Omnipredictors

Parikshit Gopalan
VMware Research, Palo Alto, CA, USA

Adam Tauman Kalai
Microsoft Research, Boston, MA, USA

Omer Reingold
Stanford University, CA, USA

Vatsal Sharan
University of Southern California, Los Angeles, CA, USA

Udi Wieder
VMware Research, Palo Alto, CA, USA

Abstract

Loss minimization is a dominant paradigm in machine learning, where a predictor is trained to minimize some loss function that depends on an uncertain event (e.g., "will it rain tomorrow?"). Different loss functions imply different learning algorithms and, at times, very different predictors. While widespread and appealing, a clear drawback of this approach is that the loss function may not be known at the time of learning, requiring the algorithm to use a best-guess loss function. Alternatively, the same classifier may be used to inform multiple decisions, which correspond to multiple loss functions, requiring multiple learning algorithms to be run on the same data. We suggest a rigorous new paradigm for loss minimization in machine learning where the loss function can be ignored at the time of learning and only be taken into account when deciding an action.

We introduce the notion of an $(\mathcal{L},\mathcal{C})$-omnipredictor, which could be used to optimize any loss in a family $\mathcal{L}$. Once the loss function is set, the outputs of the predictor can be post-processed (a simple univariate data-independent transformation of individual predictions) to do well compared with any hypothesis from the class $\mathcal{C}$. The post processing is essentially what one would perform if the outputs of the predictor were true probabilities of the uncertain events. In a sense, omnipredictors extract all the predictive power from the class $\mathcal{C}$, irrespective of the loss function in $\mathcal{L}$.

We show that such "loss-oblivious" learning is feasible through a connection to multicalibration, a notion introduced in the context of algorithmic fairness. A multicalibrated predictor doesn't aim to minimize some loss function, but rather to make calibrated predictions, even when conditioned on inputs lying in certain sets c belonging to a family $\mathcal{C}$ which is weakly learnable. We show that a $\mathcal{C}$-multicalibrated predictor is also an $(\mathcal{L},\mathcal{C})$-omnipredictor, where $\mathcal{L}$ contains all convex loss functions with some mild Lipschitz conditions. The predictors are even omnipredictors with respect to sparse linear combinations of functions in $\mathcal{C}$. As a corollary, we deduce that distribution-specific weak agnostic learning is complete for a large class of loss minimization tasks.

In addition, we show how multicalibration can be viewed as a solution concept for agnostic boosting, shedding new light on past results. Finally, we transfer our insights back to the context of algorithmic fairness by providing omnipredictors for multi-group loss minimization.

2012 ACM Subject Classification Computing methodologies → Machine learning approaches

Keywords and phrases Loss-minimzation, multi-group fairness, agnostic learning, boosting

Digital Object Identifier 10.4230/LIPIcs.ITCS.2022.79

Related Version *Full version available at* *https://arxiv.org/abs/2109.05389*: https://arxiv.org/abs/2109.05389

Funding *Omer Reingold*: Most of the work performed while visiting VMware Research. Research supported in part by NSF Award IIS-1908774, The Simons Foundation collaboration on the theory of algorithmic fairness and The Sloan Foundation grant G-2020-13941.

Vatsal Sharan: Most of the work performed while at MIT.

1 Introduction

In machine learning, it is well-known that the best classifier may depend heavily on the choice of a loss function, and therefore correctly modeling the loss function is crucial for success in many applications. Modern machine learning libraries such as PyTorch, Tensorflow, and scikit-learn each offer a choice of over a dozen loss functions. However, this poses a challenge in applications where the loss is not known in advance or multiple losses may be used. For motivation, suppose you are training a binary classifier to predict whether a person has a certain medical condition ($y = 1$), such as COVID-19 or some heart condition, given their attributes x. The cost for misclassification may vary dramatically, depending on the application. For an infectious disease, the question of whether an individual should be allowed to go out for a stroll is different than whether a person should be allowed to go to work in a nursing home. Similarly, deciding on whether to advise a daily dose of aspirin carries very different risks than recommending cardiac catheterization. Furthermore, medical interventions that may be developed in the future may carry yet other, unforeseen, risks and benefits that may require retraining with a different loss function.

We adopt the following problem setup: we are given a distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$ where $\mathcal{X}$ is the domain and $\mathcal{Y}$ is the set of labels (for example $\mathcal{Y}$ can be $\{0, 1\}$ or $[0, 1]$). On each x, we take an action $t(x) \in \mathbb{R}$, and suffer loss $\ell(y, t(x))$ which depends on the label y and the action $t(x)$. We will refer to the function $t : \mathcal{X} \to \mathbb{R}$ which maps points in the domain to actions as the hypothesis. Our goal is to find a hypothesis that minimizes $\mathbf{E}_{\mathcal{D}}[\ell(\mathbf{y}, t(\mathbf{x}))]$ in comparison to some reference class of hypotheses.[1] We will refer to a function f which maps $\mathcal{X}$ to probability distributions over $\mathcal{Y}$ as a predictor. The goal of a predictor is to model the conditional distribution of labels $\mathbf{y}|\mathbf{x} = x$ for every point in the domain.

A classic example of different optimal hypotheses for different loss functions is the ℓ_2 vs. ℓ_1 losses which are minimized by the mean and median respectively. Consider a joint distribution $\mathcal{D}$ on $\mathcal{X} \times [0, 1]$, where $\mathbf{x} \in \mathcal{X}$ is a set of attributes (given) and $\mathbf{y} \in [0, 1]$ is an outcome. Consider an x such that the corresponding y is uniform in $[0.8, 1]$ with probability 0.6 and 0 with probability 0.4. In this case, to minimize the expected ℓ_2 loss $\ell(y, t) = (y - t)^2$ you would set $t(x) = 0.45$, whereas to minimize the expected ℓ_1 loss $\ell(y, t) = |y - t|$ you would set $t(x) = 0.83$. Not only is t different for the two losses, you cannot learn one from the other. In other words, learning to minimize the ℓ_2 loss looses information that is necessary to minimize the ℓ_1 loss and vice versa.

The phenomenon that there is no simple way to get one loss-minimizing predictor from another is not unique to ℓ_2 vs. ℓ_1 losses. Consider the distribution illustrated in Figure 1, which is known as a nested halfspace [20]. Consider a common and simple family of loss functions where $\ell(y, t) = c_y|y - t|$ where $y \in \{0, 1\}$ and $c_0, c_1 \geq 0$ are the costs of false positives and false negatives respectively. Even for linear classification, as the ratio c_0/c_1 varies, a different direction is optimal. The standard ML approach of minimizing a given loss would require separate classifiers for each loss. There is no clear way to infer the optimal classifier for one set of costs from the classifier for another; applying standard post-processing techniques, such as Platt Calibration [32] or Isotonic Regression [40], to the predictions so that $\Pr[y = 1|t = z] \approx z$ will not fix the issue since the optimal direction is different.

[1] Such a loss function that is the expectation of a loss for individual examples is called a decomposable loss in the literature.

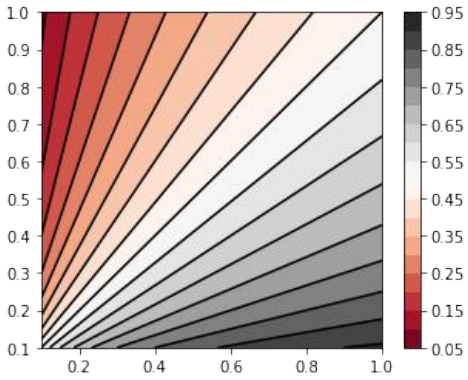

Figure 1 Binary classification with target function $\Pr[y = 1|x] = \frac{x_1}{x_1+x_2}$ for $x \in [0.1, 1]^2$. As can be seen from the level sets, the direction of the optimal linear classifier varies depending on the cost of false positives and negatives. This example is learned to near optimal loss for any loss with fixed costs of false-positives and false-negatives by an omnipredictor for the class $\mathcal{C} = \{x_1, x_2\}$.

1.1 Omnipredictors: one predictor to rule them all

This paper advocates a new paradigm for loss minimization: train a single predictor that could later be used for minimizing a wide range of loss functions, without having to further look at the data. Why should such predictors exist, and are they computationally tractable? One source of optimism is that the *ground-truth* predictor f^* does allow exactly that.

First consider the case of Boolean labels, and let $\mathcal{D}$ denote a distribution on $\mathcal{X} \times \{0, 1\}$ (our results also apply to real-valued outcomes). We define $f^*(x) = \mathbf{E}_{\mathbf{y}\sim\mathcal{D}}[\mathbf{y}|x] \in [0, 1]$ to be the conditional expectation of the label for $\mathbf{x}$. The value $t(x)$ which minimizes the loss $\mathbf{E}_{\mathbf{y}|x}[\ell(\mathbf{y}, t(x))]$ depends only on $f^*(x)$. Furthermore, as long as ℓ is smooth and easy to compute, t can easily be computed from f^*. We denote the univariate post-processing function that optimizes loss ℓ given true probabilities by $k^*_\ell : [0, 1] \to \mathbb{R}$. So for example for $\ell_2(y, t) = (y - t)^2$, we have $k^*_{\ell_2}(p) = p$ and for $\ell_1(y, t) = |y - t|$, $k^*_{\ell_1}(p) = \mathbb{1}(p \geq 1/2)$.

For every loss function ℓ, the composition of f^* and k^*_ℓ minimizes the loss ℓ, even conditioned on complete knowledge of $\mathcal{D}$. The connection between perfect predictors and choosing the optimal action is well understood and plays an important role in the Statistics literature on proper scoring rules and forecasting (cf. [35]). But learning f^* from samples from D is information-theoretically and computationally impossible in general. The natural approach is to learn a model f for f^* and then compose k^*_ℓ with f. Common instantiations of this approach (such as using logistic regression to model f) do not yield particularly strong guarantees in the realistic non-realizable setting where f^* does not come from the class of model distributions (see the further discussion in Section 2). **Our main conceptual contribution is to introduce the notion of Omnipredictors**, which provide a framework to derive strong rigorous guarantees using this composition approach even in the non-realizable setting.

The goal of an omnipredictor is to learn a predictor f that could replace f^* for the purpose of minimizing any loss from a class $\mathcal{L}$ compared to some hypothesis class $\mathcal{C}$. For a family $\mathcal{L}$ of loss functions, and a family $\mathcal{C}$ of hypotheses $c : \mathcal{X} \to \mathbb{R}$, we introduce the notion of an **$(\mathcal{L}, \mathcal{C})$-omnipredictor**, which is a predictor $f : \mathcal{X} \to [0, 1]$ with the property that for every loss function $\ell \in \mathcal{L}$, there is a post-processing function k_ℓ such that the expected loss of the composition $k_\ell \circ f$ measured using ℓ is almost as small as that of the best hypothesis $c \in \mathcal{C}$.

