R(f)} \Pr_{x \sim \mathcal{D}_{\mathcal{X}}} [f(x) = v] (v \cdot \ell(\rho(v), 1) + (1 - v)\ell(\rho(v), 0)) - C\alpha \\
&\geq \sum_{v \in R(f)} \Pr_{x \sim \mathcal{D}_{\mathcal{X}}} [f(x) = v] (v \cdot \ell(\rho_{\ell}^*(v), 1) + (1 - v)\ell(\rho_{\ell}^*(v), 0)) - C\alpha \\
&= \tilde{\ell}(\rho_{\ell}^*, f, \mathcal{D}_{\mathcal{X}}) - C\alpha \\
&\geq \ell(\rho_{\ell}^*, f, \mathcal{D}) - 2C\alpha
\end{aligned}$$

Here the first and last inequalities follow from Theorem 61. The middle inequality follows from the fact that by definition, $\rho_{\ell}^*(v)$ is the minimizer of:

$$v \cdot \ell(\rho(v), 1) + (1 - v)\ell(\rho(v), 0)$$

Going forward, we consider the special case of $\mathcal{A} = [0, 1]$ and aim to show that if f is multicalibrated with respect to a class of real valued functions \mathcal{H} , then for any convex loss function ℓ , the policy ρ_{ℓ}^* has optimal loss not just compared to other policies ρ of f , but compared to any $h \in \mathcal{H}$. Note that functions $h : \mathcal{X} \rightarrow [0, 1]$ are functions of x directly, rather than functions of $f(x)$, and so Lemma 12.1.1 does not imply that that $\rho_{\ell}^*(f(x))$ has lower loss than $h(x)$. But Lemma 12.1.1 does point in the direction of our proof strategy: we will show that if f is (approximately) multicalibrated with respect to \mathcal{H} then in fact every $h \in \mathcal{H}$ is (almost) dominated by a policy $\rho(f(x))$. Thus for any loss functions satisfying the conditions of our theorem, we can do (almost) as well as any $h \in \mathcal{H}$ by playing the *optimal* policy for the f -estimated loss ρ_{ℓ}^* . As mentioned, our results will apply to any *convex* loss function:

Definition 85 A loss function $\ell : [0, 1] \times \{0, 1\} \rightarrow \mathbb{R}$ is convex in its first argument if for all $v, v', \alpha \in [0, 1]$ and for all $y \in \{0, 1\}$:

$$\ell(\alpha v + (1 - \alpha)v', y) \leq \alpha \cdot \ell(v, y) + (1 - \alpha) \cdot \ell(v', y)$$

A direct consequence of convexity that we will make use of is called Jensen's inequality:

Claim 12.1.1 (Jensen's Inequality) Fix any loss function $\ell : [0, 1] \times \{0, 1\} \rightarrow \mathbb{R}$ that is convex in its first argument. For any $y \in \{0, 1\}$ and for any distribution $\mathcal{P} \in \Delta[0, 1]$, we have:

$$\mathbb{E}_{v \sim \mathcal{P}} [\ell(v, y)] \geq \ell\left(\mathbb{E}_{v \sim \mathcal{P}} [v], y\right)$$

How closely we can relate the performance of a model $h(x)$ to the performance of a policy $\rho(f(x))$ will depend both on the multicalibration error that f has on the models in \mathcal{H} and on how much small errors in prediction are magnified by the loss function ℓ , which we will measure by its Lipschitz constant:

Definition 86 A loss function $\ell : [0, 1] \times \{0, 1\} \rightarrow \mathbb{R}$ is L -Lipschitz in its first argument if for all v, v' and for all $y \in \{0, 1\}$:

$$|\ell(v, y) - \ell(v', y)| \leq L \cdot |v - v'|$$

Finally we will prove a useful statement about multicalibration—if f is multicalibrated with respect to \mathcal{H} , then for any $h \in \mathcal{H}$, its conditional expectation (conditional on the value of f) doesn't change by very much if we additionally condition on the value of the label:

Lemma 12.1.2 Fix any distribution \mathcal{D} and class of real valued functions \mathcal{H} . Suppose that f is α -approximately L_1 multicalibrated with respect to \mathcal{D} and \mathcal{H} (as defined in Definition 78). and α -approximately L_1 -calibrated. Then for any $h \in \mathcal{H}$ and $v \in R(f)$:

$$\begin{aligned} \sum_{v \in R(f)} \Pr[f(x) = v]v(1-v) \left| \mathbb{E}_{(x,y) \sim \mathcal{D}}[h(x)|f(x) = v, y = 1] - \mathbb{E}_{(x,y) \sim \mathcal{D}}[h(x)|f(x) = v, y = 0] \right| \\ \leq 2\alpha \end{aligned}$$

Proof 91 Let:

$$k_v = \Pr_{(x,y) \sim \mathcal{D}}[f(x) = v] \left| v - \mathbb{E}_{(x,y) \sim \mathcal{D}}[y|f(x) = v] \right|$$

By hypothesis, we know that:

$$\alpha \geq K_1(f, \mathcal{D}) = \sum_{v \in R(f)} k_v.$$

We can now compute:

$$\begin{aligned}
\alpha &\geq K_1(f, h, \mathcal{D}) \\
&= \sum_{v \in R(f)} \Pr_{(x,y) \sim \mathcal{D}} [f(x) = v] \left| \mathbb{E}_{(x,y) \sim \mathcal{D}} [h(x)(y - v) | f(x) = v] \right| \\
&= \sum_{v \in R(f)} \Pr_{(x,y) \sim \mathcal{D}} [f(x) = v] \left| \Pr[y = 1 | f(x) = v] \mathbb{E}_{(x,y) \sim \mathcal{D}} [h(x) | f(x) = v, y = 1] (1 - v) \right. \\
&\quad \left. - \Pr[y = 0 | f(x) = v] \mathbb{E}_{(x,y) \sim \mathcal{D}} [h(x) | f(x) = v, y = 0] v \right| \\
&\geq \sum_{v \in R(f)} \Pr_{(x,y) \sim \mathcal{D}} [f(x) = v] \left| v \mathbb{E}_{(x,y) \sim \mathcal{D}} [h(x) | f(x) = v, y = 1] (1 - v) \right. \\
&\quad \left. - (1 - v) \mathbb{E}_{(x,y) \sim \mathcal{D}} [h(x) | f(x) = v, y = 0] v \right| - \sum_{v \in R(f)} k_v ((1 - v) + v) \\
&= \sum_{v \in R(f)} \Pr_{(x,y) \sim \mathcal{D}} [f(x) = v] \left| v(1 - v) \left(\mathbb{E}_{(x,y) \sim \mathcal{D}} [h(x) | f(x) = v, y = 1] \right. \right. \\
&\quad \left. \left. - \mathbb{E}_{(x,y) \sim \mathcal{D}} [h(x) | f(x) = v, y = 0] \right) \right| - \sum_{v \in R(f)} k_v \\
&\geq \sum_{v \in R(f)} \Pr_{(x,y) \sim \mathcal{D}} [f(x) = v] v(1 - v) \left| \mathbb{E}_{(x,y) \sim \mathcal{D}} [h(x) | f(x) = v, y = 1] \right. \\
&\quad \left. - \mathbb{E}_{(x,y) \sim \mathcal{D}} [h(x) | f(x) = v, y = 0] \right| - \alpha
\end{aligned}$$

With this lemma in hand, we can prove the main theorem of this section: that if f is multicalibrated with respect to \mathcal{H} , then for any convex Lipschitz loss function ℓ , the policy ρ_ℓ^* obtains loss nearly as good as the loss of the best $h \in \mathcal{H}$. Thus, once we train f , we can use it to optimize *any* such loss function ℓ and have performance guarantees relative to \mathcal{H} , rather than needing to solve a fresh optimization problem for each new loss function. The proof strategy is just as we have already laid out: show that the loss for any $h \in \mathcal{H}$ is comparable to the loss of some *policy* ρ of f , and therefore only higher than the loss of the *best* policy ρ_ℓ^* for ℓ .

Theorem 62 *Let $\ell : [0, 1] \times \{0, 1\} \rightarrow [0, 1]$ be a bounded loss function that is both convex and L -Lipschitz in its first argument. Suppose that f is α -approximately L_1 calibrated with respect to a distribution \mathcal{D} , and also with respect to \mathcal{D} and a class of real valued functions \mathcal{H} . That is, both that $K_1(f, \mathcal{D}) \leq \alpha$ and that for every $h \in \mathcal{H}$, $K_1(f, h, \mathcal{D}) \leq \alpha$. Let ρ_ℓ^* be the policy*

that optimizes the f -estimated loss $\tilde{\ell}$:

$$\rho_{\tilde{\ell}}^*(v) = \arg \min_{a \in \mathcal{A}} (v\ell(a, 1) + (1 - v)\ell(a, 0))$$

Then the loss of policy $\rho_{\tilde{\ell}}^*$ is almost as low as the loss of any $h \in \mathcal{H}$:

$$\ell(\rho_{\tilde{\ell}}^*, f, \mathcal{D}) \leq \mathbb{E}_{(x,y) \sim \mathcal{D}}[\ell(h(x), y)] + (4 + 4L)\alpha$$

Proof 92 *Let:*

$$k_v = \Pr_{(x,y) \sim \mathcal{D}}[f(x) = v] \left| v - \mathbb{E}_{(x,y) \sim \mathcal{D}}[y | f(x) = v] \right|$$

By hypothesis, we know that:

$$\alpha \geq K_1(f, \mathcal{D}) = \sum_{v \in R(f)} k_v.$$

Let $H(v, 1) = \mathbb{E}_{(x,y) \sim \mathcal{D}}[h(x) | f(x) = v, y = 1]$ and $H(v, 0) = \mathbb{E}_{(x,y) \sim \mathcal{D}}[h(x) | f(x) = v, y = 0]$. Using Jensen's inequality and the convexity of ℓ , we can calculate:

$$\begin{aligned} & \mathbb{E}_{(x,y) \sim \mathcal{D}}[\ell(h(x), y)] \\ &= \sum_{v \in R(f)} \Pr[f(x) = v] \left(\Pr[y = 1 | f(x) = v] \mathbb{E}_{(x,y) \sim \mathcal{D}}[\ell(h(x), 1) | f(x) = v, y = 1] \right. \\ & \quad \left. + (\Pr[y = 0 | f(x) = v]) \mathbb{E}_{(x,y) \sim \mathcal{D}}[\ell(h(x), 0) | f(x) = v, y = 0] \right) \\ &\geq \sum_{v \in R(f)} \Pr[f(x) = v] \left(\Pr[y = 1 | f(x) = v] \ell(H(v, 1), 1) + (\Pr[y = 0 | f(x) = v]) \ell(H(v, 0), 0) \right) \\ &\geq \sum_{v \in R(f)} \Pr[f(x) = v] \left(v\ell(H(v, 1), 1) + (1 - v)\ell(H(v, 0), 0) \right) - 2 \sum_{v \in R(f)} k_v \\ &\geq \sum_{v \in R(f)} \Pr[f(x) = v] \left(v\ell(H(v, 1), 1) + (1 - v)\ell(H(v, 0), 0) \right) - 2\alpha \end{aligned}$$

Now consider the policy ρ defined such that:

$$\rho(v) = \begin{cases} H(v, 1) & v \geq \frac{1}{2} \\ H(v, 0) & v < \frac{1}{2} \end{cases}$$

Lets compare the loss of this policy ρ with the loss of h . Continuing our derivation above we find:

$$\begin{aligned}
& \mathbb{E}_{(x,y) \sim \mathcal{D}} [\ell(h(x), y)] \\
& \geq \sum_{v \in R(f)} \Pr[f(x) = v] \left(v\ell(H(v, 1), 1) + (1-v)\ell(H(v, 0), 0) \right) - 2\alpha \\
& \geq \sum_{v < \frac{1}{2}} \Pr[f(x) = v] \left(v\ell(H(v, 0), 1) + (1-v)\ell(H(v, 0), 0) - vL|H(v, 0) - H(v, 1)| \right) + \\
& \quad \sum_{v \geq \frac{1}{2}} \Pr[f(x) = v] \left(v\ell(H(v, 1), 1) + (1-v)\ell(H(v, 1), 0) - (1-v)L|H(v, 0) - H(v, 1)| \right) - 2\alpha \\
& = \tilde{\ell}(\rho, f, \mathcal{D}_{\mathcal{X}}) - 2\alpha - L \sum_{v \in R(f)} \Pr[f(x) = v] \min(v, 1-v) |H(v, 0) - H(v, 1)| \\
& \geq \tilde{\ell}(\rho, f, \mathcal{D}_{\mathcal{X}}) - 2\alpha - 2L \sum_{v \in R(f)} \Pr[f(x) = v] v \cdot (1-v) |H(v, 0) - H(v, 1)| \\
& \geq \tilde{\ell}(\rho, f, \mathcal{D}_{\mathcal{X}}) - (2 + 4L)\alpha \\
& \geq \tilde{\ell}(\rho_{\ell}^*, f, \mathcal{D}_{\mathcal{X}}) - (2 + 4L)\alpha \\
& \geq \ell(\rho_{\ell}^*, f, \mathcal{D}) - (4 + 4L)\alpha
\end{aligned}$$

Here, in the 3rd to last line, we have applied Lemma 12.1.2, which tells us that:

$$\sum_{v \in R(f)} \Pr[f(x) = v] v(1-v) |H(v, 0) - H(v, 1)| \leq 2\alpha$$

In the second to last line we have used the fact that ρ_{ℓ}^* is the minimizer of $\tilde{\ell}(\rho, f, \mathcal{D}_{\mathcal{X}})$ among all policies ρ . In the final line we have applied Theorem 61 to relate the f -estimated loss $\tilde{\ell}$ to the true loss ℓ , using the fact that $K_1(f, \mathcal{D}) \leq \alpha$, and that $C = \max_{a \in \mathcal{A}} (\ell(a, 0) + \ell(a, 1))$ is at most 2 since we have assumed that ℓ takes values in $[0, 1]$.

12.2 Sufficient Statistics for Constrained Optimization

In Section 12.1 we showed that if f is multicalibrated with respect to \mathcal{H} , then for any (convex, Lipschitz) loss function ℓ , using the policy of f , ρ_{ℓ}^* that is optimal for minimizing ℓ is almost as good as using the best $h \in \mathcal{H}$, in terms of minimizing ℓ over the true data distribution. Note that this was an *unconstrained* optimization problem, in that there were no restrictions at all on what our policy $\rho^*(\ell)$ could look like. In this section, we consider constrained optimization problems. We continue to consider action spaces $\mathcal{A} = [0, 1]$ and binary label spaces $\mathcal{Y} = \{0, 1\}$.

Definition 87 Fix a collection of real valued functions \mathcal{H} of the form $h : \mathcal{X} \rightarrow [0, 1]$, a collection of group indicator functions \mathcal{G} of the form $g : \mathcal{X} \rightarrow \{0, 1\}$, and a scalar $C \in \mathbb{R}$. An $(\mathcal{H}, \mathcal{G}, C)$ -convex minimization problem with linear constraints is defined by:

1. An objective function $\ell : [0, 1] \times \{0, 1\} \rightarrow [-C, C]$ that is convex in its first argument,
2. A collection of k constraints j each defined by a loss function $\ell_j : [0, 1] \times \{0, 1\} \rightarrow [-C, C]$ that is affine in its first argument, a group indicator function $g_j \in \mathcal{G}$, and a subset of labels $S_j \subseteq \{0, 1\}$.

