Machine Learning: Social Values, Data Efficiency, and Beyond Prediction

Travis Dick

November 22, 2018

Computer Science Department
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
tdick@cs.cmu.edu

Thesis Committee:
Maria-Florina Balcan (chair)
Yishay Mansour (Tel Aviv University)
Tom Mitchell (Carnegie Mellon University)
Ariel Procaccia (Carnegie Mellon University)

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

Abstract

In this thesis, we build on the theory and practice of Machine Learning to accommodate several modern requirements of learning systems. In particular, we focus on new requirements stemming from three distinct sources: making the best use of available data, applying learning tools to problems beyond standard prediction, and incorporating social values. Each of these themes provides an exciting research direction for the practice and theory of Machine Learning.

Learning from Mostly Unlabeled Data in Multi-class Settings. Large scale multi-class learning tasks with an abundance of unlabeled data are ubiquitous in modern machine learning. For example, an in-home assistive robot needs to learn to recognize common household objects, familiar faces, facial expressions, gestures, and so on in order to be useful. Such a robot can acquire large amounts of unlabeled training data simply by observing its surroundings, but it would be prohibitively time consuming to ask its owner to annotate any large portion of this data. More generally, in many modern learning problems we often have easy and cheap access to large quantities of unlabeled training data but obtaining high-quality labeled examples is relatively expensive. The first chapter of my thesis will focus on theory for large-scale multi-class learning with limited labeled data. We begin by assuming that a given supervised learning algorithm would succeed at the learning task if it had access to labeled data. Then we use the implicit assumptions made by that algorithm to show that different label-efficient algorithms will also succeed.

Machine Learning Beyond Standard Prediction Problems. While most machine learning focuses on learning to make predictions, there are important learning problems where the output of the learner is not a prediction rule. We focus on data-driven algorithm configuration, where the goal is to find the best algorithm parameters for a specific application domain. A recent exciting line of work has considered data-driven algorithm configuration in the statistical setting, where each application domain is modeled as a distribution over problem instances. In this work, we extend the theoretical foundations of this field to accommodate two new learning settings: the online setting where problems are chosen by an adversary and arrive one at a time, and the private setting, where each problem instance contains sensitive information that should not be released. Algorithm configuration problems often reduce to the maximization of a collection of piecewise Lipschitz functions. Unfortunately, the discontinuities of these functions lead to worst-case impossibility results for both the online and private settings. Our main contribution is a novel condition called dispersion that, when satisfied, allows for meaningful regret bounds and utility guarantees in these settings. We also show that dispersion is satisfied for many problems under mild assumptions, making dispersion useful both in theory and practice.

Social Values for Machine Learning Systems. Machine learning systems are becoming central to the infrastructure of our society. The wide-spread use of such systems creates exciting possibilities for profoundly positive impacts in our lives though improvements to, for example, medicine, communication, and transportation. Since these systems are often not explicitly designed with social values in mind, there is a risk that their adoption could result in undesired outcomes such as privacy violations of unfair treatment of individuals. The final chapter of my thesis will focus on developing principled techniques for incorporating two social values into machine learning algorithms: privacy and fairness. We import the fairness notion of envy-freeness from fair division into machine learning and consider whether it is possible to learn envy-free classifiers. We also develop general tools for differentially private optimization of piecewise Lipschitz functions, a problem that arises naturally in several learning settings, including applications beyond standard prediction problems.

Contents

Chapter 1

Introduction

In this thesis we extend the practice and theory of machine learning to accommodate several new requirements. We focus on three themes, each stemming from the application of machine learning to modern real-world scenarios. This proposal describes three ongoing projects with exciting future research directions that each connect with one or more of these themes. First we look at making the best use of the types of data that are readily available. Next, we apply machine learning tools to problems beyond standard prediction. Finally, we consider incorporating social values, like privacy and fairness, into machine learning systems.

Data Efficiency in Multi-class Learning. In many real-world classification problems, unlabeled data is cheap and readily available, while obtaining high-quality labels is time consuming or expensive. Machine learning systems should make the best use of the cheap and abundantly available unlabeled data to minimize the need for the more expensive labeled data. This has been the focus of the fields of semi-supervised and active learning. In Chapter 2, we explore conditions for large-scale multi-class learning under which unlabeled data provably reduces the labeled sample complexity of learning. Our results argue that if the underlying learning task is one for which a given supervised learning algorithm would succeed at learning, given a fully labeled dataset, then much more label efficient algorithms will also succeed. We obtain these results by carefully examining the implicit assumptions made by specific classes of supervised learning algorithms, showing that when they succeed it implies geometric structure in the underlying learning problem that can be exploited in semi-supervised and active learning settings.