1.2 Omnipredictors for convex loss minimization

Omnipredictors can replace perfect predictors for the sake of minimizing loss in $\mathcal{L}$ compared with the class $\mathcal{C}$. In this sense they *extract all the predictive power* of $\mathcal{C}$ for such tasks. The key questions are of efficiency and simplicity: how strong a learning primitive do we need to assume to get an omnipredictor for $\mathcal{L}, \mathcal{C}$, and how complex is the predictor. The main result of this paper is that **one can efficiently learn simple omnipredictors for broad classes of loss functions and hypotheses, from weak learning primitives**.

Our main result

We show that for any hypothesis class $\mathcal{C}$, a *weak agnostic learner* for $\mathcal{C}$ is sufficient to efficiently compute an $(\mathcal{L}, \mathcal{C})$-omnipredictor where $\mathcal{L}$ consists of all decomposable convex loss functions obeying mild Lipshcitz conditions; it includes popular loss functions such as the ℓ_p losses for all p, the exponential loss and the logistic loss. Weak agnostic learnability captures a common modeling assumption in practice, and is a well-studied notion in the computational learning literature [4, 22, 19, 10]. The weak agnostic learning assumption says that if there is a hypothesis in $\mathcal{C}$ that labels the data reasonably well (say with 0-1 loss of 0.7), then we can efficiently find one that has a non-trivial advantage over random guessing (say with 0-1 loss of 0.51). In essence, our main result derives strong optimality bounds for a broad and powerful class of loss functions starting from a weak optimality condition for the 0-1 loss.

Perhaps surprisingly, our results are obtained not via the machinery of convex optimization, but by drawing a connection to work on fairness in machine learning, specifically the notion of multicalibration [16]. Multicalibration is a notion motivated by the goal of preventing unfair treatment of protected sub-populations in prediction; it does not explicitly consider loss minimization. We draw a connection to omnipredictors using a covariance-based recasting of the notion of multicalibration, that clarifies the connection of multicalibration to the literature on boosting [23, 18, 19]. A multicalibrated predictor satisfying this definition can be computed by a branching program, building on existing work in the literature on boosting [30, 23, 18] and multicalibration [14]. This new connection shows that the well-known boosting by branching programs algorithms [25, 30] yield multicalibrated predictors and can in fact be used to derive strong guarantees for a broad family of convex loss functions.

The post-processing function k_ℓ used in our positive results is essentially k^*_ℓ with small modifications. As the example in Figure 1 demonstrates, even in natural cases, an $(\mathcal{L}, \mathcal{C})$-omnipredictor cannot be a function in $\mathcal{C}$; in other words, the learning task we solve is inherently not proper.

Omnipredictors for larger classes

An advantage of our covariance-based notion of multicalibration is that it is closed under linear combinations. We use this to show that any multicalibrated predictor for $\mathcal{C}$ is in fact an $(\mathcal{L}, \mathrm{Lin}_{\mathcal{C}})$-omnipredictor where $\mathrm{Lin}_{\mathcal{C}}$ consists of linear combinations of functions in $\mathcal{C}$. We give negative results for slightly larger classes, showing that a multicalibrated predictor for $\mathcal{C}$ is not necessarily an omnipredictor for the class $\mathrm{Thr}_{\mathcal{C}}$ which consist of thresholds of functions in $\mathcal{C}$. Similarly, it need not be an omnipredictor for the class $\mathcal{C}$ but with non-convex loss functions. This shows that the connection between multicalibration and omnipredictors that we present is fairly tight.

Omnipredictors for multi-group loss minimization

A very recent line of research defines multi-group notions of loss minimization [5, 33].[2] Multi-group loss minimization is well motivated from the point of view of fairness as it guarantees that no sub-population's loss is sacrificed for the sake of global loss minimization. Say we have a collection of sub-populations $\mathcal{T}$ and hypotheses $\mathcal{P}$ and seek to take actions such that for every set $T \in \mathcal{T}$, our actions can compare with the best hypothesis in $P \in \mathcal{P}$ for that sub-population T. Indeed, one may wish to vary the loss function for various subgroups: say in a medical scenario where different age-groups are known to react differently to the same treatment.

We derive strong multi-group loss minimization guarantees using the closure of multicalibration under conditioning on subsets in $\mathcal{C}$. We show that in the scenario above, a multicalibrated predictor for $\mathcal{T} \times \mathcal{P} = \{T \cdot P, T \in \mathcal{T}, P \in \mathcal{P}\}$ gives an $(\mathcal{L}, \mathcal{P})$-omnipredictor for every sub-population $T \in \mathcal{T}$. Hence given a sub-population $T \in \mathcal{T}$ and a loss $\ell \in \mathcal{L}$, the predictions of the omnipredictor can be post-processed to be competitive with the best hypothesis from $\mathcal{P}$ for that loss function ℓ and sub-population T.

Omnipredictors for real-valued labels

We extend the notion of omnipredictors to the setting where the labels come from an arbitrary subset $\mathcal{Y} \subseteq \mathbb{R}$. Our primary interest is in multi-class prediction where $\mathcal{Y} = [k]$ and the bounded real-valued setting where $\mathcal{Y} = [0, 1]$. We show that omnipredictors can be learned in this setting, for similar families of loss functions, again assuming weak learnability of $\mathcal{C}$. We also show a stronger bound for the ℓ_2 loss than what the general theorem implies.

1.3 Multicalibration and agnostic boosting

One of our contributions in this work is to formalize and leverage the connections between multicalibration and the literature on agnostic boosting. We propose a covariance-based recasting of the notion of multicalibration, inspired by the literature on boosting [18, 19]. We show that this definition has several advantages, for instance it implies some general closure properties for multicalibration. In the other direction, our work suggests **multicalibration as a solution concept for agnostic boosting**. By specializing our main result on omnipredictors to the ℓ_1 loss, we derive a new proof of the classic result of [21] on agnostic learning. We elaborate on these connections in this subsection.

A covariance-based formulation of multicalibration

Calibration has been well-studied in the statistics literature in the context of forecasting [7]. It was introduced to the algorithmic fairness literature by [28]. In the setting of Boolean labels, we are given a distribution $\mathcal{D}$ on $\mathcal{X} \times \{0, 1\}$ of labelled examples, and wish to learn a predictor $f : \mathcal{X} \to [0, 1]$, where $f(x)$ is our model for $\mathbf{E}_{\mathcal{D}}[\mathbf{y}|x]$. The predictor f is (approximately) calibrated if for every value v in its range we have that $\Pr[y = 1|f(x) = v] \approx v$. This means that the prediction $f(x)$ can be interpreted as a probability that is correct in expectation over individuals in the same level set of f. By itself, calibration is a very weak property, both in terms of fairness as well as in terms of accuracy. This motivated [16] to introduce the notion of multicalibration that asks for f to be calibrated on a rich collection of subgroups $\mathcal{C}$ rather than just a few protected sets.

[2] The notion of loss in [33] is more general than in this paper and includes global functions of loss rather than the expectation of loss on individual elements. See further comparison in Section 2.

We draw a connection to omnipredictors by introducing a covariance-based recasting of the notion of multicalibration, that clarifies the connection to the literature on boosting [18, 23, 19]. Prior definitions consider multicalibration for a family $\mathcal{C}$ of sets or equivalently Boolean functions $c : \mathcal{X} \to \{0,1\}$, whereas we allow arbitrary real-valued functions. Rather than work with predictors, we define multicalibration for partitions, inspired by the recent work of [14] in the unsupervised setting. A partition $\mathcal{S} = \{S_1, \ldots, S_m\}$ of the domain is a collection of disjoint subsets whose union is $\mathcal{X}$. Intuitively, we want these sets to be (approximate) level sets for f^*. We let $\mathcal{D}_i$ denote the distribution $\mathcal{D}$ conditioned on $\mathbf{x} \in S_i$. The partition $\mathcal{S}$ gives a *canonical predictor* $f^{\mathcal{S}}$ where for each $x \in S_i$, we predict $f^{\mathcal{S}}(x) = \mathbf{E}_{\mathcal{D}_i}[\mathbf{y}]$. In analogy to boosting (see, e.g., [18, 23, 19]), we phrase multicalibration in terms of the covariance[3] between $c(\mathbf{x})$ and $\mathbf{y}$ conditioned on each state S_i of the partition. We say that $\mathcal{S}$ is α-multicalibrated for $\mathcal{C}$ if for every $i \in [m]$ and $c \in \mathcal{C}$,

$$\left|\underset{\mathcal{D}_i}{\mathbf{Cov}}[c(\mathbf{x}), \mathbf{y}]\right| \leq \alpha. \tag{1}$$

In reality, we will weaken the definition to only hold in expectation under $\mathcal{D}$ rather than require it for every state of the partition, but we ignore this distinction for now. While our definition is formulated differently, we show that it matches the original definition when $\mathcal{C}$ consists of Boolean functions (see the full version of the paper). This lets us adapt existing algorithms in the literature [30, 18, 14] to give an efficient procedure to compute multicalibrated partitions, assuming a weak agnostic learner for the class $\mathcal{C}$. Working with covariance which is bilinear lets us derive powerful closure properties for multicalibration. For instance, if f is multicalibrated with respect to $\mathcal{C}$ it is also multicalibrated (with some deterioration in parameters) with respect to the class $\mathrm{Lin}_{\mathcal{C}}$ of (sparse) linear combinations of $\mathcal{C}$. We also show that conditioning on sets in $\mathcal{C}$ preserves multicalibration.

Agnostic boosting from multicalibration

The problem of agnostically learning a class $\mathcal{C}$ is, given samples from a distribution $\mathcal{D}$ on $\mathcal{X} \times \{0,1\}$, find a binary classifier $f : \mathcal{X} \to \{0,1\}$ whose classification error aka 0-1 loss defined as $\mathrm{err}(f) = \Pr_{\mathcal{D}}[f(\mathbf{x}) \neq \mathbf{y}]$ is not much larger than that of the best classifier from $\mathcal{C}$. The boosting approach to agnostic learning is to start from a weak agnostic learner, which only guarantees some non-trivial correlation with the labels, and boost it to obtain a classifier that agnostically learns $\mathcal{C}$ [4, 22, 10].

Our work suggests multicalibration as a solution concept for agnostic boosting. Indeed, our definition of multicalibration based on covariance parallels that of [18, 23], who use covariance as a splitting criterion. Hence when the algorithms of [30, 18, 23] terminate, they have found a multicalibrated partition. Viewed in this light, our results show that these algorithms give a broad and powerful guarantee beyond just 0-1 loss: they are competitive with sparse linear combinations over $\mathcal{C}$ in optimizing a large family of convex, Lipschitz loss functions (with a simple post-processing step). While AdaBoost or Logistic regression are known to minimize the exponential and logistic loss respectively over sparse linear combinations of $\mathcal{C}$ [37], no similar result was known for algorithms based on branching programs [30, 18, 23].