Together they define the following optimization problem:

$$\arg \min_{\mathcal{P} \in \Delta \mathcal{H}} \mathbb{E}_{h \sim \mathcal{P}, (x, y) \sim \mathcal{D}} [\ell(h(x), y)]$$

Subject to the constraint that for each $j \in [k]$:

$$\mathbb{E}_{h \sim \mathcal{P}, (x, y) \sim \mathcal{D}} [\ell_j(h(x), y) | g_j(x) = 1, y \in S_j] \leq 0$$

If there is any solution \mathcal{P} that satisfies all of the constraints, we say that the optimization problem is feasible. We write \mathcal{P}^* for the solution that minimizes the objective function while satisfying the constraints, and write $\text{OPT}(\mathcal{H}) = \mathbb{E}_{h \sim \mathcal{P}^*, (x, y) \sim \mathcal{D}} [\ell(h(x), y)]$ for the objective value of an optimal feasible solution.

12.2.1 Convex Optimization

We now review some facts about convex optimization with linear constraints.

Definition 88 Fix a $(\mathcal{H}, \mathcal{G}, C)$ -convex minimization problem with linear constraints, defined by $(\ell, \{(\ell_j, g_j, S_j)\}_{j=1}^k)$. The corresponding Lagrangian is the function $L : \mathcal{H} \times \mathbb{R}_{\geq 0}^k \rightarrow \mathbb{R}$ defined as:

$$L(\mathcal{P}, \lambda) = \mathbb{E}_{h \sim \mathcal{P}, (x, y) \sim \mathcal{D}} [\ell(h(x), y)] + \sum_{j=1}^k \lambda_j \mathbb{E}_{h \sim \mathcal{P}, (x, y) \sim \mathcal{D}} [\ell_j(h(x), y) | g_j(x) = 1, y \in S_j]$$

Definition 89 Fix a $(\mathcal{H}, \mathcal{G}, C)$ -convex minimization problem with linear constraints, and let $L : \mathcal{H} \times \mathbb{R}_{\geq 0}^k \rightarrow \mathbb{R}$ be its Lagrangian. We say that $\mathcal{P}^* \in \Delta \mathcal{H}$ and $\lambda^* \in \mathbb{R}^k$ are an optimal primal/dual pair for L if we have both that:

1.

$$\mathcal{P}^* \in \arg \min_{\mathcal{P} \in \Delta \mathcal{H}} L(\mathcal{P}, \lambda^*)$$

and

2.

$$\lambda^* \in \arg \max_{\lambda \in \mathbb{R}^k} L(\mathcal{P}^*, \lambda)$$

We'll state an important theorem in convex optimization here without proof:

Theorem 63 (Strong Duality and Complementary Slackness) *Fix a feasible $(\mathcal{H}, \mathcal{G}, C)$ -convex minimization problem with linear constraints, defined by $(\ell, \{(\ell_j, g_j, S_j)\}_{j=1}^k)$.*

For every optimal solution \mathcal{P}^ , there is a corresponding vector λ^* such that $(\mathcal{P}^*, \lambda^*)$ form an optimal primal/dual pair.*

Moreover every primal dual pair $(\mathcal{P}^, \lambda^*)$, satisfies:*

1. \mathcal{P}^* is a feasible, optimal solution to the optimization problem, and
2. For every constraint $j \in [k]$:

$$\lambda_j^* \cdot \left(\mathbb{E}_{h \sim \mathcal{P}^*, (x,y) \sim \mathcal{D}} [\ell_j(h(x), y) | g_j(x) = 1, y \in S_j] \right) = 0$$

The second condition is called “Complementary Slackness”

A simple corollary of Theorem 63 is that the Lagrangian of an optimization problem takes value OPT when evaluated at an optimal primal/dual pair.

Corollary 12.2.1 *Fix a feasible $(\mathcal{H}, \mathcal{G}, C)$ -convex minimization problem with linear constraints, defined by $(\ell, \{(\ell_j, g_j, S_j)\}_{j=1}^k)$. Let $L : \mathcal{H} \times \mathbb{R}_{\geq 0}^k \rightarrow \mathbb{R}$ be its corresponding Lagrangian, and let $(\mathcal{P}^*, \lambda^*)$ be an optimal primal/dual pair for L . Then:*

$$L(\mathcal{P}^*, \lambda^*) = OPT$$

Proof 93 *Using both parts of Theorem 63, we can compute:*

$$\begin{aligned} & L(\mathcal{P}^*, \lambda^*) \\ &= \mathbb{E}_{h \sim \mathcal{P}^*, (x,y) \sim \mathcal{D}} [\ell(h(x), y)] + \sum_{j=1}^k \lambda_j^* \mathbb{E}_{h \sim \mathcal{P}^*, (x,y) \sim \mathcal{D}} [\ell_j(h(x), y) | g_j(x) = 1, y \in S_j] \\ &= OPT + \sum_{j=1}^k \lambda_j^* \mathbb{E}_{h \sim \mathcal{P}^*, (x,y) \sim \mathcal{D}} [\ell_j(h(x), y) | g_j(x) = 1, y \in S_j] \\ &= OPT \end{aligned}$$

Here the second to last inequality follows from the fact that \mathcal{P}^ is an optimal solution to the optimization problem, and the last inequality follows from complementary slackness.*

12.2.2 f -estimated Optimization

The optimization problems we have defined (and their corresponding Lagrangians) are defined as expectations over both x and y — so in order to evaluate a solution \mathcal{P} (or to solve for one), we need access to labelled examples. Just as we did in Section 12.1 for unconstrained optimization, given a model $f : \mathcal{X} \rightarrow \mathbb{R}$ that purports to encode $f(x) = \mathbb{E}[y|x]$, we can define an f -estimated optimization problem whose definition only involves expectations taken over features $x \sim \mathcal{D}_{\mathcal{X}}$.

Definition 90 Fix an $(\mathcal{H}, \mathcal{G}, C)$ -convex minimization problem with objective ℓ and linear constraints is defined by $\{(\ell_j, g_j, S_j)\}_{j=1}^k$. Fix a model $f : \mathcal{X} \rightarrow [0, 1]$. The corresponding f -estimated optimization problem defined as:

$$\arg \min_{\mathcal{P} \in \Delta \mathcal{H}} \mathbb{E}_{h \sim \mathcal{P}, x \sim \mathcal{D}_{\mathcal{X}}} [f(x)\ell(h(x), 1) + (1 - f(x))\ell(h(x), 0)]$$

Subject to the constraints that for each $j \in [k]$ with $S = \{0, 1\}$:

$$\mathbb{E}_{h \sim \mathcal{P}, x \sim \mathcal{D}_{\mathcal{X}}} [f(x)\ell_j(h(x), 1) + (1 - f(x))\ell_j(h(x), 0)|g_j(x) = 1] \leq 0$$

for each $j \in [k]$ with $S = \{1\}$:

$$\mathbb{E}_{h \sim \mathcal{P}, x \sim \mathcal{D}_{\mathcal{X}}} [\ell_j(h(x), 1)|g_j(x) = 1] \leq 0$$

and for each $j \in [k]$ with $S = \{0\}$:

$$\mathbb{E}_{h \sim \mathcal{P}, x \sim \mathcal{D}_{\mathcal{X}}} [\ell_j(h(x), 0)|g_j(x) = 1] \leq 0$$

If there is any solution \mathcal{P} that satisfies all of the constraints, we say that the optimization problem is feasible. We write $\tilde{\mathcal{P}}^*$ for the solution that minimizes the objective function while satisfying the constraints, and write $O\tilde{P}T = \mathbb{E}_{h \sim \tilde{\mathcal{P}}^*, x \sim \mathcal{D}_{\mathcal{X}}} [f(x)\ell(h(x), 1) + (1 - f(x))\ell(h(x), 0)]$ for the objective value of an optimal feasible solution.

We can similarly define the f -estimated Lagrangian:

Definition 91 Fix an f -estimated $(\mathcal{H}, \mathcal{G}, C)$ -convex minimization problem with linear constraints, defined by $(\ell, \{(\ell_j, g_j, S_j)\}_{j=1}^k)$. Partition the constraints such that $C_0 = \{j \in [k] : S_j = \{0\}\}$, $C_1 = \{j \in [k] : S_j = \{1\}\}$, and $C_{01} = \{j \in [k] : S_j = \{0, 1\}\}$.

The corresponding f -estimated Lagrangian is the function $\tilde{L} : \mathcal{H} \times \mathbb{R}_{\geq 0}^k \rightarrow \mathbb{R}$ defined as:

$$\begin{aligned} \tilde{L}(\mathcal{P}, \lambda) &= \mathbb{E}_{h \sim \mathcal{P}, x \sim \mathcal{D}_{\mathcal{X}}} [f(x)\ell(h(x), 1) + (1 - f(x))\ell(h(x), 0)] \\ &+ \sum_{j \in C_0} \lambda_j \mathbb{E}_{h \sim \mathcal{P}, x \sim \mathcal{D}_{\mathcal{X}}} [\ell_j(h(x), 0)|g_j(x) = 1] + \sum_{j \in C_1} \lambda_j \mathbb{E}_{h \sim \mathcal{P}, x \sim \mathcal{D}_{\mathcal{X}}} [\ell_j(h(x), 1)|g_j(x) = 1] \\ &+ \sum_{j \in C_{01}} \lambda_j \mathbb{E}_{h \sim \mathcal{P}, x \sim \mathcal{D}_{\mathcal{X}}} [f(x)\ell_j(h(x), 1) + (1 - f(x))\ell_j(h(x), 0)|g_j(x) = 1] \end{aligned}$$

12.2.3 Solving Optimization Problems Without Labelled Data

Our goal is to derive a constrained optimization analogue of our results from Section 12.1, which were for *unconstrained* optimization. Namely, we would like to train a single model f using labelled data from \mathcal{D} , such that f is sufficient to solve a wide variety of downstream *constrained optimization* problems using only unlabelled data $x \sim \mathcal{D}_{\mathcal{X}}$ and minimal additional computation. The main idea will be to train a predictor f that is multicalibrated with respect to \mathcal{G} , \mathcal{H} , and their corresponding *product class*:

Definition 92 Fix two classes of functions \mathcal{G} and \mathcal{H} mapping features to real numbers. The product class is defined as:

$$\mathcal{G} \cdot \mathcal{H} = \{g(x) \cdot h(x) : g(x) \in \mathcal{G}, h(x) \in \mathcal{H}\}$$

We will argue that if we have such a predictor f , then if we want to solve some $(\mathcal{H}, \mathcal{G}, C)$ -convex minimization problem with linear constraints, it will be sufficient to solve its corresponding f -estimated variant in which \mathcal{H} has been replaced by \mathcal{H}_{all} , the set of *all* functions $f : \mathcal{X} \rightarrow [0, 1]$, which (as we will see) is a computationally easier task, and one that does not require a randomized solution.

Definition 93 We write $\mathcal{H}_{\text{all}} = \{f : \mathcal{X} \rightarrow [0, 1]\}$ for the set of all real-valued functions mapping features to the unit interval.

In particular, the optimal solution h to an f -estimated $(\mathcal{H}_{\text{all}}, \mathcal{G}, C)$ -optimization problem will be a *policy* in the sense that we can write it in the form $h(x) = \rho(f(x))$, that depends only on $f(x)$.

Lemma 12.2.1 Fix any model $f : \mathcal{X} \rightarrow [0, 1]$ and any f -estimated $(\mathcal{H}_{\text{all}}, \mathcal{G}, C)$ -convex optimization problem with linear constraints. Let $h \in \mathcal{H}_{\text{all}}$ be an optimal solution to the problem. Then $h(x)$ can be written as a policy of $f(x)$: $h(x) = \rho(f(x))$ for some $\rho : [0, 1] \rightarrow [0, 1]$.

Proof 94

Theorem 64 Fix any feasible $(\mathcal{H}, \mathcal{G}, C)$ -convex minimization problem with linear constraints, defined by $(\ell, \{(\ell_j, g_j, S_j)\}_{j=1}^k)$. Fix any $f : \mathcal{X} \rightarrow [0, 1]$ that is α -approximately L_1 -calibrated and L_1 -multicalibrated with respect to \mathcal{H} , \mathcal{G} , and $\mathcal{H} \cdot \mathcal{G}$. Let $\tilde{\mathcal{P}}^*$ be an optimal solution to the corresponding f -estimated $(\mathcal{H}_{\text{all}}, \mathcal{G}, C)$ -optimization problem. Then we have that $\tilde{\mathcal{P}}^*$ is approximately optimal according to the original objective function and approximately satisfies the original constraints:

BLAH

Proof 95 First we argue about the objective value. From Theorem 63, we know that there exists a $\tilde{\lambda}^*$ such that $(\tilde{\mathcal{P}}^*, \tilde{\lambda}^*)$ form an optimal primal/dual

pair for the corresponding f -estimated Lagrangian \tilde{L} . From Corollary 12.2.1 we know that $\tilde{L}(\tilde{\mathcal{P}}^*, \tilde{\lambda}^*) = \tilde{OPT}$. Similarly, let \mathcal{P}^* be an optimal solution to the original $(\mathcal{H}, \mathcal{G}, C)$ -optimization problem. We know from Theorem 63 that there exists a λ^* such that $(\mathcal{P}^*, \lambda^*)$ form an optimal primal/dual pair for the corresponding Lagrangian L , and from Corollary 12.2.1 that $L(\mathcal{P}^*, \lambda^*) = OPT$.

We also know from Theorem 61 since f is α -approximately calibrated with respect to \mathcal{H} that:

||

Thus we can calculate:

$$\begin{aligned} \tilde{OPT} &= L(\tilde{\mathcal{P}}^*, \tilde{\lambda}^*) \\ &\leq \end{aligned}$$

Can prove this if we need it

We recall a piece of notation from our earlier chapters: $\mu(g, \mathcal{D}) = \Pr_{(x,y) \sim \mathcal{D}}[g(x) = 1]$.