Machine Learning Beyond Prediction. Most machine learning systems learn to make predictions from data. For example, in medical settings, the learner may receive a training dataset consisting of medical feature representations describing patients, together with an expert diagnosis for each patient. Then the learner's goal is to identify patterns in the expert diagnoses so that it can predict what the diagnosis will be for new unseen patients. A key requirement is that the learned prediction rules generalize, in the sense that they make accurate predictions on new unseen data.

There are many other important learning problems where the output of the learner is not a prediction rule. An alternate setting that we consider in Chapter 3 is data-driven algorithm configuration. Our goal is to find the best parameter settings for a parameterized algorithm when it is used for problem instances arising in some specific target application. The parameter-tuning procedure learns about the specific application of the algorithm through a collection of example problem instances. For

example, if our goal is to learn good parameters for a clustering algorithm, then the training data could consist of clustering instances together with target clusterings for each instance. Then the learner's goal is to find parameters that produce clusterings with the best possible agreement on the training instances. Of course, generalization is also crucial in this setting: we care about finding parameters that will have good agreement with the (unknown) target clusterings on future clustering instances.

While there are many similarities between learning to predict and data-driven algorithm configuration, there are also a number of important differences that need to be overcome. For example, the output of an algorithm is often a volatile function of its parameters, and very small changes to the parameters can lead to a cascade of differences in the output. Chapter 3 explores online and private optimization of piecewise Lipschitz functions, an optimization problem that captures many of the challenges of online data-driven algorithm configuration.

Social Values in Machine Learning. When machine learning systems interact with society, we expect them to uphold our social values. For example, if a model is trained on data containing sensitive information about individuals, it should not be possible to learn about specific individuals in the training data by inspecting either its predictions or its parameters. Similarly, when machine learned models are used to make consequential decisions, such as whether to release a defendant on bail, or whether to grant an individual a loan, we want guarantees that the learned models uphold our notions of fair treatment. Since models trained to maximize predictive accuracy may not inherently have these properties, an important research direction is to design principled techniques for explicitly incorporating social values into learning algorithms.

In Chapter 4, we import *envy-freeness* from fair-division as a notion of fairness in machine learning. Envy-freeness is particularly well-suited to situations where we are learning to assign one of many outcomes to individuals who have heterogeneous preferences for those outcomes. This is in contrast with much of the fairness literature, which focuses on binary classification settings with one desirable outcome and one undesirable. Intuitively, a classifier that assigns individuals to outcomes is envy-free if no individual would prefer to change their assignment to that of someone else. Our current results focus on the generalizability of envy-freeness, showing that as long as the learner restricts itself to a set of relatively low-complexity classifiers, classifiers that are envy-free on a large enough sample will remain approximately envy-free on the underlying distribution.

The project on piecewise Lipschitz optimization discussed in Chapter 3 also connects with the theme of incorporating social values into machine learning systems. Our results show how to optimize a collection of piecewise Lipschitz functions while at the same time providing very strong privacy guarantees when those functions encode sensitive information about individuals.

Chapter 2

Multiclass Classification with Less Labeled Data

2.1 Introduction

Large scale multi-class learning problems with an abundance of unlabeled data are ubiquitous in modern machine learning. For example, an in-home assistive robot needs to learn to recognize common household objects, familiar faces, facial expressions, gestures, and so on in order to be useful. Such a robot can acquire large amounts of unlabeled training data simply by observing its surroundings, but it would be prohibitively time consuming (and frustrating) to ask its owner to annotate any significant portion of this raw data. More generally, in many modern learning problems we often have easy and cheap access to large quantities of unlabeled training data (e.g., on the internet) but obtaining high-quality labeled examples is relatively expensive. More examples include text understanding, recommendation systems, or wearable computing [70, 72, 71, 57]. The scarcity of labeled data is especially pronounced in problems with many classes, since supervised learning algorithms require labeled examples from every class. In such settings, algorithms should make the best use of unlabeled data in order to minimize the need for expensive labeled examples.