Let us now focus on 0-1 loss. Since 0-1 loss for Boolean functions equals ℓ_1 loss, our results apply to it. We show that for any multicalibrated partition $\mathcal{S}$, the predictor $k_{\ell_1} \circ f^{\mathcal{S}}$ is competitive not just with the best classifier in $\mathcal{C}$, but with the best classifier in the larger

[3] Recall that $\mathbf{Cov}[\mathbf{z}_1\mathbf{z}_2] = \mathbf{E}[\mathbf{z}_1\mathbf{z}_2] - \mathbf{E}[\mathbf{z}_1]\,\mathbf{E}[\mathbf{z}_2]$

class $\mathcal{H}$ of functions that are approximated (in ℓ_1) by linear combinations of $\mathcal{C}$. This lets us re-derive the classic result of [21] on agnostic learning. While not a new result, we feel that our treatment clarifies and unifies existing results (see the full version of the paper). It strengthens known results on the noise-tolerant boosting abilities of such programs [23, 22]. Further, we show examples where $\mathrm{err}(k_{\ell_1} \circ f^{\mathcal{S}})$ is markedly better than any linear combination of $\mathcal{C}$, showing that multicalibration is a stronger solution concept than those considered previously.

1.4 Technical overview: Omnipredictors from multicalibration

Let $\ell : \{0,1\} \times \mathbb{R} \to \mathbb{R}$ be a loss function that takes a label $y \in \mathcal{Y}$ and an action $t \in \mathbb{R}$ as arguments. A hypothesis $t : \mathcal{X} \to \mathbb{R}$ which prescribes an action for every point in the domain suffers a loss of $E_{\mathcal{D}}[\ell(\mathbf{y}, t(\mathbf{x})]$. Our main technical result states that if $\mathcal{S}$ is multicalibrated, then $f^{\mathcal{S}}$ is an $(\mathcal{L}, \mathcal{C})$-omnipredictor where $\mathcal{L}$ consists of all convex losses satisfying some mild Lipschitz conditions. Here for simplicity, we make the stronger assumption that the loss $\ell(y,t)$ is convex and Lipschitz everywhere as a function of t. We do not assume anything about the relation between $\ell(0,t)$ and $\ell(1,t)$.

We sketch how multicalibration leads $f^{\mathcal{S}}$ to be an omnipredictor, emphasizing intuition over rigor. We will argue that the loss $\mathbf{E}_{\mathcal{D}}[\ell(\mathbf{y}, k^*_\ell(f^{\mathcal{S}}(\mathbf{x}))]$ is not much more than $\mathbf{E}_{\mathcal{D}}[\ell(\mathbf{y}, c(\mathbf{x}))]$ for any $c \in \mathcal{C}$. We fix a state $S_i \in \mathcal{S}$ and analyze the loss suffered by $c(x)$ under $\mathcal{D}_i$ as follows.

1. **Reduction to predicting two values:** In general, $c(x)$ could take on many values under $\mathcal{D}_i$. However, since the goal is to minimize $\mathbf{E}_{\mathcal{D}_i}[\ell(\mathbf{y}, c(\mathbf{x}))]$, we can *pretend* that c takes only two values, $\mathbf{E}_{\mathcal{D}_i|\mathbf{y}=0}[c(\mathbf{x})]$ whenever $\mathbf{y}=0$ and $\mathbf{E}_{\mathcal{D}_i|\mathbf{y}=1}[c(\mathbf{x})]$ whenever $\mathbf{y}=1$ (we say pretend since the actions taken can only depend on x and not on y). By the *convexity* of the loss functions, this can only reduce the expected loss.
2. **Reduction to predicting one value:** A consequence of multicalibration, which follows from the definition of covariance, is that conditioning on the label $\mathbf{y}=b$ does not change the expectation of $c(\mathbf{x})$ much. Formally for $b \in \{0,1\}$,

$$\Pr_{\mathcal{D}_i}[\mathbf{y}=b] \left| \mathop{\mathbf{E}}_{\mathcal{D}_i|\mathbf{y}=b}[c(\mathbf{x})] - \mathop{\mathbf{E}}_{\mathcal{D}_i}[c(\mathbf{x})] \right| \leq \alpha. \tag{2}$$

   Since the loss functions $\ell(b,t)$ are *Lipschitz* in t for $b \in \{0,1\}$, we can replace $\mathbf{E}_{\mathcal{D}_i|\mathbf{y}=b}[c(\mathbf{x})]$ with $\mathbf{E}_{\mathcal{D}_i}[c(\mathbf{x})]$ with only a small increase in the loss. At this point, we have reduced to the case where c predicts the constant value $\mathbf{E}_{\mathcal{D}_i}[c(\mathbf{x})]$ under $\mathcal{D}_i$.
3. **The best value:** Let $\mathbf{E}_{\mathcal{D}_i}[\mathbf{y}] = p_i$, thus $\mathbf{y}$ is distributed as a Bernoulli random variable with parameter p_i. Thus, the best single value to predict is $k^*_\ell(p_i)$, which is the minimizer of the expected loss $p_i\ell(0,t) + (1-p_i)\ell(1,t)$. But we defined $f^{\mathcal{S}}(x) = p_i$ for all $x \in S_i$, so $k^*_\ell(p_i) = k^*_\ell(f^{\mathcal{S}}(x))$.

We conclude that post-processing the predictions $f^{\mathcal{S}}$ by the function k^*_ℓ is nearly as good as any $c \in \mathcal{C}$ for minimizing expected loss under $\mathcal{D}$ for any convex, Lipschitz loss function ℓ (up to an additive error that goes to 0 with α). Hence $f^{\mathcal{S}}$ is an $(\mathcal{L}, \mathcal{C})$-omnipredictor. As a consequence, having a weak agnostic learner for $\mathcal{C}$ suffices to learn a predictor that can minimize any such loss function competitively to predictors in $\mathcal{C}$, even without knowing the loss function in advance.

1.5 Organization of this paper

We survey related work in Section 2, and set up notation in Section 3. We define the notion of omnipredictors in Section 4. We introduce our notion of multicalibration in Section 5 and derive closure properties for it in Section 5.2. We prove our main result on $(\mathcal{L}, \mathcal{C})$-omnipredictors for binary labels (Theorem 19) in Section 6, along with the extensions to $\mathrm{Lin}_{\mathcal{C}}$ (Corollary 21), and its application to multi-group loss minimization (Corollary 22). Section 6.3 shows that multicalibration for $\mathcal{C}$ does not yield omnipredictors for thresholds of $\mathcal{C}$ or for non-convex losses. The full version of the paper [13] has further results. It also presents applications of our results to agnostic learning, including an example showing that multicalibration can give stronger guarantee than $\mathrm{OPT}(\mathrm{Lin}_{\mathcal{C}})$. The full version presents the extension to the real valued setting, where we derive a stronger bound for ℓ_2 loss. The full version also gives a detailed discussion of how our definition compares to previous definitions. Finally, the full version presents the algorithm for computing multicalibrated partitions.

2 Related work

While the notion of an omnipredictor is introduced in this work, our definitions draw on two previous lines of work. The first is the notion of multicalibration for predictors that was defined in the work of Hebert-Johnson *et al.* [16].[4] A detailed discussion of how our definitions of multicalibration compare to previous definitions appears in the full version of the paper [13]. The other is work on boosting by branching programs of Mansour-McAllester [30], which built on Kearns-Mansour [25] and the notion of correlation boosting [18, 23].

Group fairness and multicalibration

While multicalibration was introduced with the motivation of algorithmic fairness, it has been shown to be quite useful from the context of accuracy when learning in an heterogeneous environment. This was done both experimentally [27, 1] and in real-life implementations [2]. From a theoretical perspective, it has been shown in [16] that post-processing a predictor to make it multicalibrated cannot increase the ℓ_2 loss. This was extended to showing some optimality results of multicalibrated predictors with respect to ℓ_2 and even log-loss [12, 26]. Multicalibrated predictors are also connected to loss minimization through the notion of outcome-indistinguishability [9] in the work of [33]. Outcome indistinguishability shows that a multicalibrated predictor is indistinguishable from the true probabilities predictor in a particular technical sense. In [33] this is used to create a predictor that can be used to minimize a rather general and potentially global notion of loss even when restricted to sub-groups. The proof constructs a family of distinguishers, for a fixed loss function, such that if there exists a subset on which the predictor doesn't minimize the loss function then one of the predictors can distinguish the predictor from the true-probabilities predictor in the sense of outcome indistinguishability. The main way in which all of these results are different from what we show is that they do not seek to simultaneously allow for the minimization of such a rich family of loss functions. In the case of [33], since the result goes through outcome indistinguishability, to minimize a loss with respect to a class $\mathcal{C}$, a predictor needs to be multicalibrated with respect to a different class that relies on the loss function and incorporates the reduction from multicalibration and outcome indistinguishability. In

[4] See also [24], who in parallel with [16] introduced notions of multi-group fairness.

contrast, our result addresses minimization of a family of losses that may not be known at the time of learning, and we assume the learnability of $\mathcal{C}$ alone. Convexity of the losses plays a key role in our upper bounds.

Agnostic boosting

Our work is closely related to work on boosting via branching programs [25, 30, 23, 18]. Indeed, the splitting criterion used in the work of Kalai [18] is precisely that $\mathbf{Cov}[c(\mathbf{x}), \mathbf{y}] \geq \alpha$ for some $c \in \mathcal{C}$, for the task of learning generalized linear models. The *okay learners* in the work of [23] are also based on having non-trivial Covariance with the target. Our results show that upon termination, these algorithms yields an approximately multicalibrated partition; hence we can view multicalibration as a solution concept for the output of Boosting by Branching Program family of algorithms. It was known that these algorithms have stronger noise tolerance properties than potential based algorithms such as AdaBoost, see [23, 22]. Our results significantly strengthen our understanding of the power of these algorithms, showing that they give guarantees for a broad family of convex loss functions, and not just 0-1 loss.

Naive instantiations of the composition approach

The obvious attempt to building omnipredictors would be to learn a model f for f^* using a model family $\mathcal{F}$ such as logistic regression, and then compose it with the right post-processing function k^*_ℓ for a given loss ℓ. In the context of binary classification, for certain families of non-decomposable accuracy metrics (including F-scores and AUC), it is shown in [31, 8] that this approach gives the best predictor for the hypothesis class $\mathcal{C}$ of binary classifiers derived by thresholding models from $\mathcal{F}$.

These results can be seen as a form of omnipredictors, but with strong restrictions on $\mathcal{L}$ and $\mathcal{C}$. Their results do not apply to the decomposable convex losses we consider; indeed it seems unlikely that the output of logistic regression can give reasonable guarantees for say the exponential loss or squared loss, even with post-processing. More importantly, our results hold for arbitrary classes $\mathcal{C}$ that are weakly agnostically learnable. For any such class, we show how to construct a model f which is an omnipredictor. In contrast, their results prove optimality for a rather limited hypothesis class $\mathcal{C}$ derived from the model family $\mathcal{F}$.