Lemma 12.2.2 Fix any distribution $\mathcal{D} \in \Delta \mathcal{Z}$, any class of group indicator functions \mathcal{G} containing functions $g : \mathcal{X} \rightarrow \{0, 1\}$ and any class of real valued functions \mathcal{H} containing functions $h : \mathcal{X} \rightarrow \mathbb{R}$. For each $g \in \mathcal{G}$ define the distribution $\mathcal{D}_g = \mathcal{D}|g(x) = 1$ be the conditional distribution conditional on $g(x) = 1$. Suppose a model $f : \mathcal{X} \rightarrow \mathbb{R}$ is α L_1 -multicalibrated with respect to \mathcal{D} and $\mathcal{G} \cdot \mathcal{H}$. Then for every $g \in \mathcal{G}$ and $h \in \mathcal{H}$:

$$K_1(f, h, \mathcal{D}_g) \leq \frac{\alpha}{\mu(g, \mathcal{D})}$$

Proof 96 By hypothesis we know that for every $h \in \mathcal{H}$ and $g \in \mathcal{G}$:

$$\begin{aligned} K_1(f, g \cdot h, \mathcal{D}) &= \sum_{v \in R(f)} \Pr[f(x) = v] \mathbb{E}[g(x)h(x)(y - v)|f(x) = v] \\ &\leq \alpha \end{aligned}$$

Fix any $g \in \mathcal{G}$ with $\mu(g, \mathcal{D}) > 0$. We can now calculate:

$$\begin{aligned} &K_1(f, h, \mathcal{D}_g) \\ &= \sum_{v \in R(f)} \Pr[f(x) = v|g(x) = 1] \mathbb{E}[h(x)(y - v)|g(x) = 1, f(x) = v] \end{aligned}$$

References and Further Reading

The fact that a model f that is multicalibrated with respect to a class of functions \mathcal{H} can be post-processed in such a way to be competitive with any $h \in \mathcal{H}$

as measured by any convex Lipschitz loss function was proven in Gopalan et al. [2022b], who called such models “omnipredictors”. Gopalan et al. [2022b] use a slightly different notion of calibration than we do (based on partitions of the feature space and covariance), but if a model satisfies our notion of multicalibration and is also calibrated, then it also satisfies the covariance based notion and vice versa. Gopalan et al. [2022a] give an incomparable omniprediction theorem — they show that group conditional mean consistency together with (marginal) calibration is sufficient to be competitive with any $h \in \mathcal{H}$ on any Lipschitz loss function ℓ (no longer requiring convexity of ℓ or full multicalibration) — but in general this requires group conditional mean consistency with respect to all *level sets* of functions in \mathcal{H} , rather than just with \mathcal{H} itself.



13

Ensembling, Model Multiplicity, and the Reference Class Problem

CONTENTS

13.1	Reference Classes and Model Multiplicity	205
13.2	Model Ensembling	206
13.3	Sample Complexity	211
	References and Further Reading	215

Suppose we have a prediction problem of some import: perhaps we are selling life insurance, and we want to predict the probability that particular customers will die in the next 12 months. We are in a familiar regression setting, in which we have some space of individuals \mathcal{X} and would like a model $f : \mathcal{X} \rightarrow [0, 1]$, where ideally $f(\text{Bob})$ should have the semantics that “ $f(\text{Bob})$ is the probability that Bob will die in the next 12 months”. But what does this mean? We are predicting a probability for a single event that will occur or not — there are to be no repeated trials for which we can measure an empirical frequency. If I propose a model f_1 that purports to assign individual probabilities to people like Bob, and you propose a different model f_2 , how are we to resolve which model is “better”?

13.1 Reference Classes and Model Multiplicity

Suppose we posit that there are true “individual probabilities” underlying reality — i.e. that there is in principle some number p_{Bob} that represents the probability that Bob will die in the next 12 months. This is after all the formalism that has underlied our studies so far: we have been modeling the world as if there is a distribution \mathcal{D} over labelled examples (x, y) , and for each individual x a conditional label distribution $\mathcal{D}_y(x)$. We still cannot get access to these individual probabilities through data. Nevertheless, we know that the function f^* encoding the true conditional label distribution

$f^*(x) = \mathbb{E}_{y \sim \mathcal{D}(x)}[y]$ minimizes the expected Brier score:

$$f^* \in \arg \min_{f: \mathcal{X} \rightarrow [0,1]} \mathbb{E}_{(x,y) \sim \mathcal{D}} [(f(x) - y)^2]$$

Hence if we have two models such that $B(f_1) < B(f_2)$, this falsifies the hypothesis that $f_2 = f^*$ — i.e. it cannot be the case that f_2 represents the true individual probabilities, and gives us an empirical (and practical!) justification for adopting model f_1 rather than model f_2 .

The “model multiplicity” problem refers to the worry that there may be multiple models f_1, f_2 that are equally accurate (such that $B(f_1) = B(f_2)$) that disagree in their predictions. In this case, accuracy gives us no basis on which to reject either model, and yet if $f_1(\text{Bob})$ is very different from $f_2(\text{Bob})$, what basis do we have to act on our predictions? Are we justified in denying Bob life insurance if it seems unprofitable according to the individual probability assigned by f_2 but seems profitable according to the individual probability assigned by f_1 ?

This can indeed be a problem if the models f are chosen to optimize accuracy in some fixed class. But as we will see, the situation cannot arise if the parties proposing their models are willing to update (and improve!) their models in the face of evidence that can be found in the data before them and in the competing models that are proposed! The updates needed will be of exactly the same simple “patch” form that we have studied when deriving algorithms for multicalibration and group conditional mean consistency.

13.2 Model Ensembling

Suppose we are given two models $f_1, f_2 : \mathcal{X} \rightarrow [0, 1]$. We will be interested in regions in which these models disagree substantially in their predictions. We will define “substantially” by an arbitrarily small discretization parameter ϵ :

Definition 94 *Two models f_1 and f_2 have an ϵ -disagreement on a point $x \in \mathcal{X}$ if $|f_1(x) - f_2(x)| > \epsilon$.*

Let $U_\epsilon(f_1, f_2)$ be the set of points on which f_1 and f_2 have an ϵ -disagreement:

$$U_\epsilon(f_1, f_2) = \{x : |f_1(x) - f_2(x)| > \epsilon\}$$

Informally, we will say that if f_1 and f_2 do not have an ϵ -disagreement on x that they agree on x . We will show a quantitative version of the following statement. It must be the case that *either*

1. f_1 and f_2 agree on almost all of their predictions, or
2. f_1 , or f_2 , or both can be proven from the data to violate a group

conditional mean consistency condition on a large set of points. In this case, the falsified model can be patched using our patch operations in a way that improves its accuracy.

The result is that there can be no substantial disagreements about individual probabilities by people who are willing to be convinced by the evidence of the data before them: models which disagree on a substantial fraction of their predictions witness for each other places in which their predictions are falsified by the data, and provide the means to correct (and improve) each other. Thus disagreements can be leveraged to produce improved models, and this process necessarily converges only when the models agree.

To formalize this, we start by partitioning the set of ϵ -disagreements $U_\epsilon(f_1, f_2)$ into two additional sets that will be important — the set of disagreements on which $f_1(x) > f_2(x)$, and the set of disagreements on which $f_1(x) < f_2(x)$.

Definition 95 Fix any two models $f_1, f_2 : \mathcal{X} \rightarrow [0, 1]$ and any $\epsilon > 0$. Define the sets:

$$U_\epsilon^>(f_1, f_2) = \{x \in U_\epsilon(f_1, f_2) : f_1(x) > f_2(x)\}$$

$$U_\epsilon^<(f_1, f_2) = \{x \in U_\epsilon(f_1, f_2) : f_1(x) < f_2(x)\}$$

Based on these sets, for $\bullet \in \{>, <\}$ and $i \in \{1, 2\}$ define the quantities:

$$v_*^\bullet = \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | x \in U_\epsilon^\bullet(f_1, f_2)] \quad v_i^\bullet = \mathbb{E}_{(x,y) \sim \mathcal{D}} [f_i(x) | x \in U_\epsilon^\bullet(f_1, f_2)]$$

Lemma 13.2.1 Fix any two models $f_1, f_2 : \mathcal{X} \rightarrow [0, 1]$ and any $\epsilon > 0$.

If the fraction of points on which f_1 and f_2 have an ϵ disagreement has mass $\mu(U_\epsilon(f_1, f_2)) = \alpha$ then for some $\bullet \in \{>, <\}$ some $i \in \{1, 2\}$, we have that:

$$\mu(U_\epsilon^\bullet(f_1, f_2)) \cdot (v_*^\bullet - v_i^\bullet)^2 \geq \frac{\alpha \epsilon^2}{8}$$

Proof 97 Since $U_\epsilon(f_1, f_2)$ can be written as the disjoint union:

$$U_\epsilon(f_1, f_2) = U_\epsilon^>(f_1, f_2) \cup U_\epsilon^<(f_1, f_2)$$

we must have that for at least one value of $\bullet \in \{>, <\}$ we have that:

$$\mu(U_\epsilon^\bullet(f_1, f_2)) \geq \frac{\alpha}{2}.$$

Since the points in $\mu(U_\epsilon^\bullet(f_1, f_2))$ are ϵ -separated, we must have that $|v_1^\bullet - v_2^\bullet| \geq \epsilon$. Therefore, for at least one of $i \in \{1, 2\}$ we must have that

$$|v_i^\bullet - v_*^\bullet| \geq \frac{\epsilon}{2}$$

Combining these two claims, we must have that:

$$\mu(U_\epsilon^\bullet(f_1, f_2)) \cdot (v_i^\bullet - v_*^\bullet)^2 \geq \frac{\alpha \epsilon^2}{8}$$

Lets consider the significance of this Lemma. Most basically, if we have two models f_1 and f_2 that disagree substantially, this lemma gives an easily constructable set ($U_\epsilon^>(f_1^{t_1}, f_2^{t_2})$ or $U_\epsilon^<(f_1^{t_1}, f_2^{t_2})$) that falsifies either the assertion that f_1 encodes true conditional label expectations or the assertion that f_2 does. And not only does it falsify that at least one of f_1 or f_2 are a “correct” model — it provides a directly actionable way to improve one of the models. Recall Lemma 6.1.1, which we proved when analyzing an algorithm for guaranteeing group conditional mean consistency, and we reproduce here:

Lemma 13.2.2 Fix any model $f_t : \mathcal{X} \rightarrow [0, 1]$ and group $g : \mathcal{X} \rightarrow \{0, 1\}$. Let

$$\Delta_t = \mathbb{E}_{(x,y) \sim \mathcal{D}} [y|g_t(x) = 1] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [f_t(x)|g_t(x) = 1]$$

and

$$f_{t+1} = h(x, f_t; g_t, \Delta_t)$$

where:

$$h(x, f; g, \Delta) = \begin{cases} f(x) + \Delta & g(x) = 1 \\ f(x) & \text{otherwise} \end{cases}$$

Then:

$$B(f_t) - B(f_{t+1}) = \mu(g_t) \cdot \Delta_t^2$$

Summarizing, whenever we have two models that have ϵ disagreements on an α -fraction of points, we can always constructively falsify at least one of the models, and update it to improve its Brier score by at least $O(\alpha\epsilon^2)$.

We put this all together in Algorithm 36 (Reconciler).

Algorithm 36 Reconcile($f_1, f_2, \alpha, \epsilon$)

Let $t = t_1 = t_2 = 0$ and $f_1^{t_1} = f_1, f_2^{t_2} = f_2$.

Let $m = \lceil \frac{2}{\sqrt{\alpha\epsilon}} \rceil$

while $\mu(U_\epsilon(f_1^{t_1}, f_2^{t_2})) \geq \alpha$ **do**

For each $\bullet \in \{>, <\}$ and $i \in \{1, 2\}$ Let:

$$v_*^\bullet = \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | x \in U_\epsilon^\bullet(f_1^{t_1}, f_2^{t_2})] \quad v_i^\bullet = \mathbb{E}_{(x,y) \sim \mathcal{D}} [f_i^{t_i}(x) | x \in U_\epsilon^\bullet(f_1^{t_1}, f_2^{t_2})]$$

Let:

$$(i_t, \bullet_t) = \arg \max_{i \in \{1,2\}, \bullet \in \{>, <\}} \mu(U_\epsilon^\bullet(f_1^{t_1}, f_2^{t_2})) \cdot (v_*^\bullet - v_i^\bullet)^2$$

Let:

$$g_t(x) = \begin{cases} 1 & x \in U_\epsilon^{\bullet_t}(f_1^{t_1}, f_2^{t_2}) \\ 0 & \text{otherwise} \end{cases}$$

Let:

$$\tilde{\Delta}_t = \mathbb{E}_{(x,y) \sim \mathcal{D}} [y | g_t(x) = 1] - \mathbb{E}_{(x,y) \sim \mathcal{D}} [f_i^{t_i}(x) | g_t(x) = 1]$$

$$\Delta_t = \text{Round}(\tilde{\Delta}_t; m)$$

Let: $f_i^{t_i+1}(x) = h(x, f_i^{t_i}, g_t, \Delta_t)$, $t_i = t_i + 1$, $t = t + 1$.

Output $(f_1^{t_1}, f_2^{t_2})$.

Theorem 65 For any pair of models $f_1, f_2 : \mathcal{X} \rightarrow [0, 1]$ and any $\alpha, \epsilon > 0$, Algorithm 36 (Reconcile) runs for $T = T_1 + T_2$ many rounds and outputs a pair of models $(f_1^{T_1}, f_2^{T_2})$ such that:

1. $T \leq (B(f_1) + B(f_2)) \cdot \frac{16}{\alpha\epsilon^2}$
2. $B(f_1^{T_1}) \leq B(f_1) - T_1 \cdot \frac{\alpha\epsilon^2}{16}$ and $B(f_2^{T_2}) \leq B(f_2) - T_2 \cdot \frac{\alpha\epsilon^2}{16}$
3. $\mu(U_\epsilon(f_1^{T_1}, f_2^{T_2})) < \alpha$.