We approach label-efficient learning by making the implicit assumptions of popular multi-class learning algorithms explicit and showing that they can also be exploited when learning from limited labeled data. We focus on a family of techniques called *output codes* that work by decomposing a given multi-class problem into a collection of binary classification tasks [58, 32, 54, 17]. The novelty of our results is to show that the existence of various low-error output codes constrains the distribution of unlabeled data in ways that can be exploited to reduce the label complexity of learning. We consider both the consistent setting, where the output code achieves zero error, and the agnostic setting, where the goal is to compete with the best output code. The most well known output code technique is one-vs-all learning, where we learn one binary classifier for distinguishing each class from the union of the rest. When output codes are successful at learning from labeled data, it often implies geometric structure in the underlying problem. For example, if it is possible to learn an accurate one-vs-all classifier with linear separators, it implies that no three classes can be collinear, since then it would be impossible for a single linear separator to distinguish the middle class from the union of the others. In this work, we exploit this implicitly assumed structure to design label-efficient algorithms for the commonly assumed cases of one-vs-all and error correcting output codes, as well as a novel boundary features condition that captures the intuition that every bit of the codewords should be significant.

This project is joint work with Maria-Florina Balcan and Yishay Mansour. Our current results appeared in AAAI 2017 [13].

2.2 Preliminaries and Results

We consider multiclass learning problems over an instance space $\mathcal{X} \subset \mathbb{R}^d$ where each point is labeled by $f^* : \mathcal{X} \to \{1, \ldots, k\}$ to one out of k classes and the probability of observing each outcome $x \in \mathcal{X}$ is determined by a data distribution $\mathcal{P}$ on . The density function of $\mathcal{P}$ is denoted by $p : \mathcal{X} \to [0, \infty)$. In all of our results we assume that there exists a consistent (but unknown) linear output-code classifier defined by a code matrix $C \in \{\pm 1\}^{L \times m}$ and m linear separators $h_1, \ldots, h_m$. We denote class i's code word by C_i and define $h(x) = (\text{sign}(h_1(x)), \ldots, \text{sign}(h_m(x)))$ to be the code word for point x. We let (c, c') denote the Hamming distance between any codewords $c, c' \in \{\pm 1\}^m$. To simplify notation, we assume that has diameter ≤ 1.

Our goal is to learn a hypothesis $\hat{f} :\to \{1, \ldots, k\}$ minimizing $\text{err}(\hat{f}) = \Pr_{X \sim \mathcal{P}}(\hat{f}(x) \neq f^*(x))$ from an unlabeled sample drawn from the data distribution $\mathcal{P}$ together with a small set of actively queried labeled examples.

Finally, we use the following notation throughout the paper: for any set A in a metric space $(, d)$, the σ-interior of A is the set $\text{int}_\sigma(A) = \{x \in A : B(x, \sigma) \subset A\}$. The notation $\tilde{O}(\cdot)$ suppresses logarithmic terms.

Before discussing our results, we briefly review the output code methodology. For a problem with L classes, a domain expert designs a code matrix $C \in \{\pm 1\}^{L \times m}$ where each column partitions the classes into two meaningful groups. The number of columns m is chosen by the domain expert. For example, when recognizing household objects we could use the following true/false questions to define the partitions: "is it made of wood?", "is it sharp?", "does it have legs?", "should I sit on it?", and so on. Each row of the code matrix describes one of the classes in terms of these partitions (or semantic features). For example, the class "table" could be described by the vector $(+1, -1, +1, -1)$, which is called the class' codeword. Once the code matrix has been designed, we train an output code by learning a binary classifier for each of the binary partitions (e.g., predicting whether an object is made of wood or not). To predict the class of a new example, we predict its codeword in $\{\pm 1\}^m$ and output the class with the nearest codeword under the Hamming distance. Two popular special cases of output codes are one-vs-all learning, where C is the identity matrix (with -1 in the off-diagonal entries), and error correcting output codes, where the Hamming distance between the codewords is large.