Condtional density estimation

For real-valued $y \in \mathbb{R}$, an omnipredictor solves the problem of *Conditional Density Estimation* (CDE) [15]. While CDE is recognized as an important problem in practice and a number of CDE algorithms have been proposed, it has received little attention in the computational learning theory literature. The notion of omnipredictor is related to the statistical notion of a *sufficient statistic*, which is a statistic that captures all relevant information about a distribution.

Surrogate loss functions for classification

There is a large literature in statistics which shows that a convex surrogate loss function (with certain properties) can be used instead of the hard to optimize 0-1 loss, and any hypothesis $c \in \mathcal{C}$ which minimizes the surrogate loss will also minimize the original 0-1 loss with respect to $\mathcal{C}$ [29, 38, 3]. There are also similar results for the multi-class 0-1 loss [39], and in the asymmetric setting when the false positive and false negative costs are known [36]. However,

this line of work is quite different from ours, crucially an $(\mathcal{L},\mathcal{C})$-omnipredictor optimizes not just for a single loss but for any loss in the family $\mathcal{L}$ (such as different false positive and false negative costs, we recall that as Fig. 1 shows this cannot be achieved with a single hypothesis from $\mathcal{C}$).

3 Notation and Preliminaries

Let $\mathcal{D}$ denote a distribution on $\mathcal{X} \times \mathcal{Y}$. The set $\mathcal{X}$ represents points in our space, it could be continuous or discrete. The set $\mathcal{Y}$ represents the labels, we will typically consider $\mathcal{Y} = \{0,1\}$, $\mathcal{Y} = [k]$ or $\mathcal{Y} = [0,1]$. We use $(\mathbf{x},\mathbf{y}) \sim \mathcal{D}$ where $\mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}$ to denote a sample from $\mathcal{D}$. We use boldface for random variables. For any $S \subset \mathcal{X}$, let $\mathcal{D}(S) = \Pr_{\mathbf{x}\sim\mathcal{D}}[\mathbf{x} \in S]$. We will use $\mathbf{y} \sim \mathrm{Ber(p)}$ to denote sampling from the Bernoulli distribution with parameter p.

Let $\mathcal{C} = \{c : \mathcal{X} \to \mathbb{R}\}$ be a collection of real-valued hypotheses on a domain $\mathcal{X}$, which could be continuous or discrete. The hypotheses in $\mathcal{C}$ should be efficiently computable, and the reader can think of them as monomials, decision trees or neural nets. We will denote

$$\|\mathcal{C}\|_\infty = \max_{c\in\mathcal{C}, x\in\mathcal{X}} |c(x)|.$$

A loss function ℓ takes a label $y \in \mathcal{Y}$, an action $t \in \mathbb{R}$ and returns a loss value $\ell(y,t)$. Common examples are the ℓ_p losses $\ell_p(y,t) = |y-t|^p$ and logistic loss $\ell(y,t) = \log(1+\exp(-yt))$. The problem of minimizing a loss function ℓ is to learn a hypotheses $h : \mathcal{X} \to \mathbb{R}$ such that the expected loss $\ell_{\mathcal{D}}(h) := \mathbf{E}_{\mathcal{D}}[\ell(\mathbf{y}, h(\mathbf{x}))]$ is small. Let $\mathcal{L} = \{\ell : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}\}$ denote a collection of loss functions.

A partition $\mathcal{S} = \{S_1, \ldots, S_m\}$ of the domain $\mathcal{X}$, is a collection of disjoint subsets whose union equals $\mathcal{X}$, we refer to m as its size. We refer to each S_i as a state in the partition. Given a partition $\mathcal{S} = \{S_1, \ldots, S_m\}$ of the domain $\mathcal{X}$, define the conditional distribution $\mathcal{D}_i$ over $S_i \times \mathcal{Y}$ as $\mathcal{D}_i = \mathcal{D}|\mathbf{x} \in S_i$.

A (binary) predictor is a function $f : \mathcal{X} \to [0,1]$, where $f(x)$ is interpreted as the probability conditioned on x that $\mathbf{y} = 1$. We define the ground truth predictor as $f^*(x) := \mathbf{E}_{\mathcal{D}}[\mathbf{y}|\mathbf{x} = x]$. For general label sets $\mathcal{Y}$, let $\mathcal{P}(\mathcal{Y})$ denote the space of probability distributions on $\mathcal{Y}$. Define the ground truth predictor $f^* : \mathcal{X} \to \mathcal{P}(\mathcal{Y})$ where $f^*(x)$ is the distribution of $\mathbf{y}|x$. A predictor is a function $f : \mathcal{X} \to \mathcal{P}(\mathcal{Y})$ which is intended to be an approximation of f^*.

We denote by $g \circ h$ the composition of functions.

3.1 Nice loss functions

We say $\ell : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$ is a convex loss function, if $\ell(y,t)$ is a convex function of t for every $y \in \mathcal{Y}$. Note that in the binary setting, our formulation allows for binary classification with different false-positive/negative costs, e.g., $\ell(y,t) = c_y|t-y|$ where c_0 and c_1 are the different costs. A function $f : \mathbb{R} \to \mathbb{R}$ is said to be B-Lipschitz over interval I if $|f(t) - f(t')| \le B|t-t'|$ for all $t, t' \in I$. We say the function is B-Lipschitz if the condition holds for $I = \mathbb{R}$. While assuming the loss is B-Lipschitz is sufficient for us, we can work with a weaker notion that only requires the Lipschitz property on a sufficiently large interval. This weaker notion covers most commonly used loss functions such as the exponential loss that are not Lipschitz everywhere. Also, we will define loss functions in the setting of labels that come from $\mathcal{Y}$. In this section though, we will focus on the case $\mathcal{Y} = \{0,1\}$.

▶ **Definition 1.** *For $B, \epsilon > 0$, a convex loss function $\ell : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$ is (B,ϵ)-nice if there is a closed interval $I = I_\ell \subseteq \mathbb{R}$ satisfying:*

1. **(B-Lipschitzness)** *For all y, $\ell(y,t)$ is B-Lipschitz in t over I_ℓ:*

$$\forall y \in \mathcal{Y},\ t, t' \in I_\ell, |\ell(y,t) - \ell(y,t')| \le B|t-t'|.$$

2. **(ϵ-optimality)** *For $y \in \mathcal{Y}$, I_ℓ contains an ϵ-optimal minimizer of $\ell(y,t)$:*

$$\inf_{t \in I_\ell} \ell(y,t) \leq \inf_{t \in \mathbb{R}} \ell(y,t) + \epsilon.$$

Let $\mathcal{L}(B,\epsilon)$ be the set of all (B,ϵ)-nice functions.

Observe that if a function is B-Lipschitz on $\mathbb{R}$, then it is $(B,0)$-nice with $I_\ell = \mathbb{R}$.

For a closed interval $I = [c,d]$ define the function

$$\mathrm{clip}(t,I) = \begin{cases} c \text{ if } t \leq c \\ t \text{ if } c \leq t \leq d \\ d \text{ if } d \leq t. \end{cases}$$

We use some simple facts about this function without proof, the first is a simple consequence of ϵ-optimality and convexity.

▶ **Lemma 2.** *If ℓ is (B,ϵ)-nice, then $\ell(y, \mathrm{clip}(t, I_\ell)) \leq \ell(y,t) + \epsilon$. The function $\mathrm{clip}(t, I_\ell)$ is 1-Lipschitz as a function of t.*

Here are a few examples of nice loss functions:

- Binary classification with different false-positive/negative costs, e.g., $\ell(y,t) = \kappa_y |y - t|$ where $\kappa_0 \neq \kappa_1$ are the different costs. Here $I_\ell = [0,1], B = \max(\kappa_0, \kappa_1), \epsilon = 0$.
- The ℓ_p losses for $p \geq 1$:

$$\ell_p(y,t) := |y - t|^p$$

  take $I_\ell = [0,1], B = p, \epsilon = 0$.
- The exponential loss $\ell(y,t) = e^{(1-2y)t}$. Here, for any $\epsilon > 0$, we can take $I_\ell = [-\ln(1/\epsilon), \ln(1/\epsilon)]$ and $B = 1/\epsilon$.
- The logistic loss $\ell(y,t) = \log(1 + \exp((1-2y)t))$. Here we take $I_\ell = \mathbb{R}$, $B = 1$ and $\epsilon = 0$.
- The hinge loss $\ell(y,t) = \max(0, 1 + (1-2y)t)$. Here we take $I_\ell = \mathbb{R}$, $B = 1$ and $\epsilon = 0$.

It is worth noting that only for exponential loss did we need $\epsilon > 0$.

4 Omnipredictors

In this section, we define our notion of Omnipredictors. Our definitions are simpler for the case of binary labels where $\mathcal{Y} = \{0,1\}$, hence we present that case first. Recall that for a predictor h, $\ell_{\mathcal{D}}(h)$ is the expected loss of h under $\mathcal{D}$.

▶ **Definition 3** (Omnipredictor)**.** *Let $\mathcal{C}$ be family of functions on $\mathcal{X}$, and let $\mathcal{L}$ be a family of loss functions. The predictor $f : \mathcal{X} \to [0,1]$ is an $(\mathcal{L}, \mathcal{C}, \delta)$-omnipredictor if for every $\ell \in \mathcal{L}$ there exists a function $k : [0,1] \to \mathbb{R}$ so that*

$$\ell_{\mathcal{D}}(k \circ f) \leq \min_{c \in \mathcal{C}} \ell_{\mathcal{D}}(c) + \delta.$$

The definition states that for every loss ℓ, there is a simple (univariate) transformation k of the predictions f, such that the composition $k \circ f$ has loss comparable to the best hypothesis $c \in \mathcal{C}$, which is chosen tailored to the loss ℓ.

Setting aside efficiency considerations, it is easy to show that f^* is an omnipredictor for every $\mathcal{C}, \mathcal{L}$.