Proof 98 By Lemma 13.2.1, for each round $t < T$ we must have that:

$$\arg \max_{i \in \{1,2\}, \bullet \in \{>, <\}, v \in [1/2\epsilon]} \mu(S(v, \bullet)) \cdot (v_*^\bullet - v_i^\bullet)^2 \geq \frac{\alpha\epsilon^2}{8}$$

Let $\tilde{f}_t^{t_i+1} = h(x, f_i^{t_i}, g_t, \tilde{\Delta}_t)$ — i.e. the update that would have resulted at round t had the algorithm used the unrounded measurement $\tilde{\Delta}_t$ rather than the rounded measurement Δ_t . By Lemma 13.2.2, we have that:

$$B(f_t^{t_i}) - B(\tilde{f}_t^{t_i+1}) \geq \frac{\alpha\epsilon^2}{8}$$

We can now compute

$$\begin{aligned} B(f_t^{t_i}) - B(f_t^{t_i+1}) &= (B(f_t^{t_i}) - B(\tilde{f}_t^{t_i+1})) - (B(f_t^{t_i+1}) - B(\tilde{f}_t^{t_i+1})) \\ &\geq \frac{\alpha\epsilon^2}{8} - (B(f_t^{t_i+1}) - B(\tilde{f}_t^{t_i+1})) \end{aligned}$$

So it remains to upper bound $(B(f_t^{t_i+1}) - B(\tilde{f}_t^{t_i+1}))$. Let $\hat{\Delta} = \tilde{\Delta}_t - \Delta_t$. We make several observations: First, $\tilde{f}_t^{t_i+1} = h(x, f_t^{t_i+1}, g_t, \hat{\Delta})$. Second,

$$\begin{aligned} \hat{\Delta} &= \mathbb{E}_{(x,y)\sim\mathcal{D}}[y|g_t(x)=1] - \mathbb{E}_{(x,y)\sim\mathcal{D}}[f_i^{t_i}(x)|g_t(x)=1] - \Delta_t \\ &= \mathbb{E}_{(x,y)\sim\mathcal{D}}[y|g_t(x)=1] - \mathbb{E}_{(x,y)\sim\mathcal{D}}[f_i^{t_i+1}(x)|g_t(x)=1] \end{aligned}$$

Third, by definition of the Round operation, $|\hat{\Delta}| \leq \frac{1}{2m}$. Therefore we can again apply lemma 13.2.2 to conclude that:

$$\begin{aligned} B(f_t^{t_i+1}) - B(\tilde{f}_t^{t_i+1}) &= \mu(g_t)\hat{\Delta}^2 \\ &\leq \frac{1}{4m^2} \end{aligned}$$

Combining this with our initial calculation lets us conclude that:

$$B(f_t^{t_i}) - B(f_t^{t_i+1}) \geq \frac{\alpha\epsilon^2}{8} - \frac{1}{4m^2} \geq \frac{\alpha\epsilon^2}{16}$$

Here we are using the fact that we have set $m \geq \frac{2}{\sqrt{\alpha\epsilon}}$. Applying this lemma for each of the T_1 and T_2 updates of f_1 and f_2 respectively we get that: $B(f_1^{T_1}) \leq B(f_1) - T_1 \cdot \frac{\alpha\epsilon^2}{16}$ and $B(f_2^{T_2}) \leq B(f_2) - T_2 \cdot \frac{\alpha\epsilon^2}{16}$. Since Brier scores are non-negative, we conclude that $T_1 \leq B(f_1) \frac{16}{\alpha\epsilon^2}$ and $T_2 \leq B(f_2) \frac{16}{\alpha\epsilon^2}$. Thus $T = T_1 + T_2 \leq (B(f_1) + B(f_2)) \cdot \frac{16}{\alpha\epsilon^2}$

Finally the halting condition of the algorithm implies that $\mu(U_\epsilon(f_1^{T_1}, f_2^{T_2})) < \alpha$.

Thus if we start with any two models that have substantial disagreement, we are guaranteed to be able to efficiently produce *strictly improved* models that almost agree almost everywhere. In particular, we can never be in a position in which we have two equally accurate *but unimprovable* models that have substantial disagreements: in this case, we can always improve the models. The only time we can have substantial model disagreement is if we refuse to improve the models even in the face of efficiently verifiable and actionable evidence that one of the models is suboptimal and improvable.

We observe that any pair of models that have gone through the ‘‘Reconcile’’ process must also produce very similar probability estimates for any sufficiently large conditional probability.

Corollary 13.2.1 *Let $E \subset \mathcal{X}$ be any subset of the data space. Let f_1 and f_2 be any two models that have been output by Algorithm 36 (Reconcile) with parameters ϵ and α . Let:*

$$p_1(E) = \sum_{x \in E} \frac{\mu(x) \cdot f_1(x)}{\mu(E)} \text{ and } p_2(E) = \sum_{x \in E} \frac{\mu(x) \cdot f_2(x)}{\mu(E)}$$

be the estimates for $\mathbb{E}[y|x \in E]$ implied by models f_1 and f_2 respectively. Then:

$$|p_1(E) - p_2(E)| \leq \frac{\alpha}{\mu(E)} + \epsilon$$

Proof 99 *Let $S_\epsilon(f_1, f_2) = \{x : x \notin U_\epsilon(f_1, f_2)\}$ be the set of points on which f_1 and f_2 do not have an ϵ -disagreement. Recall that $\mu(S_\epsilon(f_1, f_2)) \geq 1 - \alpha$. We compute:*

$$\begin{aligned} \mu(E)|p_1(E) - p_2(E)| &= \left| \sum_{x \in E} \mu(x) \cdot (f_1(x) - f_2(x)) \right| \\ &= \left| \sum_{x \in E \cap U_\epsilon(f_1, f_2)} \mu(x) \cdot (f_1(x) - f_2(x)) + \sum_{x \in E \cap S_\epsilon(f_1, f_2)} \mu(x) \cdot (f_1(x) - f_2(x)) \right| \\ &\leq \alpha + \mu(E \cap S_\epsilon(f_1, f_2))\epsilon \\ &\leq \alpha + \mu(E)\epsilon \end{aligned}$$

Dividing by $\mu(E)$ yields the corollary.

13.3 Sample Complexity

We have once again presented our algorithm 36 as if it has direct access to the distribution \mathcal{D} . Of course in general we do not have access to \mathcal{D} , but rather have access to some set D of n i.i.d. *samples* from \mathcal{D} . We will typically instead run Algorithm 36 over the *empirical distribution* over D — i.e. the distribution that puts weight $1/n$ on each datapoint $(x_i, y_i) \in D$. We will prove that with high probability over the sample of D , when Algorithm 36 is run over the empirical distribution on D , then its guarantees translate over to the distribution \mathcal{D} from which D was drawn, with error parameters that go to zero with the size n of the data sample.

We begin by counting the number of potential models $f_1^{t_1}, f_2^{t_2}$ that Algorithm 36 might output.

Lemma 13.3.1 *Fix any pair of models $f_1, f_2 : \mathcal{X} \rightarrow [0, 1]$ and any $\alpha, \epsilon > 0$. Then there is a set C of pairs of models of size at most $|C| \leq$*

$(4 \cdot (m+1))^{32/\alpha\epsilon^2+1}$ such that for any dataset distribution \mathcal{D} on which Algorithm 36 is run, the output models $(f_1^{t_1}, f_2^{t_2}) \in C$. Here, as in Algorithm 36, $m = \lceil \frac{2}{\sqrt{\alpha\epsilon}} \rceil$.

Proof 100 Given a run of Algorithm 36 for T rounds, let $\pi = \{(i_t, \bullet_t, \Delta_t)\}_{t=1}^T$ denote the record of the quantities $(i_t, \bullet_t, \Delta_t)$ chosen at each round t . Let $\pi^{<t} = \{(i_{t'}, \bullet_{t'}, \Delta_{t'})\}_{t'=1}^{t-1}$ denote the prefix of this transcript up through round $t-1$. Observe that once we fix $\pi^{<t}$ we have also fixed the models $f_1^{t_1}$ and $f_2^{t_2}$ that are defined at the start of round t . To see this, assume the claim holds true at round t . In particular, $\pi^{<t}$ fixes the disagreement regions $U_\epsilon^\bullet(f_1^{t_1}, f_2^{t_2})$ of these two models, and therefore given the choices $(i_t, \bullet_t, \Delta_t)$, we have inductively defined the models present at the start of round $t+1$.

We let C denote the set of all pairs of models defined by transcripts $\pi^{<T}$ for all $T \leq \frac{32}{\alpha\epsilon^2}$. Since we know from Theorem 65 that Algorithm 36 halts after at most $T \leq (B(f_1) + B(f_2)) \cdot \frac{16}{\alpha\epsilon^2} \leq \frac{32}{\alpha\epsilon^2}$ many rounds, and hence the models output by Algorithm 36 must be contained in C as claimed. It remains to count the set of transcripts of length $T \leq \frac{32}{\alpha\epsilon^2}$. At each round t , there are two possible values for i_t , two possible values for \bullet_t , and $m+1$ possible choices for Δ_t . Hence the number of transcripts of length T is $(4(m+1))^T$. Thus we have:

$$|C| \leq \sum_{T=0}^{\frac{32}{\alpha\epsilon^2}} (4(m+1))^T \leq (4(m+1))^{\frac{32}{\alpha\epsilon^2}+1}$$

We can now argue that if we have a sample of n datapoints D that are sampled i.i.d. from some unknown distribution \mathcal{D} , then if we run Algorithm 36 using the empirical distribution over D , then its guarantees hold also over \mathcal{D} , with error terms that tend to 0 as n grows large.

Theorem 66 Fix any data distribution \mathcal{D} and consider a run of Algorithm 36 over the empirical distribution over points in a dataset $D \sim \mathcal{D}^n$ consisting of n points sampled i.i.d. from \mathcal{D} . For any pair of models $f_1, f_2 : \mathcal{X} \rightarrow [0, 1]$ and any $\alpha, \epsilon > 0$, Algorithm 36 (Reconcile) runs for $T = T_1 + T_2$ many rounds and outputs a pair of models $(f_1^{T_1}, f_2^{T_2})$ such that:

1. $T \leq \frac{16}{\alpha\epsilon^2}$
2. For any $\delta > 0$, with probability at least $1-\delta$ over the randomness of $D \sim \mathcal{D}^n$ we have that:

$$B(f_1^{T_1}) \leq B(f_1) - T_1 \cdot \frac{\alpha\epsilon^2}{16} + 2\sqrt{\frac{(\frac{16}{\alpha\epsilon^2} + 1) \log\left(\frac{64(\lceil \frac{2}{\sqrt{\alpha\epsilon}} \rceil + 1)}{\delta}\right)}{n}}$$

and

$$B(f_2^{T_2}) \leq B(f_2) - T_2 \cdot \frac{\alpha\epsilon^2}{16} + 2\sqrt{\frac{(\frac{16}{\alpha\epsilon^2} + 1) \log\left(\frac{64(\lceil \frac{2}{\sqrt{\alpha\epsilon}} \rceil + 1)}{\delta}\right)}{n}}$$

3. For any $\delta > 0$, with probability at least $1 - \delta$ over the randomness of $D \sim \mathcal{D}^n$:

$$\mu(U_\epsilon(f_1^{T_1}, f_2^{T_2})) < \alpha + \sqrt{\frac{\left(\frac{32}{\alpha\epsilon^2} + 1\right) \log\left(\frac{8\left(\lceil \frac{2}{\sqrt{\alpha\epsilon}} \rceil + 1\right)}{\delta}\right)}{n}}.$$

Remark 13.3.1 Theorem 66 tells us that the guarantees we proved for Algorithm 36 in Theorem 65 (when we assumed direct access to the distribution \mathcal{D}) continue to hold when all we have access to is a finite sample of n points from the data distribution, with additional error terms that tend to zero as n grows large. How large is large? If we want the final disagreement region to have mass at most 2α (i.e. we want the third conclusion of Theorem 66 to tell us that $\mu(U_\epsilon(f_1^{T_1}, f_2^{T_2})) < 2\alpha$), then solving for n in the error bound, we find that it suffices to have n samples for n on the order of:

$$n \in \tilde{O}\left(\frac{\log(1/\delta)}{\alpha^3\epsilon^2}\right)$$

where the $\tilde{O}()$ notation hides logarithmic terms in $1/\alpha$ and $1/\epsilon$.

This is a remarkably small amount of data: We would need $\approx \frac{\log(1/\delta)}{\alpha\epsilon^2}$ samples just to estimate the conditional label expectation $\Pr[y = 1 | x \in S]$ for a conditional event S with $\mu(S) = \alpha$ up to error ϵ with probability $1 - \delta$ (or for two parties with disjoint samples to agree on this conditional label expectation up to error ϵ). Theorem 66 tells us that in fact two parties can be made to agree on a $1 - \alpha$ fraction of points up to error ϵ with an additional amount of data only on the order of $\tilde{O}(1/\alpha^2)$.

Proof 101 (Proof of Theorem 66) The bound on T follows directly from Theorem 65 without modification. We focus on bounding the Brier scores and the uncertainty region for the resulting models.

Consider any pair of models f_1, f_2 . Given a finite dataset D we write $(x, y) \sim D$ to denote uniformly sampling a single datapoint from D . We start by comparing $\Pr_{(x,y) \sim D}[x \in U_\epsilon(f_1, f_2)]$ with $\Pr_{(x,y) \sim \mathcal{D}}[x \in U_\epsilon(f_1, f_2)]$. We have that:

$$\Pr_{(x,y) \sim D}[x \in U_\epsilon(f_1, f_2)] = \frac{1}{n} \sum_{i=1}^n \mathbb{1}[x_i \in U_\epsilon(f_1, f_2)]$$

Since $\mathbb{1}[x_i \in U_\epsilon(f_1, f_2)] \in [0, 1]$ and

$$\mathbb{E}_{D \sim \mathcal{D}^n} \left[\Pr_{(x,y) \sim D}[x \in U_\epsilon(f_1, f_2)] \right] = \Pr_{(x,y) \sim \mathcal{D}}[x \in U_\epsilon(f_1, f_2)]$$

we can apply Hoeffding's inequality (Theorem 67) to conclude that for every $\eta > 0$:

$$\Pr_{D \sim \mathcal{D}^n} \left[\left| \Pr_{(x,y) \sim D}[x \in U_\epsilon(f_1, f_2)] - \Pr_{(x,y) \sim \mathcal{D}}[x \in U_\epsilon(f_1, f_2)] \right| \geq \eta \right] \leq 2 \exp(-2\eta^2 n)$$

Let C be the set of pairs of models guaranteed in the statement of Lemma 13.3.1. Recall that Lemma 13.3.1 guarantees us that $|C| \leq (4(m+1))^{32/\alpha\epsilon^2+1}$. We can apply the union bound to all pairs of models $(f_1, f_2) \in C$ to conclude that with probability at least $1 - 2|C| \exp(-2\eta^2 n)$ (over the randomness of D) we have that for every pair $(f_1, f_2) \in C$:

$$\left| \Pr_{(x,y) \sim D} [x \in U_\epsilon(f_1, f_2)] - \Pr_{(x,y) \sim \mathcal{D}} [x \in U_\epsilon(f_1, f_2)] \right| \leq \eta$$

Choosing

$$\eta = \sqrt{\frac{\log\left(\frac{2|C|}{\delta}\right)}{2n}}$$

we get that with probability $1 - \delta$ over the draw of D , for every pair $(f_1, f_2) \in C$:

$$\begin{aligned} \left| \Pr_{(x,y) \sim D} [x \in U_\epsilon(f_1, f_2)] - \Pr_{(x,y) \sim \mathcal{D}} [x \in U_\epsilon(f_1, f_2)] \right| &\leq \sqrt{\frac{\log\left(\frac{2|C|}{\delta}\right)}{2n}} \\ &\leq \sqrt{\frac{\left(\frac{32}{\alpha\epsilon^2} + 1\right) \log\left(\frac{8\left(\lceil \frac{2}{\sqrt{\alpha\epsilon}} \rceil + 1\right)}{\delta}\right)}{n}} \end{aligned}$$

where the final inequality follows from plugging in our bound on $|C|$ and the definition of m .