In each of our results we assume that there exists a consistent or low-error linear output code classifier and we impose constraints on the code matrix and the distribution that generates the data. We present algorithms and analysis techniques for a wide range of different conditions on the code matrix and data distribution to showcase the variety of implicit structures that can be exploited. For the code matrix, we consider the case when the codewords are well separated (i.e., the output code is error correcting), the case of one-vs-all (where the code matrix is the identity), and a natural boundary features condition. These conditions can loosely be compared in terms of the Hamming distance between codewords. In the case of error correcting output codes, the distance between codewords is large (at least $d + 1$ when the data is d-dimensional), in one-vs-all the distance is always exactly 2, and finally in the boundary features condition the distance can be as small as 1. In the latter cases, the lower Hamming distance requirement is balanced by other structure in the code matrix. For the distribution, we either assume that the data density function satisfies a thick level set condition or

that the density is upper and lower bounded on its support. Both regularity conditions are used to ensure that the geometric structure implied by the consistent output code will be recoverable based on a sample of data.

Error correcting output codes. We first showcase how to exploit the implicit structure assumed by the commonly used and natural case of linear output codes where the Hamming distance between codewords is large. In practice, output codes are designed to have this property in order to be robust to prediction errors for the binary classification tasks [32]. We suppose that the output code makes at most β errors when predicting codewords and has codewords with Hamming distance at least $2\beta + d + 1$ in a d-dimensional problem. The key insight is that when the code words are well separated, this implies that points belonging to different classes must be geometrically separated as well. This suggests that tight clusters of data will be label-homogeneous, so we should be able to learn an accurate classifier using only a small number of label queries per cluster. The main technical challenge is to show that our clustering algorithm will not produce too many clusters (in order to keep the label complexity controlled), and that with high probability, a new sample from the distribution will have the same label as its nearest cluster. We show that when the data density satisfies a thick-level set condition (requiring that its level sets do not have bridges or cusps that are too thin), then a single-linkage clustering algorithm can be used to recover a small number of label-homogeneous clusters.

One-vs-all. Next, we consider the classic one-vs-all setting for data in the unit ball. This is an interesting setting because of the popularity of one-vs-all classification and because it significantly relaxes the assumption that the codewords are well separated (in a one-vs-all classifier, the Hamming distance between codewords is exactly 2). The main challenge in this setting is that there need not be a margin between classes and a simple single-linkage style clustering might group multiple classes into the same cluster. To overcome this challenge, we show that the classes are probabilistically separated in the following sense: after projecting onto the surface of the unit ball, the level sets of the projected density are label-homogeneous. Equivalently, the high-density regions belonging to different classes must be separated by low-density regions. We exploit this structure by estimating the connected components of the ϵ level set using a robust single-linkage clustering algorithm.

The boundary features condition. We introduce an interesting and natural condition on the code matrix capturing the intuition that every binary learning task should be significant. This condition has the weakest separation requirement, allowing the codewords to have a Hamming distance of only 1. This setting is our most challenging, since it allows for the classes to be very well connected to one another, which prevents clustering or level set estimation from being used to find a small number of label-homogeneous clusters. Nevertheless, we show that the implicit geometric structure implied by the output code can be exploited to learn using a small number of label queries. In this case, rather than clustering the unlabeled sample, we apply a novel hyperplane-detection algorithm that uses the *absence* of data to learn local information about the boundaries between classes. We then use the implicit structure of the output code to extend these local boundaries into a globally accurate classifier.

Agnostic Setting. Finally, we show that our results for all three settings can be extended to an agnostic learning scenarios, where we do not assume that there exists a consistent output code classifier

and the goal is to compete with the best linear output code.

Our results show an interesting trend: when linear output codes are able to learn from labeled data, it is possible to exploit the same underlying structure in the problem to learn using a small number of label requests. Our results hold under several natural assumptions on the output code and general conditions on the data distribution, and employ both clustering and hyperplane detection strategies to reduce the label complexity of learning.

2.3 Related Work

Reduction to binary classification is one of the most widely used techniques in applied machine learning for multi-class problems. Indeed, the one-vs-all, one-vs-one, and the error correcting output code approaches [32] all follow this structure [58, 54, 17, 29, 1].

There is no prior work providing error bounds for output codes using unlabeled data and interaction. There has been a long line of work for providing provable bounds for semi-supervised learning [9, 7, 18, 24] and active learning [10, 30, 8, 44]. These works provide bounds on the benefits of unlabeled data and interaction for significantly different semi-supervised and active learning methods that are based different assumptions, often focusing on binary classification, thus the results are largely incomparable. Another line of recent work considers the multi-class setting and uses unlabeled data to consistently estimate the risk of classifiers when the data is generated from a known family of models [34, 5, 6]. Their results do not immediately imply learning algorithms and they consider generative assumptions, while in contrast our work explicitly designs learning algorithms under commonly used discriminative assumptions.