▶ **Lemma 4.** *For every $\mathcal{C}, \mathcal{L}$, the ground-truth predictor f^* is an $(\mathcal{L}, \mathcal{C}, 0)$-omnipredictor.*

Proof. By the definition of omnipredictors, given an arbitrary loss function $\ell : \{0,1\} \times \mathbb{R} \to \mathbb{R}$, our goal is to find $k^*_\ell : \{0,1\} \to \mathbb{R}$ so that

$$\ell_{\mathcal{D}}(k^*_\ell \circ f) \leq \min_{c:\mathcal{X}\to\mathbb{R}} \ell_{\mathcal{D}}(c). \tag{3}$$

Define the function $k^*_\ell : [0,1] \to \mathbb{R}$ which minimizes expected loss under the Bernoulli distribution:

$$k^*_\ell(p) = \arg\min_{t\in\mathbb{R}} \mathop{\mathbf{E}}_{\mathbf{y}\sim \mathrm{Ber(p)}}[\ell(\mathbf{y}, t)] = \arg\min_{t\in\mathbb{R}} p\ell(1,t) + (1-p)\ell(0,t). \tag{4}$$

If there are multiple minima we break ties arbitrarily. Conditioned on $\mathbf{x} = x$, $\mathbf{y} \sim \mathrm{Ber(f^*(x))}$, so $k^*_\ell(f^*(x)) \in \mathbb{R}$ is the value that minimizes the expected loss. Hence for every $x \in \mathcal{X}$,

$$\mathop{\mathbf{E}}_{\mathcal{D}|\mathbf{x}=x}[\ell(\mathbf{y}, k^*_\ell(f^*(x)))] \leq \mathop{\mathbf{E}}_{\mathcal{D}|\mathbf{x}=x}[\ell(\mathbf{y}, c(x))].$$

Equation (3) follows by averaging over all values of x. ◀

Note that the function k^*_ℓ depends on ℓ but is independent of the distribution $\mathcal{D}$. For instance, for the ℓ_1 loss, $\ell_1(y,t) = |y-t|$, we have $k^*_{\ell_1}(p) = \mathbb{1}(p \geq 1/2)$. For the ℓ_2 loss $\ell_2(y,t) = (y-t)^2$, we have $k^*_{\ell_2}(p) = p$. While Definition 3 does not place any restrictions on the post-processing function k, in our upper bounds, we will choose k which is very close to the function k^* above. Our upper bounds will be efficient for (convex) functions ℓ such that a good approximation to k^* can be be approximated efficiently. Our lower bounds will hold for arbitrary k.

Finally, a natural family of predictors arising from partitions plays a key role in our results.

▶ **Definition 5.** *Given a partition $\mathcal{S}$ of $\mathcal{X}$ of size m, let $\mathbf{E}_{\mathcal{D}_i}[\mathbf{y}] = p_i \in [0,1]$ for $i \in [m]$. The canonical predictor for $\mathcal{S}$ is $f^{\mathcal{S}}(x) = p_i$ for all $x \in S_i$.*

The canonical predictor simply predicts the expected label in each state of the partition. Since $f^{\mathcal{S}}$ is constant within each state of the partition, it can be viewed as a function $f^{\mathcal{S}} : \mathcal{S} \to \mathbb{R}$. This view will be useful in our results.

Omnipredictors for general $\mathcal{Y}$

Consider the setting where we are given a distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$ for $\mathcal{Y} \subseteq \mathbb{R}$, hence the labels can take on real values. We are primarily interested in the real-valued setting of $\mathcal{Y} = [0,1]$, and the multi-class setting where $\mathcal{Y} = [l]$.

Given a state S_i in a partition $\mathcal{S}$, let $P_i \in \mathcal{P}(\mathcal{Y})$ denote the distribution of $\mathbf{y}$ under $\mathcal{D}_i$. The canonical predictor $f^{\mathcal{S}} : \mathcal{X} \to \mathcal{P}(\mathcal{Y})$ is given by $f^{\mathcal{S}}(x) = P_i$ for all $x \in S_i$.

We now define the notion of an omnipredictor. The main difference from the binary case is that the predictor now predicts a distribution in $\mathcal{P}(\mathcal{Y})$, so the post-processing function k maps *distributions* to real values.

▶ **Definition 6.** *Let $\mathcal{C}$ be family of functions on $\mathcal{X}$, and let $\mathcal{L}$ be a family of loss functions $\ell : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}$. The predictor $f : \mathcal{X} \to \mathcal{P}(\mathcal{Y})$ is an $(\mathcal{L}, \mathcal{C}, \delta)$-omnipredictor if for every $\ell \in \mathcal{L}$ there exists a function $k : \mathcal{P}(\mathcal{Y}) \to \mathbb{R}$ so that*

$$\ell_{\mathcal{D}}(k \circ f) \leq \min_{c\in\mathcal{C}} \ell_{\mathcal{D}}(c) + \delta.$$

5 Multicalibration

In this section, we present our definitions of multicalibration. We first consider the case of binary labels, and then extend it to the multi-class and real-valued settings. Due to space limitations, the proofs are to be found in the full version of the paper. Given real-valued random variables $\mathbf{z}_1, \mathbf{z}_2$ from a joint distribution $\mathcal{D}$, we define

$$\underset{\mathcal{D}}{\mathbf{Cov}}[\mathbf{z}_1, \mathbf{z}_2] = \underset{\mathcal{D}}{\mathbf{E}}[\mathbf{z}_1\mathbf{z}_2] - \underset{\mathcal{D}}{\mathbf{E}}[\mathbf{z}_1]\underset{\mathcal{D}}{\mathbf{E}}[\mathbf{z}_2] = \underset{\mathcal{D}}{\mathbf{E}}\left[\mathbf{z}_1(\mathbf{z}_2 - \underset{\mathcal{D}}{\mathbf{E}}[\mathbf{z}_2])\right].$$

We will use the fact that covariance is bilinear. The following identity will be useful for Boolean $\mathbf{y}$.

▶ **Corollary 7.** *For random variables $(\mathbf{z}, \mathbf{y}) \sim \mathcal{D}$ where $\mathbf{y} \in \{0, 1\}$,*

$$\underset{\mathcal{D}}{\mathbf{Cov}}[\mathbf{z}, \mathbf{y}] = \Pr_{\mathcal{D}}[\mathbf{y} = 1]\left(\underset{\mathcal{D}|\mathbf{y}=1}{\mathbf{E}}[\mathbf{z}] - \underset{\mathcal{D}}{\mathbf{E}}[\mathbf{z}]\right) = \Pr_{\mathcal{D}}[\mathbf{y} = 0]\left(\underset{\mathcal{D}}{\mathbf{E}}[\mathbf{z}] - \underset{\mathcal{D}|\mathbf{y}=0}{\mathbf{E}}[\mathbf{z}]\right). \tag{5}$$

5.1 Multicalibration via covariance

In this section, we define multicalibration for the binary labels setting where $\mathcal{Y} = \{0, 1\}$. We build on a recent line of work [16, 17, 27, 14]. A detailed discussion of how our definitions compare to previous definitions is presented in the full version of the paper. The following definition is a generalization of the notion of α-multicalibration to real-valued c, and in the setting of partitions.

▶ **Definition 8.** *Let $\mathcal{D}$ be a distribution on $\mathcal{X}\times\{0, 1\}$. The partition $\mathcal{S}$ of $\mathcal{X}$ is α-multicalibrated for $\mathcal{C}, \mathcal{D}$ if for every $i \in [m]$ and $c \in \mathcal{C}$, the conditional distribution $\mathcal{D}_i = \mathcal{D}|\mathbf{x} \in S_i$ satisfies*

$$\left|\underset{\mathcal{D}_i}{\mathbf{Cov}}[c(\mathbf{x}), \mathbf{y}]\right| \leq \alpha. \tag{6}$$

A consequence of this definition is that for each $\mathcal{D}_i$, conditioning on $\mathbf{y}$ does not change the expectation of $c(\mathbf{x})$ by much. Formally, by Equation (5), for $i \in [m]$ and $b \in \{0, 1\}$,

$$\Pr_{\mathcal{D}_i}[\mathbf{y} = b]\left|\underset{\mathcal{D}_i|\mathbf{y}=b}{\mathbf{E}}[c(\mathbf{x})] - \underset{\mathcal{D}_i}{\mathbf{E}}[c(\mathbf{x})]\right| \leq \alpha. \tag{7}$$

Definition 8 requires a bound on the covariance for every distribution $\mathcal{D}_i$. This might be hard to achieve if $\mathcal{D}(S_i)$ is tiny, and hence we hardly see samples from $\mathcal{D}_i$ when sampling from $\mathcal{D}$. This motivated a relaxed definition called (α, β)-multicalibration in [16, 14]. We propose a different definition for which it is also easy to achieve sample efficiency. Rather than requiring the covariance be small for every i, we only require it to be small on average. Let $\mathbf{i} \sim \mathcal{D}$ denote sampling (the index of) a set from the partition $\mathcal{S}$ according to $\mathcal{D}$ so that $\Pr[\mathbf{i} = i] = \mathcal{D}(S_i)$.

▶ **Definition 9.** *The partition $\mathcal{S}$ of $\mathcal{X}$ is α-approximately multicalibrated for $\mathcal{C}, \mathcal{D}$ if for every $c \in \mathcal{C}$,*

$$\underset{\mathbf{i}\sim\mathcal{D}}{\mathbf{E}}\left|\underset{\mathcal{D}_\mathbf{i}}{\mathbf{Cov}}[c(\mathbf{x}), \mathbf{y}]\right| \leq \alpha. \tag{8}$$

The next lemma shows that approximate multicalibration implies closeness to (strict) multicalibration under the distribution $\mathcal{D}$. The proof is by applying Markov's inequality to Definition 9.

▶ **Lemma 10.** *If $\mathcal{S}$ is α-approximately multicalibrated for $\mathcal{C}, \mathcal{D}$, then for every $c \in \mathcal{C}$ and $\beta \in [0,1]$*

$$\Pr_{\mathbf{i}\sim\mathcal{D}}\left[\left|\mathop{\mathbf{Cov}}_{\mathcal{D}_\mathbf{i}}[c(\mathbf{x}), \mathbf{y}\right| \geq \frac{\alpha}{\beta}\right] \leq \beta. \tag{9}$$

This lemma shows that being $\alpha\beta$-approximately multicalibrated is closely related to the notion of (α, β)-multicalibration in [16, 14], which roughly says that the α-multicalibration condition holds for all but a β fraction of the space $\mathcal{X}$. Conversely, one can show that (α, β)-multicalibration gives $(\alpha + \beta \left\|\mathcal{C}\right\|_\infty)$-approximate multicalibration. We find the single parameter notion of α-approximate multicalibration more elegant. It is also easy to achieve sample efficiency, since as the next lemma shows, it only requires strong conditional guarantees for large states.

▶ **Lemma 11.** *Let the partition $\mathcal{S}$ be such that for all $i \in [m]$ where*

$$\mathcal{D}(S_i) \geq \frac{\alpha}{2m \left\|\mathcal{C}\right\|_\infty} \tag{10}$$

it holds that for every $c \in \mathcal{C}$,

$$\left|\mathop{\mathbf{Cov}}_{\mathcal{D}_i}[c(\mathbf{x}), \mathbf{y}]\right| \leq \frac{\alpha}{2}. \tag{11}$$

Then $\mathcal{S}$ is α-approximately multicalibrated for $\mathcal{C}, \mathcal{D}$.

Approximate multicalibration implies that for an *average* state in the partition, conditioning on the label does not change the expectation of $c(x)$ much. The proof follows by plugging Equation (5) in the definition of approximate multicalibration.