Because we know from Theorem 65 that the models $f_1^{T_1}, f_2^{T_2}$ output by Algorithm 36 satisfy that $\Pr_{(x,y) \sim D} [x \in U_\epsilon(f_1^{T_1}, f_2^{T_2})] \leq \alpha$ we can conclude that with probability $1 - \delta$:

$$\Pr_{(x,y) \sim D} [x \in U_\epsilon(f_1^{T_1}, f_2^{T_2})] \leq \sqrt{\frac{\left(\frac{32}{\alpha\epsilon^2} + 1\right) \log\left(\frac{8\left(\lceil \frac{2}{\sqrt{\alpha\epsilon}} \rceil + 1\right)}{\delta}\right)}{n}}$$

We can bound the Brier score of the resulting models in exactly the same way. For any fixed model $f : \mathcal{X} \rightarrow [0, 1]$, we can write the empirical Brier score (i.e. the Brier score as evaluated over D) as:

$$B_D(f) = \frac{1}{n} \sum_{i=1}^n (f(x_i) - y_i)^2$$

Since $(f(x_i) - y_i)^2 \in [0, 1]$ and $\mathbb{E}_{D \sim \mathcal{D}^n} [B_D(f)]$, we can apply Hoeffding's inequality (Theorem 67) exactly as before to conclude that for every pair of models $(f_1^{T_1}, f_2^{T_2}) \in C$, with probability $1 - \delta$:

$$\left| B_D(f_1^{T_1}) - B(f_1^{T_1}) \right| \leq \sqrt{\frac{\left(\frac{16}{\alpha\epsilon^2} + 1\right) \log\left(\frac{16\left(\lceil \frac{2}{\sqrt{\alpha\epsilon}} \rceil + 1\right)}{\delta}\right)}{n}}$$

and with probability $1 - \delta$:

$$\left| B_D(f_2^{T_2}) - B(f_2^{T_2}) \right| \leq \sqrt{\frac{\left(\frac{16}{\alpha\epsilon^2} + 1\right) \log\left(\frac{16\left(\lceil \frac{2}{\sqrt{\alpha\epsilon}} \rceil + 1\right)}{\delta}\right)}{n}}$$

Observe that the same holds true for the original pair of models (f_1, f_2) , since $(f_1, f_2) \in C$ (they correspond to the models output after transcripts of length 0). We further know from Theorem 65 that: $B_D(f_1^{T_1}) \leq B_D(f_1) - T_1 \cdot \frac{\alpha\epsilon^2}{16}$ and $B_D(f_2^{T_2}) \leq B_D(f_2) - T_2 \cdot \frac{\alpha\epsilon^2}{16}$.

Instantiating these bounds for the four models $\{f_1, f_2, f_1^{T_1}, f_2^{T_2}\}$, and setting $\delta \leftarrow \delta/4$ so that we can union bound over all four models, we have that with probability $1 - \delta$ that we simultaneously have:

$$B(f_1^{T_1}) \leq B(f_1) - T_1 \cdot \frac{\alpha\epsilon^2}{16} + 2\sqrt{\frac{\left(\frac{16}{\alpha\epsilon^2} + 1\right) \log\left(\frac{64\left(\lceil \frac{2}{\sqrt{\alpha\epsilon}} \rceil + 1\right)}{\delta}\right)}{n}}$$

$$B(f_2^{T_2}) \leq B(f_2) - T_2 \cdot \frac{\alpha\epsilon^2}{16} + 2\sqrt{\frac{\left(\frac{16}{\alpha\epsilon^2} + 1\right) \log\left(\frac{64\left(\lceil \frac{2}{\sqrt{\alpha\epsilon}} \rceil + 1\right)}{\delta}\right)}{n}}$$

References and Further Reading

The material from this Chapter is taken from Roth et al. [2022].



Bibliography

- Dmitry Adamskiy, Wouter M Koolen, Alexey Chernov, and Vladimir Vovk. A closer look at adaptive regret. In *International Conference on Algorithmic Learning Theory*, pages 290–304. Springer, 2012.
- Anastasios N Angelopoulos and Stephen Bates. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *arXiv preprint arXiv:2107.07511*, 2021.
- Anastasios Nikolas Angelopoulos, Stephen Bates, Michael Jordan, and Jitendra Malik. Uncertainty sets for image classifiers using conformal prediction. In *International Conference on Learning Representations*, 2020.
- Christina Baek, Yiding Jiang, Aditi Raghunathan, and Zico Kolter. Agreement-on-the-line: Predicting the performance of neural networks under distribution shift. *arXiv preprint arXiv:2206.13089*, 2022.
- Osbert Bastani, Varun Gupta, Christopher Jung, Georgy Noarov, Ramya Ramalingam, and Aaron Roth. Practical adversarial multivald conformal prediction. *arXiv preprint arXiv:2206.01067*, 2022.
- Avrim Blum and Thodoris Lykouris. Advancing subgroup fairness via sleeping experts. In *Innovations in Theoretical Computer Science Conference (ITCS)*, volume 11, 2020.
- Avrim Blum and Yishay Mansour. From external to internal regret. *Journal of Machine Learning Research*, 8(6), 2007.
- Maya Burhanpurkar, Zhun Deng, Cynthia Dwork, and Linjun Zhang. Scaffolding sets. *arXiv preprint arXiv:2111.03135*, 2021.
- A. P. Dawid. Calibration-Based Empirical Probability. *The Annals of Statistics*, 13(4):1251 – 1274, 1985. doi: 10.1214/aos/1176349736. URL <https://doi.org/10.1214/aos/1176349736>.
- A Philip Dawid. The well-calibrated bayesian. *Journal of the American Statistical Association*, 77(379):605–610, 1982.
- Dean P Foster. A proof of calibration via blackwell’s approachability theorem. *Games and Economic Behavior*, 29(1-2):73–78, 1999.

- Dean P Foster and Sergiu Hart. Forecast hedging and calibration. *Journal of Political Economy*, 129(12):3447–3490, 2021.
- Dean P Foster and Rakesh Vohra. Regret in the on-line decision problem. *Games and Economic Behavior*, 29(1-2):7–35, 1999.
- Dean P Foster and Rakesh V Vohra. Asymptotic calibration. *Biometrika*, 85(2):379–390, 1998.
- Drew Fudenberg and David K Levine. An easier way to calibrate. *Games and economic behavior*, 29(1-2):131–137, 1999.
- Isaac Gibbs and Emmanuel Candes. Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 34:1660–1672, 2021.
- Ira Globus-Harris, Declan Harrison, Michael Kearns, Aaron Roth, and Jessica Sorrell. Multicalibration as boosting for regression. 2023.
- Parikshit Gopalan, Lunjia Hu, Michael P Kim, Omer Reingold, and Udi Wieder. Loss minimization through the lens of outcome indistinguishability. *arXiv preprint arXiv:2210.08649*, 2022a.
- Parikshit Gopalan, Adam Tauman Kalai, Omer Reingold, Vatsal Sharan, and Udi Wieder. Omnipredictors. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022b.
- Varun Gupta, Christopher Jung, Georgy Noarov, Mallesh M Pai, and Aaron Roth. Online multivalid learning: Means, moments, and prediction intervals. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.
- Nika Haghtalab, Chara Podimata, and Kunhe Yang. Calibrated stackelberg games: Learning optimal commitments against calibrated agents. *arXiv preprint arXiv:2306.02704*, 2023.
- Sergiu Hart. Calibrated forecasts: The minimax proof. 2020. URL <http://www.ma.huji.ac.il/~hart/papers/calib-minmax.pdf>.
- Elad Hazan and Comandur Seshadhri. Efficient learning algorithms for changing environments. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 393–400, 2009.
- Ursula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1939–1948. PMLR, 2018.

- Christopher Jung, Changhwa Lee, Malleesh Pai, Aaron Roth, and Rakesh Vohra. Moment multicalibration for uncertainty estimation. In *Conference on Learning Theory*, pages 2634–2678. PMLR, 2021.
- Christopher Jung, Georgy Noarov, Ramya Ramalingam, and Aaron Roth. Batch multivald conformal prediction. *arXiv preprint arXiv:2209.15145*, 2022.
- 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.
- Michael P Kim, Christoph Kern, Shafi Goldwasser, Frauke Kreuter, and Omer Reingold. Universal adaptability: Target-independent inference that competes with propensity scoring. *Proceedings of the National Academy of Sciences*, 119(4):e2108097119, 2022.
- Daniel Lee, Georgy Noarov, Malleesh Pai, and Aaron Roth. Online minimax multiobjective optimization: Multicalibeating and other applications. *Advances in Neural Information Processing Systems*, 35:29051–29063, 2022.
- 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.
- Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gamerman. Inductive confidence machines for regression. In *European Conference on Machine Learning*, pages 345–356. Springer, 2002.
- Sangdon Park, Osbert Bastani, Nikolai Matni, and Insup Lee. Pac confidence sets for deep neural networks via calibrated prediction. In *International Conference on Learning Representations*, 2019.
- Yaniv Romano, Evan Patterson, and Emmanuel Candes. Conformalized quantile regression. *Advances in neural information processing systems*, 32, 2019.
- Aaron Roth, Alexander Tolbert, and Scott Weinstein. Reconciling individual probability forecasts. *arXiv preprint arXiv:2209.01687*, 2022.
- Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(3), 2008.
- Vladimir Vovk. Conditional validity of inductive conformal predictors. In *Asian conference on machine learning*, pages 475–490. PMLR, 2012.
- Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Icml*, volume 1, pages 609–616, 2001.



A

Useful Probabilistic Inequalities

CONTENTS

In this appendix we collect several useful probabilistic inequalities that will be handy in our analyses.

Theorem 67 (Hoeffding's Inequality) *Let X_1, \dots, X_n be independent random variables bounded such that for each i , $a_i \leq X_i \leq b_i$. Let $S_n = \sum_{i=1}^n X_i$ denote their sum. Then for all $t > 0$:*

$$\Pr[|S_n - \mathbb{E}[S_n]| \geq t] \leq 2 \exp\left(\frac{-2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right)$$

Theorem 68 (Chernoff's Bound) *Let X_1, \dots, X_n be independent random variables bounded such that for each i , $0 \leq X_i \leq 1$. Let $S_n = \sum_{i=1}^n X_i$ denote their sum. Then for all $\eta > 0$:*

$$\Pr[|S_n - \mathbb{E}[S_n]| \geq \eta \mathbb{E}[S_n]] \leq 2 \exp\left(-\frac{\mathbb{E}[S_n] \eta^2}{3}\right)$$

Theorem 69 (Azuma's Inequality) *Let X_1, \dots, X_n be random variables (not necessarily independent) bounded such that for each i , $|X_i| \leq c_i$. Let $X_{<i}$ denote the prefix X_1, X_2, \dots, X_{i-1} . Then for all $t > 0$:*

$$\Pr\left[\left|\sum_{i=1}^n X_i - \sum_{i=1}^n \mathbb{E}[X_i | X_{<i}]\right| \geq t\right] \leq 2 \exp\left(\frac{-t^2}{2 \sum_{i=1}^n c_i^2}\right)$$

Theorem 70 (The DKW (Dvoretzky–Kiefer–Wolfowitz) inequality) *Let $\mathcal{D} \in \mathcal{Z}^n$ be any distribution and let $D \sim \mathcal{D}^n$ consist of n points sampled i.i.d. from D . Let $F(c) = \Pr_{(x,y) \sim \mathcal{D}}[y \leq c]$ denote the CDF of the label distribution induced by \mathcal{D} , and let $\hat{F}_D(c) = \frac{1}{n} \sum_{(x,y) \in D} \mathbb{1}[y \leq c]$ denote the CDF of the empirical label distribution induced by D . Then for every $t > 0$:*

$$\Pr\left[\sup_{c \in \mathbb{R}} |F(c) - \hat{F}_D(c)| \geq t\right] \leq 2 \exp(-2nt^2)$$



B

Additional Tools for Online Convex Optimization

CONTENTS

B.1	Large Action Spaces: Online Linear Optimization	223
B.2	Follow the Regularized Leader and Online Gradient Descent ...	229
B.3	Online Convex Optimization	232
B.4	From the Minimax Theorem to Sequential Decision Making ...	235
	Bibliographic Notes and Further Reading	242

In this chapter we cover some additional useful topics in online convex optimization that we didn't have time to cover in Chapter 7.

B.1 Large Action Spaces: Online Linear Optimization

The multiplicative weights algorithm lets us get diminishing regret to the best fixed action in some comparison class \mathcal{A} : the regret guarantee grows only logarithmically with $k = |\mathcal{A}|$, and so in principle, we get very strong guarantees even for enormous action spaces. However, the running time of the algorithm grows linearly with k , and so we might struggle to run the algorithm if \mathcal{A} is truly enormous. What can we do in settings in which \mathcal{A} is exponentially large in some natural dimension of our problem, or even continuously large?

A useful running example to keep in mind is the online shortest paths problem, which is defined on a graph $G = (V, E)$, with two distinguished vertices, the source s and the sink t . The action space for the learner corresponds to the set of all $s \rightarrow t$ paths P , which can be exponentially large in the number of vertices d in the graph. Every round, a cost (congestion) c_e is realized for each edge $e \in E$ in the graph, and the cost of a path is the sum of the costs of the edges it contains: $c_P = \sum_{e \in P} c_e$. The goal is to select paths so that the learner's regret to the best fixed action (path) is tending to 0. We could run the multiplicative weights algorithm with one action per path to solve this problem with $O(\sqrt{dT})$ regret bounds, but with running time exponential in d . Can we get similar guarantees with running time polynomial in d ? We'll

show how to solve this problem as a special case of the more general online linear optimization problem.

Definition 96 (Online Linear Optimization) *In the online linear optimization problem:*

1. The Learner has an action space $\mathcal{A} \subseteq \mathbb{R}^d$, and
2. The Adversary has an action space $\mathcal{C} \subseteq \mathbb{R}^d$.
3. At each round t , the learner chooses an action $a^t \in \Delta\mathcal{A}$ and the adversary chooses an action $c^t \in \mathcal{C}$. The learner experiences cost $c_L^t = \langle a^t, c^t \rangle$

Remark B.1.1 *The online linear optimization setting generalizes the no-regret setting we have already studied, in which there are d actions, \mathcal{A} corresponds to the set of standard basis vectors (each indicating playing one of the d actions), and $\Delta\mathcal{A}$ is the set of probability distributions over the individual actions.*

It also allows us to represent the online shortest path problem: Here the dimension $d = |E|$, with one coordinate for each edge $e \in E$. The set of actions \mathcal{A} corresponds to the set of $s \rightarrow t$ paths in the graph, with each path $P \in \mathcal{A}$ represented as its indicator vector, in which $P_e = 1$ for each edge e in the path and $P_e = 0$ otherwise. $\Delta\mathcal{A}$ here represents the set of probability distributions over paths.

In general, $\Delta\mathcal{A}$ represents the convex hull of \mathcal{A} , which corresponds to the set of probability distributions over elements of \mathcal{A} if \mathcal{A} is not already a convex set. If \mathcal{A} is already a convex set then $\Delta\mathcal{A} = \mathcal{A}$, and the learner can choose an element of \mathcal{A} at every round.

Since the action space \mathcal{A} can have arbitrary, in general finding the best action even for a fixed cost vector c (a strictly easier problem) might be computationally hard. Thus we will assume that we have an oracle to solve static linear optimization problems over \mathcal{A} .

Definition 97 *A (static) linear optimization oracle for \mathcal{A} allows us to solve the following optimization problem for any vector $c \in \mathbb{R}$:*

$$a^*(c) \in \arg \min_{a \in \mathcal{A}} \langle a, c \rangle$$

How can we make use of a static linear optimization oracle to solve the online linear optimization problem? A natural first attempt is to simply optimize at every round t to play the action $a_t \in \mathcal{A}$ that has done best so far. We might call this approach “Follow the Leader”.