Another work related to ours is that of Balcan et al. [12], where labels are recovered from unlabeled data. The main tool that they use in order to recover the labels is the assumption that there are multiple views and an underlying ontology that are known, and restrict the possible labeling. Similarly, Steinhardt and Liang [67] show how to use the method of moments to estimate the risk of a model from unlabeled data under the assumption that the data has three independent views. Our work is more widely applicable, since it applies when we have only a single view.

The output-code formalism is also used by Palatucci et al. [62] for the purpose of zero shot learning. They demonstrate that it is possible to exploit the semantic relationships encoded in the code matrix to learn a classifier from labeled data that can predict accurately even classes that *did not appear in the training set*. These techniques make very similar assumptions to our work but require that the code matrix C is known and the problem that they solve is different.

Chapter 3

Dispersion and Piecewise Lipschitz Optimization

3.1 Introduction

In this chapter, we consider the problem of optimizing sums of piecewise Lipschitz functions in a variety of contexts connecting with two themes of the thesis: incorporating social values into machine learning and machine learning beyond prediction. We consider two optimization settings: first, we suppose that each piecewise Lipschitz function encodes sensitive information about one individual (such as their preference for various outcomes or their values for items), and our goal is to approximately maximize the functions while at the same time not revealing information about any individual. Second, we consider online optimization of piecewise Lipschitz functions, where the functions arrive one at a time chosen by an adversary and the learner's goal is to find a sequence of points that is competitive with the best fixed point in hindsight. Interestingly, we develop a single set of technical tools that are applicable to both settings. Private and online optimization of piecewise Lipschitz functions can be applied to machine learning tasks beyond standard prediction problems. In particular, in this chapter we will consider applications to data-driven algorithm configuration or parameter tuning.

Much of the prior work on both online and private optimization has focused on settings where the functions being optimized have considerable structure. For example, the functions may be linear, convex, or globally Lipschitz. The main additional challenge in dealing with *piecewise* Lipschitz functions is that they are permitted to have discontinuities. These discontinuities can be used to construct worst-case instances for which no optimization algorithm can have non-trivial performance guarantees while preserving privacy or in the online setting. Our main contribution is to introduce a sufficient and general condition, called *dispersion*. When satisfied, dispersion implies nontrivial performance guarantees for private and online optimization of piecewise Lipschitz functions. Moreover, we show that dispersion is satisfied in a variety of important optimization problems related to algorithm configuration under very mild assumptions.

In data-driven algorithm configuration, our goal is to choose the best algorithm parameters for a specific application. Rather than use off-the-shelf algorithms with only worst-case guarantees, a practitioner will often optimize over a family of parametrized algorithms, tuning the algorithm's parameters based on typical problems from his domain. Ideally, the resulting algorithm will have high performance on future problems, but these procedures have historically come with no guarantees. In a seminal work, Gupta and Roughgarden [43] study algorithm selection in a distributional learning

setting. Modeling an application domain as a distribution over typical problems, they show that a bound on the intrinsic complexity of the algorithm family prescribes the number of samples sufficient to ensure that any algorithm's empirical and expected performance are close. Our results advance the foundations of algorithm configuration in several important directions: online and private algorithm selection. In the online setting, problem instances arrive one-by-one, perhaps chosen by an adversary. The goal is to select parameters for each instance in order to minimize *regret*, which is the difference between the cumulative performance of those parameters and the optimal parameters in hindsight. We also study private algorithm selection, where the goal is to find high-performing parameters over a set of problems without revealing sensitive information contained therein. Preserving privacy is crucial when problems depend on individuals' medical or purchase data, for example.

We analyze several important, infinite families of parameterized algorithms. These include greedy techniques for canonical subset selection problems such as the knapsack and maximum weight independent set problems. We also study SDP-rounding schemes for problems that can be formulated as integer quadratic programs, such as max-cut, max-2sat, and correlation clustering. In these cases, our goal is to optimize, online or privately, the utility function that measures an algorithm's performance as a function of its parameters, such as the value of the items added to the knapsack by a parameterized knapsack algorithm.