▶ **Corollary 12.** *If $\mathcal{S}$ is α-approximately multicalibrated for $\mathcal{C}, \mathcal{D}$, then for $c \in \mathcal{C}$ and $b \in \{0,1\}$,*

$$\mathop{\mathbf{E}}_{\mathbf{i}\sim\mathcal{D}}\left[\Pr_{\mathcal{D}_i}[\mathbf{y} = b]\left|\mathop{\mathbf{E}}_{\mathcal{D}_i|\mathbf{y}=b}[c(\mathbf{x})] - \mathop{\mathbf{E}}_{\mathcal{D}_i}[c(\mathbf{x})]\right|\right] \leq \alpha. \tag{12}$$

Extension to the multi-class setting

In the multi-class setting $\mathcal{Y} = [l]$, so that $l = 2$ is exactly the Boolean case considered above. We use $\mathbb{1}(\mathbf{y} = j)$ to denote the indicator of the event that the label is j. The following definition generalizes Definition 9:

▶ **Definition 13.** *Let $\mathcal{D}$ be a distribution on $\mathcal{X} \times [l]$ where $l \geq 2$. The partition $\mathcal{S}$ of $\mathcal{X}$ is α-approximately multicalibrated for $\mathcal{C}, \mathcal{D}$ if for every $c \in \mathcal{C}$ and $j \in [l]$, it holds that*

$$\mathop{\mathbf{E}}_{\mathbf{i}\sim\mathcal{D}}\left[\left|\mathop{\mathbf{Cov}}_{\mathcal{D}_i}[c(\mathbf{x}), \mathbb{1}(\mathbf{y} = j)]\right|\right] \leq \alpha. \tag{13}$$

Extension to the bounded real-valued case

We now consider the setting where $\mathcal{Y}$ is a bounded interval, by scaling we may consider $\mathcal{Y} = [0,1]$. For interval $J = [v, w] \subset \mathcal{Y}$, let $\mathbb{1}(\mathbf{y} \in J)$ be the indicator of the event that $\mathbf{y} \in J$.

▶ **Definition 14.** *Let $\mathcal{D}$ be a distribution on $\mathcal{X} \times [0,1]$. The partition $\mathcal{S}$ of $\mathcal{X}$ is α-approximately multicalibrated for $\mathcal{C}, \mathcal{D}$ if for every $c \in \mathcal{C}$ and interval $J \subseteq [0,1]$, it holds that*

$$\mathop{\mathbf{E}}_{\mathbf{i}\sim\mathcal{D}}\left[\left|\mathop{\mathbf{Cov}}_{\mathcal{D}_i}[c(\mathbf{x}), \mathbb{1}(\mathbf{y} \in J)]\right|\right] \leq \alpha. \tag{14}$$

Computational efficiency

Given these new definitions, a natural question is about the computational complexity of computing multicalibrated partitions. Following [16], this task can be accomplished efficiently given a weak agnostic learner for the class $\mathcal{C}$. We present a formal statement of this result for the multi-class setting in the full version of the paper. The multi-class setting includes the Boolean labels setting as a special case. For the purposes of omniprediction, we show in the full version of the paper that the bounded real-valued setting reduces to the multi-class setting. Due to space limitations, those results are to be found in the full version of this paper. However, several of the ideas used in the algorithm and its analysis are present in previous work, they are presented for completeness.

5.2 Some closure properties of multicalibration

In this section, we prove that approximate multicalibration is closed under two natural operations on the class $\mathcal{C}$ and the distribution $\mathcal{D}$:

1. **Linear combinations of $\mathcal{C}$:** We take sparse linear combinations of the functions $c \in \mathcal{C}$.
2. **Conditioning $\mathcal{D}$ on a subset:** We condition the distribution $\mathcal{D}$ on a subset $\mathcal{X}' \subseteq \mathcal{X}$ whose indicator lies in the set $\mathcal{C}$.

Again we prove these for the case $\mathcal{Y} = \{0, 1\}$, but the extension to arbitrary $\mathcal{Y}$ is routine.

Multi-calibration under linear combinations

We will typically start with an approximately multicalibrated partition for a *base* class of bounded or even Boolean functions, such as decision trees or coordinate functions. Since our definition allows the functions c to be real-valued and possibly unbounded, we can consider functions arising from linear combinations over this base class. The motivation for this comes from boosting algorithms like AdaBoost or Logistic Regression, where we take a base class of weak learners, and then construct a strong learner which is a linear combination of the weak learners [11, 34].

We will denote by $\mathrm{Lin}_{\mathcal{C}}$ the set of all linear functions over $\mathcal{C}$. We associate the vector $w = (w_0, w_1, \ldots)$ with the function $g_w \in \mathrm{Lin}_{\mathcal{C}}$ defined by $g_w(x) = w_0 + \sum_j w_j c_j(x)$ where $c_j \in \mathcal{C}$ and denote $\|w\|_1 = \sum_{j \geq 1} w_j$ (note we have excluded w_0). Let

$$\mathrm{Lin}_{\mathcal{C}}(W) = \{g_w \in \mathrm{Lin}_{\mathcal{C}} : \|w\|_1 \leq W\} \tag{15}$$

be the set of all W-*sparse* linear combinations. The following simple claim shows that multicalibration is *closed* under taking sparse linear combinations. The parameter α degrades with the sparsity. The proof follows from linearity of covariance, and is given in the full version of the paper.

▶ **Lemma 15.** *For any $W > 0$, if $\mathcal{S}$ is α-approximately multicalibrated for $\mathcal{C}, \mathcal{D}$, then it is αW-approximately multicalibrated for $\mathrm{Lin}_{\mathcal{C}}(W), \mathcal{D}$.*

Multi-calibration for sub-populations

Let $T \subseteq \mathcal{X}$ be a sub-population such that its indicator function belongs to $\mathcal{C}$. Let $\mathcal{D}'$ denote the distribution $\mathcal{D}|\mathbf{x} \in T$, where $\mathcal{D}'(x) = \mathcal{D}(x)/\mathcal{D}(T)\ \forall x \in T$. Let $\mathcal{S}' = \{S_i \cap T\}$ be the partition of T induced by $\mathcal{S}$. We will use $\mathcal{D}'_i$ for the distribution $\mathcal{D}'|\mathbf{x} \in S'_i$ (which is the same as $\mathcal{D}|\mathbf{x} \in S'_i$ since $S'_i \subseteq T$), and will denote $p'_i = \mathbf{E}_{\mathcal{D}'_i}[\mathbf{y}]$. Let $\mathcal{C}' \subseteq \mathcal{C}$ denote the subset of functions from $\mathcal{C}$ that are supported on T (functions that are 0 outside of T). Note that $\mathcal{C}'$ is nonempty, since the indicator of T lies in it. The proof of the following result for the sub-population T in the full version of the paper.

▶ **Theorem 16.** *If $\mathcal{S}$ is α-approximately multicalibrated for $\mathcal{C}, \mathcal{D}$, then $\mathcal{S}'$ is $\alpha(1 + \|\mathcal{C}'\|_\infty)/\mathcal{D}(\mathcal{X}')$-approximately multicalibrated for $\mathcal{C}', \mathcal{D}'$.*

6 Omnipredictors for convex loss minimization

In this section we consider the setting of binary labels where $\mathcal{Y} = \{0, 1\}$.

6.1 Post-processing for nice loss functions

Given an (B, ϵ)-nice loss function, there is a natural post-processing of the canonical predictor $f^{\mathcal{S}}$ that we will analyze. Rather than choose the value $k^*(p) \in \mathbb{R}$ which minimizes expected loss under Ber(p), we restrict ourselves to the best value from I_ℓ. This restriction only costs us ϵ by the ϵ-optimality property.

▶ **Definition 17.** *Given a nice loss function ℓ, define the function $k_\ell : [0, 1] \to I_\ell$ by*

$$k_\ell(p) = \arg\min_{t \in I_\ell} \mathop{\mathbf{E}}_{\mathbf{y} \sim \mathrm{Ber(p)}} \ell(\mathbf{y}, t). \tag{16}$$

Given a partition $\mathcal{S}$ of $\mathcal{X}$, define the ℓ-optimized hypothesis $h_\ell^{\mathcal{S}} : \mathcal{X} \to I_\ell$ as

$$h_\ell^{\mathcal{S}}(x) = k_\ell \circ f^{\mathcal{S}}(x).$$

Since ℓ is convex as a function of t, so is

$$\mathop{\mathbf{E}}_{\mathbf{y} \sim \mathrm{Ber(p)}} \ell(\mathbf{y}, t) = p\ell(0, t) + (1 - p)\ell(1, t).$$

Hence computing k_ℓ is a one-dimensional convex minimization problem, a classical problem with several known algorithms [6]. Being able to compute an ϵ'-approximate solution suffices for us, we can absorb the ϵ' term into the error ϵ, and pretend that ℓ is $(B, \epsilon + \epsilon')$-nice instead.

We can view the hypothesis $h_\ell^{\mathcal{S}}$ as a function mapping $\mathcal{S}$ to I_ℓ, since it is constant on each $S_i \in \mathcal{S}$, and its range is I_ℓ. A simple consequence of the definition is that it is the best function in this class for minimizing expected loss.

▶ **Corollary 18.** *For all functions $h : \mathcal{S} \to I_\ell$, $\ell_{\mathcal{D}}(h_\ell^{\mathcal{S}}) \le \ell_{\mathcal{D}}(h)$.*

Proof. We sample $\mathbf{i} \sim \mathcal{D}$ and then $\mathbf{x}, \mathbf{y} \sim \mathcal{D}_{\mathbf{i}}$ and show that the inequality holds conditioned on every choice of $\mathbf{i} = i$. Since $\mathbf{y} \sim D_i$ is distributed as $\mathrm{Ber}(\mathrm{p_i})$,

$$\ell_{\mathcal{D}_i}(h_\ell^{\mathcal{S}}) = \mathop{\mathbf{E}}_{\mathrm{Ber(p_i)}}[\ell(\mathbf{y}, k_\ell(p_i))] \le \mathop{\mathbf{E}}_{\mathrm{Ber(p_i)}}[\ell(\mathbf{y}, h(S_i))] = \ell_{\mathcal{D}_i}(h)$$

where the inequality is by Definition 17. ◀

6.2 Loss minimization through Multicalibration

Our main result in this section is the following theorem.

▶ **Theorem 19.** *Let $\mathcal{D}$ be a distribution on $\mathcal{X} \times \{0, 1\}$, $\mathcal{C}$ be a family of real-valued functions on $\mathcal{X}$ and $\mathcal{L}(B, \epsilon)$ be the family of all (B, ϵ)-nice loss functions. If the partition $\mathcal{S}$ is α-approximately multicalibrated for $\mathcal{C}, \mathcal{D}$, then the canonical predictor $f^{\mathcal{S}}$ is an $(\mathcal{L}, \mathcal{C}, 2\alpha B + \epsilon)$-omnipredictor.*

The following lemma is the key ingredient in our result. Informally it says that c has limited distinguishing power within each state of the partition. Specifically, that $c(x)$ is not much better at minimizing a loss than the function obtained by taking its conditional expectation within each state of the partition $\mathcal{S}$.