Algorithm 37 Follow the Leader**for** $t = 1$ to T **do**

Let:

$$c^{<t} = \sum_{s=1}^{t-1} c^s$$

Be the cumulative cost observed so far.

Select the action:

$$a^t = \arg \min_{a \in \mathcal{A}} \langle a, c^{<t} \rangle$$

Although this is a natural algorithm, it can behave quite poorly. Here is an example, in which $\mathcal{A} = \{(1, 0), (0, 1)\}$, and the sequence of costs is:

$$c^1 = (1/2, 0) \quad c^2 = (0, 1) \quad c^3 = (1, 0) \quad c^4 = (0, 1) \quad c^5 = (1, 0) \quad c^6 = (0, 1), \dots$$

Follow the leader chooses the sequence of actions

$$a^1 = (1, 0), \quad a^2 = (0, 1) \quad a^3 = (1, 0), \quad a^4 = (0, 1), \quad a^5 = (1, 0), \quad a^6 = (0, 1), \dots$$

which is the worst possible sequence of actions, and incurs cost $T - 1/2$ after T rounds. On the other hand, the best fixed action in hindsight accumulates cost only $T/2$. So we need a new idea if we want to be able to bound our regret to the best fixed action by something that grows sublinearly with T .

The fix turns out to be simple, however: essentially the only thing we need to do is avoid the oscillatory behavior of the bad example we just saw. The way to do this is to add random noise to the cumulative observed costs before invoking the linear optimization oracle to select the next action. The final algorithm is called follow the *perturbed* leader.

Algorithm 38 Follow the Perturbed Leader**for** $t = 1$ to T **do**

Let:

$$c^{<t} = \sum_{s=1}^{t-1} c^s$$

Be the cumulative cost observed so far.

Let $N^t \sim U[0, 1/\epsilon]^d$ be a uniformly random noise vector.

Select the action:

$$a^t = \arg \min_{a \in \mathcal{A}} \langle a, c^{<t} + N^t \rangle$$

Theorem 71 *The expected regret of Follow the Perturbed Leader to each action $a \in \mathcal{A}$ can be bounded as:*

$$\sum_{t=1}^T (\mathbb{E}[\langle a^t, c^t \rangle] - \langle a, c^t \rangle) \leq \frac{A}{\epsilon} + 2CRT\epsilon$$

where: $A = \max_{a, a' \in \mathcal{A}} \|a - a'\|_1$, $C = \max_{c \in \mathcal{C}} \|c\|_1$ and $R = \max_{a \in \mathcal{A}, c \in \mathcal{C}} |\langle a, c \rangle|$

Remark B.1.2 If we set $\epsilon = \sqrt{\frac{A}{2CRT}}$ then we get that Follow the Perturbed Leader has a regret bound of:

$$\min_{a \in \mathcal{A}} \sum_{t=1}^T (\mathbb{E}[\langle a^t, c^t \rangle] - \langle a, c^t \rangle) \leq \sqrt{8CART}$$

Proof 102 (Proof of Theorem 71) To analyze follow the perturbed leader, we will analyze a sequence of hypothetical algorithms, ending with follow the perturbed leader, and relate their regret bounds to each other.

The first algorithm we will consider is called “be the leader”, and selects action \hat{a}^t at each round where:

$$\hat{a}^t = \arg \min_{a \in \mathcal{A}} \langle a, c^{<t+1} \rangle$$

Note that “be the leader” selects its action by optimizing for the cumulative loss up through and including round t : $c^{<t+1}$. In other words, it is playing “follow the leader”, but shifted one step forward in time. So it is not an implementable algorithm (because it requires seeing one step into the future), but we can still analyze its behavior.

Lemma B.1.1 Be the leader has non-positive regret to every action $a \in \mathcal{A}$:

$$\sum_{t=1}^T \langle \hat{a}^t, c^t \rangle \leq \sum_{t=1}^T \langle a, c^t \rangle$$

Proof 103 We prove this by induction on T . The base case of $T = 1$ follows from the definition of $\hat{a}_1 = \arg \min_{a \in \mathcal{A}} \langle a, c^1 \rangle$. We now assume that the claim holds for $T = k$, and show that it holds for $T = k + 1$ as well:

$$\begin{aligned} \sum_{t=1}^{k+1} \langle \hat{a}^t, c^t \rangle &= \sum_{t=1}^k \langle \hat{a}^t, c^t \rangle + \langle \hat{a}^{k+1}, c^{k+1} \rangle \\ &\leq \sum_{t=1}^k \langle \hat{a}^{k+1}, c^t \rangle + \langle \hat{a}^{k+1}, c^{k+1} \rangle \\ &= \langle \hat{a}^{k+1}, \sum_{t=1}^{k+1} c^t \rangle \\ &\leq \langle a, \sum_{t=1}^{k+1} c^t \rangle \end{aligned}$$

Here the first inequality follows from the induction hypothesis, and the last inequality follows from the fact that \hat{a}^{k+1} is selected to be the minimizer of $\langle a, \sum_{t=1}^{k+1} c^t \rangle = \langle a, c^{<k+1} \rangle$

The next algorithm we will consider on our journey towards Follow the Perturbed Leader is called “Be the Perturbed Leader” and selects action \tilde{a}^t at each round where:

$$\tilde{a}^t = \arg \min_{a \in \mathcal{A}} \langle a, c^{<t+1} + N^t \rangle$$

In other words, the algorithm uses the same uniform perturbations as Follow the Perturbed Leader, but perturbs the one-step-lookahead cumulative costs of Be the Leader. The next step of the argument is to show that perturbations don’t hurt the performance of Be the Perturbed Leader too much:

Lemma B.1.2 For every sequence of cost vectors, Be the Perturbed Leader has expected regret to every action $a \in \mathcal{A}$ bounded as:

$$\mathbb{E}_N \left[\sum_{t=1}^T \langle \tilde{a}^t, c^t \rangle \right] - \sum_{t=1}^T \langle a, c^t \rangle \leq \frac{A}{\epsilon}$$

Where $A = \max_{a, a' \in \mathcal{A}} \|a - a'\|_1$

Proof 104 Since we are bounding the expected cost of Be the Perturbed Leader, we can imagine that the perturbations are coupled such that $N^1 = N^2 = \dots = N^T = N$ (with each still marginally distributed as $N \sim U[0, 1/\epsilon]^d$) — this does not effect the expected cost of the algorithm. With this observation, we can view “Be the Perturbed Leader” as actually playing “Be the Leader” in which there is an imagined “round 0” with costs $c^0 = N$ — this is equivalent, since the chosen action:

$$\tilde{a}^t = \arg \min_{a \in \mathcal{A}} \langle a, \sum_{s=1}^t c^s + N^t \rangle = \arg \min_{a \in \mathcal{A}} \langle a, \sum_{s=0}^t c^s \rangle$$

Thus we can apply the guarantee of Be the Leader from Lemma B.1.1 to conclude that:

$$\sum_{t=0}^T \langle \tilde{a}^t, c^t \rangle - \sum_{t=0}^T \langle a, c^t \rangle \leq 0$$

To translate this into a regret bound for Be the Perturbed Leader, we need to isolate the imagined “round 0” terms:

$$\sum_{t=1}^T \langle \tilde{a}^t, c^t \rangle - \sum_{t=1}^T \langle a, c^t \rangle \leq \langle a, c^0 \rangle - \langle \tilde{a}^0, c^0 \rangle = \langle a - \tilde{a}^0, c^0 \rangle \leq \frac{A}{\epsilon}$$

Here the last inequality follows from the fact that for any $a, a' \in \mathcal{A}$:

$$\langle a - a', c^0 \rangle \leq \|a - a'\|_1 \cdot \|c^0\|_\infty \leq A \cdot \frac{1}{\epsilon}$$

since c^0 is a vector with coordinates bounded by $1/\epsilon$.

All that remains is to relate the expected regret of *Be the Perturbed Leader* with that of *Follow the Perturbed Leader*. *Be the Perturbed Leader* and *Follow the Perturbed Leader* choose actions by optimizing for a cumulative loss vector that differ only in a single round (*Be the Perturbed Leader* aggregates over one additional round); the idea is that the noise added drowns out this difference, and so the two must have similar regret. Since we know that the regret of *Be the Perturbed Leader* is small, so must be the loss of *Follow the Perturbed Leader*.

$N^t, \tilde{N}^t \sim U[0, 1/\epsilon]$. Fix a round t , and let $p^t = c^{<t} + N^t$ be the noisy accumulated vector of costs that *Follow the Perturbed Leader* optimizes for, and let $\tilde{p}^t = c^{<t+1} + \tilde{N}^t$ be the noisy accumulated vector of costs that *Be the Perturbed Leader* optimizes for. Observe that p^t is uniformly distributed in $c^{<t} + [0, 1/\epsilon]^d$ and that \tilde{p}^t is uniformly distributed in $c^{<t+1} + [0, 1/\epsilon]^d$. We will first observe that these two regions have large overlap:

Lemma B.1.3 Let $O^t = \{c^{<t} + [0, 1/\epsilon]^d\} \cap \{c^{<t+1} + [0, 1/\epsilon]^d\}$. Then:

$$\Pr[p^t \notin O^t] \leq C\epsilon \quad \Pr[\tilde{p}^t \notin O^t] \leq C\epsilon$$

Where $C = \max_{c \in \mathcal{C}} \|c\|_1$

Proof 105 We can calculate:

$$\begin{aligned} \Pr[p^t \notin O^t] &= \Pr[(c^{<t} + N^t) \notin \{c^{<t+1} + [0, 1/\epsilon]^d\}] \\ &= \Pr[N^t \notin \{c^t + [0, 1/\epsilon]^d\}] \\ &\leq \sum_{i=1}^d \Pr[N_i^t \notin [c_i^t, 1/\epsilon + c_i^t]] \\ &\leq \sum_{i=1}^d |c_i^t| \epsilon \\ &= \|c^t\|_1 \epsilon \\ &\leq C\epsilon \end{aligned}$$

The claim for \tilde{p}^t follows identically.

The key observation is that because p^t and \tilde{p}^t are uniformly distributed in their respective ranges, conditionally on lying in O^t , they are uniformly distributed within O^t . Since the corresponding actions a^t and \tilde{a}^t taken by *Be the Perturbed Leader* and *Follow the Perturbed Leader* are deterministic post-processings of \tilde{p}^t and p^t respectively, this means that the expected loss of a^t and \tilde{a}^t are identical conditional on $p^t \in O^t$ and $\tilde{p}^t \in O^t$.

Lemma B.1.4 At each round t :

$$\mathbb{E}[\langle a^t, c^t \rangle] \leq \mathbb{E}[\langle \tilde{a}^t, c^t \rangle] + 2CR\epsilon$$

Where $C = \max_{c \in \mathcal{C}} \|c\|_1$ and $R = \max_{a \in \mathcal{A}, c \in \mathcal{C}} |\langle a, c \rangle|$

Proof 106 Fix any round t . We can calculate:

$$\begin{aligned}
\mathbb{E}[\langle a^t, c^t \rangle] &= \Pr[p^t \in O^t] \mathbb{E}[\langle a^t, c^t \rangle | p^t \in O^t] + \Pr[p^t \notin O^t] \mathbb{E}[\langle a^t, c^t \rangle | p^t \notin O^t] \\
&\leq \Pr[p^t \in O^t] \mathbb{E}[\langle a^t, c^t \rangle | p^t \in O^t] + C\epsilon \max_{a \in \mathcal{A}, c \in \mathcal{C}} \langle a, c \rangle \\
&= \Pr[p^t \in O^t] \mathbb{E}[\langle \tilde{a}^t, c^t \rangle | \tilde{p}^t \in O^t] + CR\epsilon \\
&\leq \mathbb{E}[\langle \tilde{a}^t, c^t \rangle | \tilde{p}^t \in O^t] + CR\epsilon \\
&\leq \mathbb{E}[\langle \tilde{a}^t, c^t \rangle] + 2CR\epsilon
\end{aligned}$$

Here we have used Lemma B.1.3 twice.

We are now ready to complete the proof by chaining together our Lemmas. We have that the regret of Follow the Perturbed Leader to any action a is bounded by:

$$\begin{aligned}
\sum_{t=1}^T (\mathbb{E}[\langle a^t, c^t \rangle] - \langle a, c^t \rangle) &\leq \sum_{t=1}^T (\mathbb{E}[\langle \tilde{a}^t, c^t \rangle] - \langle a, c^t \rangle) + 2CRT\epsilon \\
&\leq \frac{A}{\epsilon} + 2CRT\epsilon
\end{aligned}$$

Here the first inequality follows from Lemma B.1.4 and the second inequality follows from Lemma B.1.2.

B.2 Follow the Regularized Leader and Online Gradient Descent

In this section we take another perspective on “Follow the Leader” style algorithms, and derive another online linear optimization algorithm that will sometimes be useful. We’ll start by directly proving a regret bound for “Follow the Leader” (Algorithm 37). Recall that we have already seen a sequence on which Follow the Leader accumulates linear regret, so the bound that we prove cannot possibly guarantee sublinear regret for all sequences — but can give us insight into *what* kinds of sequences cause Follow the Leader to have large regret, and how to avoid it.

Theorem 72 *On any sequence of costs, the regret of Follow the Leader to any fixed action $a \in \mathcal{A}$ can be bounded as:*

$$\sum_{t=1}^T (\langle a^t, c^t \rangle - \langle a, c^t \rangle) \leq \sum_{t=1}^T (\langle a^t, c^t \rangle - \langle a^{t+1}, c^t \rangle)$$

Remark B.2.1 A couple of things about Theorem 72 are worth noting. First, the right hand side can be bounded as a function of the number of rounds the leader changes — i.e. the number of rounds t such that $a^t \neq a^{t+1}$. So if the “leader” is relatively stable, then Follow the Leader in fact already will have low regret. This is consistent with the bad example we say for Follow the Leader in Section B.1 — in that example, the leader changed at every round, which can make the above bound grow linearly with T .

Observe also that a^{t+1} , the action that Follow the Leader plays at round $t+1$, is also the action that the hypothetical algorithm “Be the Leader” (which has one round-lookahead) considered in Section B.1 plays at round t .

Proof 107 (Proof of Theorem 72) We’ve actually already proven this theorem! By subtracting off $\sum_{t=1}^T \langle a^t, c^t \rangle$ from both sides, we see that the statement we want to prove is that:

$$\sum_{t=1}^T \langle a^{t+1}, c^t \rangle \leq \sum_{t=1}^T \langle a, c^t \rangle$$

But this is what we proved in Lemma B.1.1 — that the one-lookahead algorithm “Be the Leader” has non-positive regret.

Theorem 72 gives us an idea of how to modify Follow the Leader to give it a worst-case regret bound: “regularize” it so that it can’t change actions too dramatically. This is morally quite similar to our solution in Follow the Perturbed Leader — the added perturbations in FTPL make sure that a^t and a^{t+1} are *distributed* similarly — here we will instead make sure that they are close in distance. Follow the Regularized Leader will be defined by a *regularization* function $\Psi : \mathcal{A} \rightarrow \mathbb{R}$.

Algorithm 39 Follow the Regularized Leader

for $t = 1$ to T **do**
 Select the action:

$$a^t = \arg \min_{a \in \mathcal{A}} \left(\sum_{s=1}^{t-1} \langle a, c^s \rangle + \Psi(a) \right)$$

Theorem 73 For any sequence of costs and any non-negative regularizer Ψ , the regret of Follow the Regularized Leader to any fixed action $a \in \mathcal{A}$ can be bounded as:

$$\sum_{t=1}^T (\langle a^t, c^t \rangle - \langle a, c^t \rangle) \leq \sum_{t=1}^T (\langle a^t, c^t \rangle - \langle a^{t+1}, c^t \rangle) + (\Psi(a) - \Psi(a^1))$$

Proof 108 We can view Follow the Regularized Leader as Follow the Leader

in which there is a round 0 in which the cost for playing each action a is $\Psi(a)$. We can therefore apply Theorem 72 (whose proof did not require that the losses be linear) including round 0, which gives:

$$(\Psi(a^0) - \Psi(a)) + \sum_{t=1}^T (\langle a^t, c^t \rangle - \langle a, c^t \rangle) \leq \sum_{t=1}^T (\langle a^t, c^t \rangle - \langle a^{t+1}, c^t \rangle) + (\Psi(a^0) - \Psi(a^1))$$

which gives the theorem after rearranging.

Remark B.2.2 Noting that $a^1 \in \arg \min_a \Psi(a)$, this implies a bound of:

$$\sum_{t=1}^T (\langle a^t, c^t \rangle - \langle a, c^t \rangle) \leq \sum_{t=1}^T (\langle a^t, c^t \rangle - \langle a^{t+1}, c^t \rangle) + (\max_{a'} \Psi(a') - \min_{a'} \Psi(a'))$$

Therefore the goal is to find a regularizer Φ that doesn't take values that are too large, but also constrains a^t to be close to a^{t+1} . There are different interesting regularizers you could pick, but we'll investigate what happens when $\Phi(a) = \frac{1}{\eta} \|a\|_2^2$, the (scaled) squared Euclidean norm of a . In this case, when \mathcal{A} is unconstrained, we can compute a closed form expression for a^t :

Lemma B.2.1 When $\mathcal{A} = \mathbb{R}^d$ and $\Phi(a) = \frac{1}{\eta} \|a\|_2^2$, then:

$$a^t = -\frac{\eta}{2} \sum_{s=1}^{t-1} c^s$$

Proof 109 By definition:

$$a^t = \arg \min_a \left(\sum_{s=1}^{t-1} \langle a, c^s \rangle + \frac{1}{\eta} \|a\|_2^2 \right)$$

a^t is minimizing a strictly convex function, and so to find the unique minimizer of this function, we can take the gradient and set it to zero. Doing so we find that a^t must solve:

$$\frac{2}{\eta} a^t = - \sum_{s=1}^{t-1} c^s$$

or in other words:

$$a^t = -\frac{\eta}{2} \sum_{s=1}^{t-1} c^s$$

Remark B.2.3 Observe that the form of the action a^t derived in Lemma B.2.1 has a simple update rule: $a^{t+1} = a^t - \frac{\eta}{2} c^t$. In other words, the algorithm simply takes a step in the direction away from the gradient of the most recent loss function. As a result, we can call this algorithm "online gradient descent".

Algorithm 40 Online Gradient DescentLet $a^1 = 0$ **for** $t = 1$ to T **do** Select the action a^t . Observe c^t and let:

$$a^{t+1} = a^t - \frac{\eta}{2} c^t$$

Theorem 74 For any sequence of costs c^1, \dots, c^T with $\|c^t\|_2 \leq C$ for all t , and for any $a \in \mathbb{R}^d$ with $\|a\|_2 \leq A$, Online Gradient Descent obtains regret:

$$\sum_{t=1}^T \langle a^t, c^t \rangle \leq \sum_{t=1}^T \langle a, c^t \rangle + T \frac{C^2 \eta}{2} + \frac{A^2}{\eta}$$

Remark B.2.4 Setting $\eta = \frac{A}{C} \sqrt{\frac{2}{T}}$ we get a regret bound of:

$$\sum_{t=1}^T \langle a^t, c^t \rangle \leq \sum_{t=1}^T \langle a, c^t \rangle + AC\sqrt{2T}$$

Observe that this bound is independent of the dimension of the actions and costs d , and depends only on their norm.

Proof 110 (Proof of Theorem 74) We know from Lemma B.2.1 that Online Gradient Descent is an instantiation of Follow the Regularized Leader with $\Psi(a) = \frac{1}{\eta} \|a\|_2^2$. Thus we can apply Theorem 73 to conclude that:

$$\sum_{t=1}^T (\langle a^t, c^t \rangle - \langle a, c^t \rangle) \leq \sum_{t=1}^T (\langle a^t, c^t \rangle - \langle a^{t+1}, c^t \rangle) + \frac{1}{\eta} \|a\|_2^2$$

We have that $a^t - a^{t+1} = \frac{\eta}{2} c^t$, and so

$$\|a^t - a^{t+1}\|_2 \leq \frac{\eta}{2} \|c^t\|_2 \leq \frac{C\eta}{2}$$

Therefore we have that:

$$\langle a^t, c^t \rangle - \langle a^{t+1}, c^t \rangle = \langle (a^t - a^{t+1}), c^t \rangle \leq \|a^t - a^{t+1}\| \cdot \|c^t\| \leq \frac{C^2 \eta}{2}$$

Thus we have the final regret bound:

$$\sum_{t=1}^T (\langle a^t, c^t \rangle - \langle a, c^t \rangle) \leq T \frac{C^2 \eta}{2} + \frac{A^2}{\eta}$$

In general, Follow the Regularized Leader is a flexible design template: As an exercise, you can derive a Multiplicative-Weights like algorithm and bound by using the negative entropy regularizer $\Psi(a) = -\frac{1}{\eta} \sum_{i=1}^d a_i \ln(1/a_i)$.

B.3 Online Convex Optimization

We've now seen three algorithms for solving the online linear optimization problem (we were explicit about this for the Follow the Perturbed Leader and Online Gradient Descent algorithms, but it's not hard to see that the Multiplicative Weights algorithm solves the online linear optimization problem where the Learner's action space is the Simplex). In fact, algorithms that solve the online linear optimization problem can also be used to solve the more general online convex optimization problem. First we will recall some basic definitions that we will need.

Definition 98 A function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is convex if for all $x_1, x_2 \in \mathbb{R}^d$, and for all $0 \leq \alpha \leq 1$:

$$f(\alpha x_1 + (1 - \alpha)x_2) \leq \alpha f(x_1) + (1 - \alpha)f(x_2)$$

Linear functions are a special case of convex functions in which the inequality always holds with equality.

Definition 99 A function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is L -Lipschitz in the L_1 norm if for all $x_1, x_2 \in \mathbb{R}^d$:

$$|f(x_1) - f(x_2)| \leq L \|x_1 - x_2\|_1$$

If f is L -Lipschitz for some L we simply say that f is Lipschitz-continuous.

Definition 100 Fix a convex function $f : \mathbb{R}^d \rightarrow \mathbb{R}$. A vector $c \in \mathbb{R}^d$ is a subgradient of f at $x \in \mathbb{R}^d$ if for all $x' \in \mathbb{R}^d$:

$$f(x') - f(x) \geq \langle c, x' - x \rangle$$

If f is differentiable, then the gradient $\nabla f(x) = c$ is always a subgradient of f at x .

The following fact (which is not hard to prove, but we will take as given so as not to be led too far astray) will be useful in conceptualizing the reduction that follows in this section. It states that convex Lipschitz functions have bounded gradients, and vice versa.

Lemma B.3.1 A convex function f is L -Lipschitz (in the L_1 norm) if and only if for every x in its domain, and for every subgradient c of f at x , $\|c\|_\infty = \max_{i \in [d]} |c_i| \leq L$.

Finally, let us define the online convex optimization problem.

Definition 101 (Online Convex Optimization) In the L -Lipschitz online convex optimization problem:

1. The Learner has a convex action space $\mathcal{A} \subseteq \mathbb{R}^d$, and

2. The Adversary has an action space \mathcal{C} consisting of L -lipschitz convex functions $\ell : \mathcal{A} \rightarrow \mathbb{R}^d$,
3. At each round t , the learner chooses an action $a^t \in \mathcal{A}$ and the adversary chooses a loss function $\ell^t \in \mathcal{C}$. The learner experiences cost $c_L^t = \ell^t(a^t)$.

After realizing a transcript π^T , the regret that the learner experiences to action $a \in \mathcal{A}$ is:

$$\text{Reg}(\pi^T, a) = \sum_{t=1}^T (\ell^t(a^t) - \ell^t(a))$$

Remark B.3.1 Here we have assumed that the Learner's action space \mathcal{A} is convex. If it is not, as in previous sections, we can take the Learner's action space to be $\Delta\mathcal{A}$, the convex hull of \mathcal{A} , which is realized through randomization. Thus the online convex optimization setting generalizes the settings we have considered so far.

We will observe a generic reduction that converts an arbitrary algorithm for online linear optimization (like multiplicative weights, online gradient descent, or follow the perturbed leader) into an algorithm for online (Lipschitz) convex optimization with similar regret bounds.

Algorithm 41 A Reduction from Online Convex Optimization to Online Linear Optimization

Given: An algorithm `LinearLearn` for d -dimensional online linear optimization.

for $t = 1$ to T **do**

From `LinearLearn`, obtain action a^t .

From the adversary, obtain L -Lipschitz loss function ℓ^t .

Let $c^t \in [0, L]^d$ be a subgradient of ℓ^t at a^t (if ℓ^t is differentiable, $c^t = \nabla \ell^t(a^t)$).

Feed c^t to `LinearLearn` as a cost vector.

Theorem 75 Suppose `LinearLearn` is an online linear optimization algorithm that obtains regret to each action $a \in \mathcal{A}$ bounded by $R(T)$ after T rounds, for all sequences of cost vectors $c^t \in [0, L]^d$. The the reduction in Algorithm 41 obtains regret to all fixed actions bounded as:

$$\text{Reg}(\pi^T, a) = \sum_{t=1}^T (\ell^t(a^t) - \ell^t(a)) \leq R(T)$$

for all sequences of L -Lipschitz convex functions.

Proof 111 Fix a comparison action $a \in \mathcal{A}$. We know from the guarantees of

the online linear optimization algorithm that:

$$\begin{aligned}
 R(T) &\geq \sum_{t=1}^T (\langle a^t, c^t \rangle - \langle a, c^t \rangle) \\
 &= \sum_{t=1}^T \langle a^t - a, c^t \rangle \\
 &\geq \sum_{t=1}^T \ell^t(a^t) - \ell^t(a) \\
 &= \text{Reg}(\pi^T, a)
 \end{aligned}$$

Here the last inequality follows from the fact that c^t is a subgradient of ℓ^t at a^t and Definition 100.

Thus, for any fixed action space \mathcal{A} and Lipschitz parameter L , we can use (e.g.) Multiplicative Weights, Online Gradient Descent, or Follow the Perturbed Leader to solve the online convex optimization problem with regret bounds scaling as $O(\sqrt{T})$.

Remark B.3.2 Finally, we note that although we have been discussing online convex minimization, all of these algorithms can be used for online concave maximization, simply by flipping the sign of the loss functions ℓ . If ℓ is convex, then $-\ell$ is concave, and a minimizer of ℓ within \mathcal{A} is a maximizer of $-\ell$ within \mathcal{A} .

B.4 From the Minimax Theorem to Sequential Decision Making

We used the existence of online convex optimization algorithms with regret guarantees to prove the minimax theorem for zero sum games. But historically, the minimax theorem came first. Suppose we knew the minimax theorem: could we use it to derive the existence of online convex optimization algorithms with regret guarantees? The answer is *yes*, and so in a strong sense, these kinds of “no regret learning” algorithms should be viewed as equivalent, constructive versions of the minimax theorem.

Recall from Section B.3 that to derive online convex optimization algorithms it suffices to derive online *linear* optimization algorithms, and that the simplest kind of online linear optimization algorithm is an algorithm (like multiplicative weights) that selects amongst k actions at each round — this is just online linear optimization in which the learner’s action space is the k -dimensional probability simplex. So that’s what we will do for simplicity

— we'll see how to derive a multiplicative weights like algorithm and regret bound in this way.

First we recall the setting: there are k actions, and cost vectors $c^t \in [0, 1]^k$. At every round, the learner chooses some (distribution on) action(s) $p^t \in \Delta\mathcal{A}$, after which an adversary chooses a cost vector c^t . The cost for the algorithm at round t if they play an action i is $c_L^t = c_i^t$ — the expected cost for the algorithm at round t is $\mathbb{E}_{i \sim p^t}[c_L^t] = \langle p^t, c^t \rangle$. After T rounds, recall that the accumulated regret to action i is:

$$\text{Reg}(\pi^T, i) = C_L^T - C_i^T = \sum_{t=1}^T (c_L^t - c_i^t)$$

And the overall regret is $\text{Reg}(\pi^T) = \max_{i \in [k]} \text{Reg}(\pi^T, i)$. Our high level strategy will be to invoke the minimax theorem to find a strategy p^t that the learner can play at each round t to minimize the increase in her overall regret — but the max term that shows up in the overall regret of the learner makes this increase a complicated, non convex-concave function of p^t and c^t , which prevents a direct invocation. Instead we need a *surrogate* function that has better analytic properties, but can be used to upper-bound overall regret. A natural choice is the softmax function, which smoothly approximates the max function using exponentials.

Definition 102 *At round T , define the softmax surrogate with parameter η to be:*

$$L(\pi^T) = \sum_{i=1}^k \exp(\eta \text{Reg}(\pi^T, i))$$

We can use the softmax surrogate to upper bound the overall regret of the algorithm as follows:

Lemma B.4.1 *For all T :*

$$\text{Reg}(\pi^T) \leq \frac{1}{\eta} \ln(L(\pi^T))$$

Proof 112

$$\begin{aligned} \eta \text{Reg}(\pi^T) &= \max_{i \in [k]} \eta \text{Reg}(\pi^T, i) \\ &= \ln \left(\exp \left(\max_{i \in [k]} \eta \text{Reg}(\pi^T, i) \right) \right) \\ &= \ln \left(\max_{i \in [k]} \exp(\eta \text{Reg}(\pi^T, i)) \right) \\ &\leq \ln \left(\sum_{i=1}^k \exp(\eta \text{Reg}(\pi^T, i)) \right) \\ &= \ln(L(\pi^T)) \end{aligned}$$

Dividing by η gives the result.

Remark B.4.1 Observe that the soft-max upper bound on regret is reasonably tight — it cannot over-estimate the regret by more than an additive term of $\frac{1}{\eta} \cdot \log k$. Because $L(\pi^T) \leq k \exp(\eta(\text{Reg}(\pi^T)))$, we have that $\frac{1}{\eta} \log(L(\pi^T)) \leq \frac{\log k}{\eta} + \text{Reg}(\pi^T)$

Thus it will suffice to design an algorithm that can control the growth of $L(\pi^T)$.