▶ **Lemma 20.** *Given $\ell \in \mathcal{L}$ and $c \in \mathcal{C}$, define the predictor $\widehat{c} : \mathcal{S} \to I_\ell$ by*

$$\widehat{c}(x) = \mathrm{clip}\left(\mathop{\mathbf{E}}_{\mathcal{D}_i}[c(x)], I_\ell\right) \text{ for } x \in S_i.$$

We have

$$\ell_{\mathcal{D}}(\widehat{c}) \leq \ell_{\mathcal{D}}(c) + 2\alpha B + \epsilon. \tag{17}$$

Proof. By the convexity of ℓ we have

$$\ell_{\mathcal{D}}(c) = \mathop{\mathbf{E}}_{\mathcal{D}}[\ell(\mathbf{y}, c(\mathbf{x}))] = \mathop{\mathbf{E}}_{\mathbf{i}\sim\mathcal{D}} \mathop{\mathbf{E}}_{\mathbf{y}\sim\mathcal{D}_\mathbf{i}} \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_\mathbf{i}|\mathbf{y}}[\ell(\mathbf{y}, c(\mathbf{x})] \geq \mathop{\mathbf{E}}_{\mathbf{i}\sim\mathcal{D}} \mathop{\mathbf{E}}_{\mathbf{y}\sim\mathcal{D}_\mathbf{i}} \left[\ell\left(\mathbf{y}, \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_\mathbf{i}|\mathbf{y}}[c(\mathbf{x})]\right)\right]. \tag{18}$$

By Lemma 2,

$$\ell\left(\mathbf{y}, \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_i|\mathbf{y}}[c(\mathbf{x})]\right) \geq \ell\left(\mathbf{y}, \mathrm{clip}\left(\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_i|\mathbf{y}}[c(\mathbf{x})], I_\ell\right)\right) - \epsilon.$$

Plugging this into Equation (18), we get

$$\ell_{\mathcal{D}}(c) \geq \mathop{\mathbf{E}}_{\mathbf{i}\sim\mathcal{D}} \mathop{\mathbf{E}}_{\mathbf{y}\sim\mathcal{D}_\mathbf{i}} \left[\ell\left(\mathbf{y}, \mathrm{clip}\left(\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_i|\mathbf{y}}[c(\mathbf{x})], I_\ell\right)\right)\right] - \epsilon. \tag{19}$$

From the definition of $\widehat{c}$,

$$\ell_{\mathcal{D}}(\widehat{c}) = \mathop{\mathbf{E}}_{\mathcal{D}}[\ell(\mathbf{y}, \widehat{c}(\mathbf{x}))] = \mathop{\mathbf{E}}_{\mathbf{i}\sim\mathcal{D}} \mathop{\mathbf{E}}_{\mathbf{y}\sim\mathcal{D}_\mathbf{i}} \left[\ell\left(\mathbf{y}, \mathrm{clip}\left(\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_i}[c(\mathbf{x})], I_\ell\right)\right)\right]. \tag{20}$$

Subtracting Equation (19) from Equation (20) we get

$$\ell_{\mathcal{D}}(\widehat{c}) - \ell_{\mathcal{D}}(c) \leq \mathop{\mathbf{E}}_{\mathbf{i}\sim\mathcal{D}} \mathop{\mathbf{E}}_{\mathbf{y}\sim\mathcal{D}_\mathbf{i}} \left[\ell\left(\mathbf{y}, \mathrm{clip}\left(\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_i}[c(\mathbf{x})], I_\ell\right)\right) - \ell\left(\mathbf{y}, \mathrm{clip}\left(\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_i|\mathbf{y}}[c(\mathbf{x})], I_\ell\right)\right)\right] + \epsilon. \tag{21}$$

Since ℓ is B-Lipschitz on I_ℓ and $\mathrm{clip}(t, I_\ell)$ is 1-Lipschitz as a function of t,

$$\begin{aligned}
&\ell\left(\mathbf{y}, \mathrm{clip}\left(\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_i}[c(\mathbf{x})], I_\ell\right)\right) - \ell\left(\mathbf{y}, \mathrm{clip}\left(\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_i|\mathbf{y}}[c(\mathbf{x})], I_\ell\right)\right) \\
&\qquad \leq B\left|\mathrm{clip}\left(\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_i}[c(\mathbf{x})], I_\ell\right) - \mathrm{clip}\left(\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_i|\mathbf{y}}[c(\mathbf{x})], I_\ell\right)\right| \\
&\qquad \leq B\left|\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_i}[c(\mathbf{x})] - \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_i|\mathbf{y}}[c(\mathbf{x})]\right|.
\end{aligned} \tag{22}$$

Plugging this into Equation (21) gives

$$\begin{aligned}
\ell_{\mathcal{D}}(\widehat{c}) - \ell_{\mathcal{D}}(c) - \epsilon &\leq B \mathop{\mathbf{E}}_{\mathbf{i}\sim\mathcal{D}} \mathop{\mathbf{E}}_{\mathbf{y}\sim\mathcal{D}_\mathbf{i}} \left|\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_i}[c(\mathbf{x})] - \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_i|\mathbf{y}}[c(\mathbf{x})]\right| \\
&= B \mathop{\mathbf{E}}_{\mathbf{i}\sim\mathcal{D}} \left[\sum_{b\in\{0,1\}} \Pr_{\mathcal{D}_\mathbf{i}}[\mathbf{y} = b] \left|\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_i}[c(\mathbf{x})] - \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_i|\mathbf{y}}[c(\mathbf{x})]\right|\right] \\
&= B \sum_{b\in\{0,1\}} \mathop{\mathbf{E}}_{\mathbf{i}\sim\mathcal{D}} \left[\Pr_{\mathcal{D}_\mathbf{i}}[\mathbf{y} = b] \left|\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_i}[c(\mathbf{x})] - \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{D}_i|\mathbf{y}}[c(\mathbf{x})]\right|\right] \\
&\leq B \sum_{b\in\{0,1\}} \alpha = 2\alpha B,
\end{aligned} \tag{23}$$

where the last inequality follows by the multicalibration condition (Equation (7)). ◀

As a consequence we can now prove Theorem 19.

Proof of Theorem 19. Let $h_\ell^{\mathcal{S}} = k_\ell \circ f^{\mathcal{S}}$ be the ℓ-optimized hypothesis. It suffices to show that for any $c \in \mathcal{C}$

$$\ell_{\mathcal{D}}(h_\ell^{\mathcal{S}}) \leq \ell_{\mathcal{D}}(c) + 2\alpha B + \epsilon. \tag{24}$$

For any $c \in \mathcal{C}$, we have

$$\ell_{\mathcal{D}}(h_\ell^{\mathcal{S}}) \leq \ell_{\mathcal{D}}(\widehat{c}) \leq \ell_{\mathcal{D}}(c) + 2\alpha B + \epsilon$$

where the first inequality is by Corollary 18, which applies since $\widehat{c}$ is a function mapping $\mathcal{S}$ to I_ℓ. The second is by Lemma 20. ◀

Consider the family $\mathrm{Lin}_{\mathcal{C}}(W)$ of linear combinations over $\mathcal{C}$ of weight at most W. By Lemma 15, $\mathcal{S}$ is αW-approximately multicalibrated for $\mathrm{Lin}_{\mathcal{C}}(W)$. Applying Theorem 19, we derive the following corollary.

▶ **Corollary 21.** *Let $\mathcal{D}$ be a distribution on $\mathcal{X} \times \{0,1\}$, $\mathrm{Lin}_{\mathcal{C}}(W)$ be linear functions in $\mathcal{C}$ of with $\|w\|_1 \leq W$ (see eq. 15) and $\mathcal{L} = \mathcal{L}(B,\epsilon)$ be the family of all (B,ϵ)-nice loss functions. If the partition $\mathcal{S}$ is α-approximately multicalibrated for $\mathcal{C}, \mathcal{D}$, then $f^{\mathcal{S}}$ is an $(\mathcal{L}, \mathrm{Lin}_{\mathcal{C}}(W), 2\alpha BW + \epsilon)$-omnipredictor.*

To interpret this, assume we have an (B,ϵ)-nice loss function and we wish to have a predictor that is within 2ϵ of any function in $\mathrm{Lin}_{\mathcal{C}}(W)$. Corollary 21 says that it suffices to have an α-approximately multicalibrated partition where $\alpha = \epsilon/2BW$. Note that algorithms for computing such partitions have running time which is polynomial in $1/\alpha$, which translates to running time polynomial in BW/ϵ.

We derive a corollary for sub-populations follows from Theorem 19 and 16. For two families of functions $\mathcal{T}, \mathcal{P} : \mathcal{X} \to \mathbb{R}$, we define their product as

$$\mathcal{T} \times \mathcal{P} = \{c : c(x) = T(x)P(x), T \in \mathcal{T}, P \in \mathcal{P}\}.$$

Note that in the case when $T \in \mathcal{T}$ is binary-valued, $\mathcal{T} \times \mathcal{P}$ contains the restriction of every $P \in \mathcal{P}$ to the support of T.

▶ **Corollary 22.** *Let $\mathcal{D}$ be a distribution on $\mathcal{X} \times \{0,1\}$, $\mathcal{T}$ be a family of binary-valued functions on $\mathcal{X}$, $\mathcal{P}$ be a family of real-valued functions and $\mathcal{L}(B,\epsilon)$ be the family of all (B,ϵ)-nice loss functions. Let the partition $\mathcal{S}$ be α-approximately multicalibrated for $\mathcal{T} \times \mathcal{P}, \mathcal{D}$. For $T \in \mathcal{T}$, let $\alpha' = \alpha(1 + \|\mathcal{P}\|_\infty)/\mathcal{D}(T)$. Then the canonical predictor $f^{\mathcal{S}}$ is an $(\mathcal{L}, \mathcal{P}, 2\alpha' B + \epsilon)$-omnipredictor for the sub-population T.*

To informally instantiate this for a simple case, let $\mathcal{T}, \mathcal{P}$ be the class of decision trees of depth d_1 and d_2 respectively. Let the partition $\mathcal{S}$ be multicalibrated with respect to decision trees of depth $d_1 + d_2$. If we now consider any sub-population T identified by decision trees of depth d_1, then the above result implies that the canonical predictor $f^{\mathcal{S}}$ is an omnipredictor for T, when compared against the class of decision trees of depth d_2 evaluated on T.