Definition 103 Fix a transcript π^{s-1} . Given an action $i \in [k]$ and a cost vector $c \in [0, 1]^k$ let $\tilde{\pi}^s = \pi^{s-1} \circ (i, c)$, the continuation of the transcript that would result if the learner picked action i and the adversary picked cost vector c . Let:

$$\Delta_2^{\pi^{s-1}}(c, i) = L(\tilde{\pi}^s) - L(\pi^{s-1})$$

be the change in the squared error surrogate that would result from the play (i, c) at round s .

Our first step is to analytically upper bound the increase in the softmax surrogate that results from playing action i against cost vector c at round s :

Lemma B.4.2 For any $\eta \leq 1$:

$$\Delta^{\pi^{s-1}}(c, i) \leq \left(\sum_{j=1}^k \exp(\eta \text{Reg}(\pi^{s-1}, j)) \eta(c_i - c_j) \right) + \eta^2 L(\pi^{s-1})$$

Proof 113

$$\begin{aligned} \Delta^{\pi^{s-1}}(c, i) &= L(\tilde{\pi}^s) - L(\pi^{s-1}) \\ &= \sum_{j=1}^k \exp(\eta \text{Reg}(\tilde{\pi}^s, j)) - \exp(\eta \text{Reg}(\pi^{s-1}, j)) \\ &= \sum_{j=1}^k \exp(\eta \text{Reg}(\pi^{s-1}, j) + \eta(c_i - c_j)) - \exp(\eta \text{Reg}(\pi^{s-1}, j)) \\ &= \sum_{j=1}^k \exp(\eta \text{Reg}(\pi^{s-1}, j)) (\exp(\eta(c_i - c_j)) - 1) \\ &\leq \sum_{j=1}^k \exp(\eta \text{Reg}(\pi^{s-1}, j)) (\eta(c_i - c_j) + (\eta(c_i - c_j))^2) \\ &\leq \sum_{j=1}^k \exp(\eta \text{Reg}(\pi^{s-1}, j)) (\eta(c_i - c_j) + \eta^2) \\ &= \left(\sum_{j=1}^k \exp(\eta \text{Reg}(\pi^{s-1}, j)) \eta(c_i - c_j) \right) + \eta^2 L(\pi^{s-1}) \end{aligned}$$

Where the second to last inequality follows from the fact that for any $x \leq 1$, $\exp(x) \leq 1 + x + x^2$ and the last inequality follows from the assumption that $c_i, c_j \in [0, 1]$.

Observe that by construction, for a transcript π^T generated by a sequence of plays (p^t, c^t) , the expected softmax surrogate regret is exactly:

$$\mathbb{E}[L(\pi^T)] = \sum_{t=1}^T \mathbb{E}_{i \sim p^t} [\Delta^{\pi^{t-1}}(c^t, i)]$$

Thus, to guarantee that the expected regret of the Learner is small, our goal will be to find distributions p^t at each round t that guarantee that $\mathbb{E}_{i \sim p^t} [\Delta^{\pi^{t-1}}(c, i)]$ is small for all c (remembering that we don't know the relevant loss vector c^t at the time that we must pick p^t).

Towards this end, we define a zero-sum game between the learner and the adversary at each round t as follows. We will identify the Learner with the minimization player and the Adversary with the maximization player.

Definition 104 *The round- t softmax-surrogate game is*

1. *The learner's action space is $\mathcal{A}_{min} = \Delta[k]$, which is a convex set.*
2. *The adversary's action space is $\mathcal{A}_{max} = [0, 1]^k$, which is a convex set.*
3. *For each $c \in \mathcal{A}_{max}$ and $p \in \mathcal{A}_{min}$, the utility function is defined as:*

$$u(c, p) = \mathbb{E}_{i \sim p} \left[\sum_{j=1}^k \exp(\eta \text{Reg}(\pi^{s-1}, j)) \eta(c_i - c_j) \right]$$

This is a bounded utility function that is linear (and hence convex/concave) in both of its arguments (as at round t , $\text{Reg}(\pi^{t-1}, j)$ is simply a fixed constant).

We observe that this game satisfies the conditions of the minimax Theorem 42.

Lemma B.4.3 *The maximin value of the round t softmax-surrogate game is:*

$$\max_{c \in [0, 1]^k} \min_{p \in \Delta[k]} u(c, p) \leq 0$$

Proof 114 *For any $c \in \mathcal{A}_{max}$, let $i^*(c) \in \arg \min_{i \in [k]} c_i$ be a coordinate of minimum cost. the Learner has a best response $p^*(c)$ corresponding to a distribution that places all of its weight on $i^*(c)$. We have:*

$$u(c, p^*(c)) = \sum_{j=1}^k \exp(\eta \text{Reg}(\pi^{s-1}, j)) \eta(c_{i^*(c)} - c_j) \leq 0$$

Where the inequality follows because term by term, $\exp(\eta \text{Reg}(\pi^{s-1}, j)) \geq 0$ because of the non-negativity of the exponential function, and $(c_{i^*(c)} - c_j) \leq 0$ because by definition of $i^*(c)$, $c_{i^*(c)} \leq c_j$ for all j .

Since the conditions of the minimax Theorem (Theorem 42) are satisfied by our round- t surrogate softmax game, we can swap the min and the max and conclude:

Lemma B.4.4 *The minimax value of the round t softmax surrogate game is:*

$$\min_{p \in \Delta[k]} \max_{c \in [0,1]^k} u(c, p) \leq 0$$

In other words, at each round t , there exists a distribution over actions $p \in \Delta[k]$ such that for all cost vectors $c \in [0,1]^k$ that the adversary might choose, $u(c, p) \leq 0$.

Because we have defined the utility function in our round- t softmax-surrogate game to be a value that we can use to upper bound the per-round expected change in softmax-surrogate regret, we can immediately use this fact to derive a regret bound on Algorithm 42, which simply plays at every round a distribution $p^t \in \Delta[k]$ such that: $\max_{c \in [0,1]^k} u(c, p) \leq 0$. The existence of such a distribution is guaranteed by Lemma B.4.4.

Algorithm 42 A Minimax Based Sequential Decision Making Algorithm

for $t = 1$ to T **do**

Construct the round t softmax-surrogate game as a function of π^{t-1} with utility function u .

Play $p^t \in \Delta[k]$ such that: $\max_{c \in [0,1]^k} u(c, p) \leq 0$.

Theorem 76 *Against any sequence of cost functions $c^1, \dots, c^T \in [0,1]^k$, Algorithm 42 has expected regret bounded by:*

$$\mathbb{E}_{\pi^T} [\text{Reg}(\pi^T)] \leq \frac{\ln k}{\eta} + T\eta$$

Choosing $\eta = \sqrt{\frac{\ln k}{T}}$ gives:

$$\mathbb{E}_{\pi^T} [\text{Reg}(\pi^T)] \leq 2\sqrt{T \ln k}$$

Remark B.4.2 *Note that this exactly matches the worst-case regret bound we proved for the Multiplicative Weights algorithm in Theorem 39!*

Proof 115 *We start by upper bounding the expected softmax surrogate regret. From the definition of $\Delta^{\pi^{s-1}}(c, i)$ and Lemma B.4.2 we have that for all rounds*

s :

$$\begin{aligned}
\mathbb{E}_{p^s}[L(\pi^s)|\pi^{s-1}] &= L(\pi^{s-1}) + \mathbb{E}_{i \sim p^s}[\Delta^{\pi^{s-1}}(c^s, i)] \\
&\leq (1 + \eta^2)L(\pi^{s-1}) + \mathbb{E}_{i \sim p^s} \left[\left(\sum_{j=1}^k \exp(\eta \text{Reg}(\pi^{s-1}, j)) \eta(c_i - c_j) \right) \right] \\
&= (1 + \eta^2)L(\pi^{s-1}) + u(c^s, p^s) \\
&\leq (1 + \eta^2)L(\pi^{s-1})
\end{aligned}$$

Where the last inequality follows from the fact (justified by Lemma B.4.4) that p^s satisfies $u(c^s, p^s) \leq 0$ for all values of c^s .

Observing that $L(\pi^0) = k$ and applying the above bound inductively, we find that:

$$\mathbb{E}_{\pi^T}[L(\pi^T)] \leq k(1 + \eta^2)^T \leq k \exp(T\eta^2)$$

We can now apply Lemma B.4.1 to bound the regret by the softmax surrogate regret.

$$\begin{aligned}
\mathbb{E}_{\pi^T}[\text{Reg}(\pi^T)] &\leq \mathbb{E}_{\pi^T} \left[\frac{1}{\eta} \ln(L(\pi^T)) \right] \\
&\leq \frac{1}{\eta} \ln \left(\mathbb{E}_{\pi^T} [L(\pi^T)] \right) \\
&\leq \frac{1}{\eta} (\ln k + T\eta^2) \\
&= \frac{\ln k}{\eta} + T\eta
\end{aligned}$$

We proved the existence of the strategy used at each round of 42 non-constructively using the minimax theorem. How would we actually implement it? We require finding a minimax strategy for the softmax surrogate game at each round to find a p^t such that $\max_{c \in [0,1]^k} u(c, p) \leq 0$. In general we could do this using linear programming, or the no-regret dynamics approaches to computing minimax optimal strategies that we studied in Section 8.1. But in this case, it turns out that there is a simple closed form expression for p^t !

Lemma B.4.5 *Let $p^t \in \Delta[k]$ be such that:*

$$p_i^t = \frac{1}{\Phi^t} \exp(\eta \text{Reg}(\pi^{t-1}, i)) \quad \Phi^t = \sum_i \exp(\eta \text{Reg}(\pi^{t-1}, i))$$

Then p^t satisfies the requirements of Algorithm 42 — for all $c \in [0, 1]^k$:

$$u(c, p^t) = \mathbb{E}_{i \sim p^t} \left[\left(\sum_{j=1}^k \exp(\eta \text{Reg}(\pi^{t-1}, j)) \eta(c_i - c_j) \right) \right] \leq 0$$

Proof 116 We can directly compute:

$$\begin{aligned}
u(c, p^t) &= \mathbb{E}_{i \sim p^t} \left[\left(\sum_{j=1}^k \exp(\eta \text{Reg}(\pi^{t-1}, j)) \eta(c_i - c_j) \right) \right] \\
&= \sum_{i=1}^k p_i^t \sum_{j=1}^k \exp(\eta \text{Reg}(\pi^{t-1}, j)) \eta(c_i - c_j) \\
&= \frac{1}{\Phi^t} \sum_{i=1}^k \sum_{j=1}^k \exp(\eta \text{Reg}(\pi^{t-1}, i)) \exp(\eta \text{Reg}(\pi^{t-1}, j)) \eta(c_i - c_j) \\
&= \frac{\eta}{\Phi^t} \sum_{i=1}^k \sum_{j=1}^k \exp(\eta \text{Reg}(\pi^{t-1}, i)) \exp(\eta \text{Reg}(\pi^{t-1}, j)) c_i - \\
&\quad \frac{\eta}{\Phi^t} \sum_{i=1}^k \sum_{j=1}^k \exp(\eta \text{Reg}(\pi^{t-1}, i)) \exp(\eta \text{Reg}(\pi^{t-1}, j)) c_j \\
&= 0
\end{aligned}$$

Thus we have derived a concrete, easy to implement algorithm, that is a slight variant of multiplicative weights —this variant is sometimes called “Exponential Weights” .

Algorithm 43 The Exponential Weights Algorithm

For each action $i \in \mathcal{A}$, set $w_i^1 = 1$. Let $W^1 = \sum_{i \in \mathcal{A}} w_i^1$.

for $t = 1$ to T **do**

 Play the distribution p^t defined as:

$$p_i^t = \frac{w_i^t}{W^t}$$

 Observe costs c^t and update weights such that for each $i \in \mathcal{A}$:

$$w_i^{t+1} = w_i^t \exp(-\eta c_i^t) \quad W^{t+1} = \sum_{i \in \mathcal{A}} w_i^{t+1}$$

Theorem 77 *The Exponential Weights Algorithm (Algorithm 43) implements Algorithm 42, and so satisfies the regret bound proven in Theorem 76:*

$$\mathbb{E}_{\pi^T} [\text{Reg}(\pi^T)] \leq \frac{\ln k}{\eta} + T\eta$$

Choosing $\eta = \sqrt{\frac{\ln k}{T}}$ gives:

$$\mathbb{E}_{\pi^T} [\text{Reg}(\pi^T)] \leq 2\sqrt{T \ln k}$$

Proof 117 From the update rule, we can compute that at round t , Algorithm 43 plays a distribution p_i^t defined as

$$p_i^t = \frac{1}{W^t} \exp\left(-\eta \sum_{s=1}^{t-1} c_i^s\right) \quad W^t = \sum_{i=1}^k \exp\left(-\eta \sum_{s=1}^{t-1} c_i^s\right)$$

Our goal is to show that this distribution is identical to the distribution that we proved in Lemma B.4.5 implements Algorithm 42. Recall that we wrote that distribution as:

$$q_i^t = \frac{1}{\Phi^t} \exp(\eta \text{Reg}(\pi^{t-1}, i)) \quad \Phi^t = \sum_i \exp(\eta \text{Reg}(\pi^{t-1}, i))$$

Consider any coordinate i of the distribution defined in Lemma B.4.5. we can calculate:

$$\begin{aligned} q_i^t &= \frac{\exp(\eta \text{Reg}(\pi^{t-1}, i))}{\sum_j \exp(\eta \text{Reg}(\pi^{t-1}, j))} \\ &= \frac{\exp\left(\eta \sum_{s=1}^{t-1} (c_L^s - c_i^s)\right)}{\sum_j \exp\left(\eta \sum_{s=1}^{t-1} (c_L^s - c_j^s)\right)} \\ &= \frac{\exp\left(\eta \sum_{s=1}^{t-1} c_L^s\right) \cdot \exp\left(-\eta \sum_{s=1}^{t-1} c_i^s\right)}{\exp\left(\eta \sum_{s=1}^{t-1} c_L^s\right) \sum_j \exp\left(-\eta \sum_{s=1}^{t-1} c_j^s\right)} \\ &= \frac{\exp\left(-\eta \sum_{s=1}^{t-1} c_i^s\right)}{\sum_j \exp\left(-\eta \sum_{s=1}^{t-1} c_j^s\right)} \\ &= p_i^t \end{aligned}$$

Thus $p^t = q^t$, and so the result follows from Lemma B.4.5 and Theorem 76.

Bibliographic Notes and Further Reading

Multiplicative weights is a classic algorithm with a long history [??] — for a thorough introduction to several variants of the algorithm and its many applications see ?. A follow the perturbed leader like algorithm was developed by ?. Follow the Perturbed Leader, as developed in this chapter (applied to online linear optimization) is due to ?. The online convex optimization framework and the first analysis of online gradient descent is due to ?. The follow the regularized leader framework was introduced in ?. Our treatment of follow the regularized leader draws from Orabona’s notes here. The derivation of exponential weights using the minimax theorem follows Lee et al. [2022].