6.3 Limits for omnipredictors from multicalibration

Corollary 21 shows that multicalibration for $\mathcal{C}$ gives omnipredictors for $\mathrm{Lin}_{\mathcal{C}}$. It is natural to ask whether we can get omnipredictors for a richer class of functions using multicalibration for $\mathcal{C}$. A natural candidate would be thresholds of functions in $\mathcal{C}$:

$$\mathrm{Thr}_{\mathcal{C}} = \{\mathbb{1}(c(x) \geq v) : c \in \mathcal{C}, v \in \mathbb{R}\}.$$

Another natural extension would be to relax the convexity condition for loss functions in $\mathcal{L}$. We present a simple counterexample which shows that multicalibration for $\mathcal{C}$ is insufficient to give omnipredictors for both these classes. This shows that a significant strengthening of the bound from Corollary 21 might not be possible.

▶ **Lemma 23.** *There exists a distribution $\mathcal{D}$, a set $\mathcal{C} : \mathcal{X} \to \mathbb{R}$ of functions, and a 0-multicalibrated partition $\mathcal{S}$ for $\mathcal{C}, \mathcal{D}$ such that for any $\delta < 1/4$,*

- *$f^{\mathcal{S}}$ is* not *an $(\mathcal{L}, \mathrm{Thr}_{\mathcal{C}}, \delta)$-omnipredictor for any $\mathcal{L}$ containing the ℓ_1 loss function.*
- *$f^{\mathcal{S}}$ is* not *an $(\mathcal{L}, \mathcal{C}, \delta)$-omnipredictor for any $\mathcal{L}$ containing the (non-convex) loss function $\ell(y, t) = |y - \mathbb{1}(t \geq 0)|$.*

Proof. Let $\mathcal{D}$ be the distribution on $\{0,1\}^3 \times \{0,1\}$ where $\mathbf{x} \sim \{0,1\}^3$ is sampled uniformly and $\mathbf{y} = \mathbb{1}(\sum_{i=1}^{3} \mathbf{x}_i \equiv 0 \bmod 2)$ is the negated Parity function. Let $\mathcal{C} = \{\sum_i w_i x_i - w_0\}$ be all affine functions. We claim the trivial partition $\mathcal{S} = \{\{0,1\}^3\}$ is 0-multicalibrated for $\mathcal{C}, \mathcal{D}$. This is because every x_i is independent of $\mathbf{y}$, so their covariance is 0. By linearity of expectation, the same is true for all functions in $\mathcal{C}$. Thus $f^{\mathcal{S}}(x) = 1/2$ for every $x \in \{0,1\}^3$.

Now consider the ℓ_1 loss. A simple calculation shows that for every $k : \{0,1\} \to \mathbb{R}$, $\bar{\ell}_1(k \circ f^{\mathcal{S}}, \mathcal{D}) \geq 1/2$. In contrast $h(x) = \mathbb{1}(x_1 + x_2 + x_3 \geq 1.5) \in \mathrm{Thr}_{\mathcal{C}}$ gives $\bar{\ell}_1(h, \mathcal{D}) = 1/4$, since it gets the two middle layers correct. This proves part (1).

To deduce part (2), let $\ell(y, t) = |y - \mathbb{1}(t \geq 0)|$ and $g(x) = x_1 + x_2 + x_3 - 1.5 \in \mathcal{C}$. Note that

$$1/4 = \bar{\ell}_1(h, \mathcal{D}) = \mathop{\mathbf{E}}_{\mathcal{D}}[|h(\mathbf{x}) - \mathbf{y}|] = \mathop{\mathbf{E}}_{\mathcal{D}}[|\mathbb{1}(g(\mathbf{x}) \geq 0) - \mathbf{y}|] = \mathop{\mathbf{E}}_{\mathcal{D}}[\ell(\mathbf{y}, g(\mathbf{x}))] = \bar{\ell}(g, \mathcal{D}).$$

In contrast, for any $k : [0,1] \to \mathbb{R}$, it follows that $\bar{\ell}(k \circ f^{\mathcal{S}}, \mathcal{D}) \geq 1/2$. ◀

In part (2), the loss function $|y - \mathbb{1}(t \geq 0)|$ is not Lipshcitz or differentiable in t. We can ensure both these conditions by replacing it with the sigmoid function, which still preserves the correlation with parity, at the cost of some reduction in δ for which the bound holds [18].

References

1 N Barda, G Yona, GN Rothblum, P Greenland, M Leibowitz, R Balicer, E Bachmat, and N. Dagan. Addressing bias in prediction models by improving subpopulation calibration. *J Am Med Inform Assoc.*, 28(3):549–558, 2021.

2 Noam Barda, Dan Riesel, Amichay Akriv, Joseph Levy, Uriah Finkel, Gal Yona, Daniel Greenfeld, Shimon Sheiba, Jonathan Somer, Eitan Bachmat, Guy N. Rothblum, Uri Shalit, Doron Netzer, Ran Balicer, and Noa Dagan. Developing a COVID-19 mortality risk prediction model when individual-level data are not available. *Nat Commun*, 11, 2020.

3 Peter L Bartlett, Michael I Jordan, and Jon D McAuliffe. Convexity, classification, and risk bounds. *Journal of the American Statistical Association*, 101(473):138–156, 2006.

4 Shai Ben-David, Philip M. Long, and Yishay Mansour. Agnostic boosting. In *14th Annual Conference on Computational Learning Theory, COLT*, 2001.

5 Avrim Blum and Thodoris Lykouris. Advancing Subgroup Fairness via Sleeping Experts. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151, pages 55:1–55:24, 2020.

6 Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

7 A. P. Dawid. Objective probability forecasts. *University College London, Dept. of Statistical Science. Research Report 14*, 1982.

8 Krzysztof Dembczyński, Wojciech Kotłowski, Oluwasanmi Koyejo, and Nagarajan Natarajan. Consistency analysis for binary classification revisited. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 961–969. PMLR, 06–11 August 2017. URL: https://proceedings.mlr.press/v70/dembczynski17a.html.

9 Cynthia Dwork, Michael P. Kim, Omer Reingold, Guy N. Rothblum, and Gal Yona. Outcome indistinguishability. In *ACM Symposium on Theory of Computing (STOC'21)*, 2021. URL: https://arxiv.org/abs/2011.13426.

10 Vitaly Feldman. Distribution-specific agnostic boosting. *arXiv preprint arXiv:0909.2927*, 2009.

11 Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

12 Sumegha Garg, Michael P. Kim, and Omer Reingold. Tracking and improving information in the service of fairness. In *Proceedings of the 2019 ACM Conference on Economics and Computation, EC 2019, Phoenix, AZ, USA, June 24-28, 2019*, pages 809–824, 2019.

13 Parikshit Gopalan, Adam Tauman Kalai, Omer Reingold, Vatsal Sharan, and Udi Wieder. Omnipredictors, 2021. arXiv:2109.05389.

14 Parikshit Gopalan, Omer Reingold, Vatsal Sharan, and Udi Wieder. Multicalibrated partitions for importance weights. *arXiv preprint arXiv:2103.05853*, 2021.

15 Bruce E Hansen. Autoregressive conditional density estimation. *International Economic Review*, pages 705–730, 1994.

16 Úrsula Hébert-Johnson, Michael P. Kim, Omer Reingold, and Guy N. Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *Proceedings of the 35th International Conference on Machine Learning, ICML*, 2018.

17 Christopher Jung, Changhwa Lee, Mallesh M. Pai, Aaron Roth, and Rakesh Vohra. Moment multicalibration for uncertainty estimation. *CoRR*, abs/2008.08037, 2020. arXiv:2008.08037.

18 Adam Kalai. Learning monotonic linear functions. In John Shawe-Taylor and Yoram Singer, editors, *Learning Theory, 17th Annual Conference on Learning Theory, COLT 2004*, volume 3120 of *Lecture Notes in Computer Science*, pages 487–501. Springer, 2004. doi:10.1007/978-3-540-27819-1_34.

19 Adam Kalai and Varun Kanade. Potential-based agnostic boosting. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009. URL: https://proceedings.neurips.cc/paper/2009/file/13f9896df61279c928f19721878fac41-Paper.pdf.

20 Adam Tauman Kalai. Learning nested halfspaces and uphill decision trees. In *Learning Theory, 20th Annual Conference on Learning Theory, COLT 2007*, volume 4539 of *Lecture Notes in Computer Science*, pages 378–392. Springer, 2007. doi:10.1007/978-3-540-72927-3_28.

21 Adam Tauman Kalai, Adam R. Klivans, Yishay Mansour, and Rocco A. Servedio. Agnostically learning halfspaces. *SIAM J. Comput.*, 37(6):1777–1805, 2008. doi:10.1137/060649057.

22 Adam Tauman Kalai, Yishay Mansour, and Elad Verbin. On agnostic boosting and parity learning. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 629–638. ACM, 2008. doi:10.1145/1374376.1374466.

23 Adam Tauman Kalai and Rocco A Servedio. Boosting in the presence of noise. *Journal of Computer and System Sciences*, 71(3):266–290, 2005.

24 Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International Conference on Machine Learning*, pages 2564–2572, 2018.

25 Michael J. Kearns and Yishay Mansour. On the boosting ability of top-down decision tree learning algorithms. *J. Comput. Syst. Sci.*, 58(1):109–128, 1999. doi:10.1006/jcss.1997.1543.

26 Michael P. Kim. A complexity-theoretic perspective on fairness. *PhD thesis, Stanford University*, 2020.

27 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.

28 Jon M. Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores. In *8th Innovations in Theoretical Computer Science Conference, ITCS*, 2017.

29 Gábor Lugosi, Nicolas Vayatis, et al. On the bayes-risk consistency of regularized boosting methods. *The Annals of statistics*, 32(1):30–55, 2004.

30 Yishay Mansour and David McAllester. Boosting using branching programs. *Journal of Computer and System Sciences*, 64(1):103–112, 2002.

31 Nagarajan Natarajan, Oluwasanmi Koyejo, Pradeep Ravikumar, and Inderjit S. Dhillon. Optimal decision-theoretic classification using non-decomposable performance metrics. *CoRR*, abs/1505.01802, 2015. arXiv:1505.01802.

32 John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999.

33 Guy N. Rothblum and Gal Yona. Multi-group agnostic PAC learnability. *CoRR*, abs/2105.09989, 2021. arXiv:2105.09989.

34 Robert E. Schapire and Yoav Freund. *Boosting: Foundations and Algorithms*. MIT Press, 2012.

35 Mark J. Schervish. A general method for comparing probability assessors. *The Annals of Statistics*, 17(4):1856–1879, 1989.

36 Clayton Scott. Calibrated asymmetric surrogate losses. *Electronic Journal of Statistics*, 6:958–992, 2012.

37 Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, 2014.

38 Ingo Steinwart. Consistency of support vector machines and other regularized kernel classifiers. *IEEE transactions on information theory*, 51(1):128–142, 2005.

39 Ambuj Tewari and Peter L Bartlett. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8(5), 2007.

40 Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001)*, pages 609–616. Morgan Kaufmann, 2001.