This project is joint work with Maria-Florina Balcan and Ellen Vitercik, and our existing results appeared in FOCS 2018 [15].

3.2 Preliminaries and Current Results

Our goal is to optimize the sum of piecewise Lipschitz functions, either in the online or private settings. In this section we briefly introduce notation, describe the two optimization settings, and state our main results.

Definition 1. A function $u : \mathcal{C} \to \mathbb{R}$ with domain $\mathcal{C} \subset \mathbb{R}^d$ is piecewisewise L-Lipschitz if there exists a partitioning $\mathcal{C}_1, \ldots, \mathcal{C}_K$ of $\mathcal{C}$ so that u is L-Lipschitz when restricted to each $\mathcal{C}_i$. That is, for every set $\mathcal{C}_i$ and $\rho, \rho' \in \mathcal{C}_i$, we have $|u(\rho) - u(\rho')| \leq L \cdot \|\rho - \rho'\|_2$.

We will refer to the partition $\mathcal{C}_1, \ldots \mathcal{C}_K$ on which the function u is piecewise Lipschitz as the Lipschitz partitioning of u, or simply the partitioning of u.

Differentially Private Optimization. First, we consider approximately maximizing the sum of piecewise Lipschitz functions while preserving privacy. We use the framework of Differential Privacy [36], which requires algorithms to be randomized and have the property that their output distribution is insensitive to any single individual in the dataset. Intuitively, this condition leads to privacy for the following reason: suppose that $\mathcal{S}$ and $\mathcal{S}'$ are two collections of piecewise Lipschitz functions differing on only my function (recall that each function encodes information about a single individual). Differential privacy requires that the output distribution of the algorithm when run on S and S' must be similar, and therefore the two output distributions are hard to distinguish based on a single sample (i.e., the output of the algorithm). This means that an adversary who observes the output of the algorithm cannot accurately infer whether the input was S or S'; that is, whether my real data or the lie was used.

Formally, we say that two collections of functions $\mathcal{S} = \{u_1, \ldots, u_T\}$ and $\mathcal{S}' = \{u'_1, \ldots, u'_T\}$ are neighboring if they differ on at most one function. That is $u_i = u'_i$ for all but at most one index

i. When each function encodes information about an individual and no individual is represented by more than one function, neighboring collections differ on the contribution of exactly one individual. With this, differential privacy is the following formal requirement:

Definition 2. A randomized optimization algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private if for any neighboring collections of functions $\mathcal{S}$ and $\mathcal{S}'$ and any subset $\mathcal{O} \subset \mathcal{C}$ of possible outputs, we have

$$\Pr(\mathcal{A}(\mathcal{S}) \in \mathcal{O}) \leq e^{\epsilon} \Pr(\mathcal{A}(\mathcal{S}') \in \mathcal{O}) + \delta,$$

where the probability is only taken over the algorithm's internal randomness.

Our goal is to design (ϵ, δ)-differentially private optimization algorithms that, given a set of piecewise Lipschitz functions $\mathcal{S} = \{u_1, \ldots, u_T\}$, each mapping $\mathcal{C} \subset \mathbb{R}^d$ to $\mathbb{R}$, outputs a point $\hat{\rho} \in \mathcal{C}$ that approximately maximizes the sum $\sum_{t=1}^{T} u_t(\cdot)$.

Online Optimization. Second, we consider online optimization of piecewise Lipschitz functions. This is a game played between the learner and an adversary over T rounds. On each round, the adversary chooses a piecewise Lipschitz function $u_t : \mathcal{C} \to \mathbb{R}$ and the learner chooses a point $\rho_t \in \mathcal{C}$. The learner obtains a reward equal to $u_t(\rho_t)$ and receives information about the adversary's choice u_t. In the *full information* setting, the learner observes the entire function $u_t : \mathcal{C} \to \mathbb{R}$. Another common regime is *bandit feedback*, where the learner only observes the scalar value $u_t(\rho_t)$. The learner's goal is to minimize their regret over the T rounds, which is defined by

$$\text{Regret}(T) = \sup_{\rho \in \mathcal{C}} \sum_{t=1}^{T} u_t(\rho) - \sum_{t=1}^{T} u_t(\rho_t).$$

The regret compares the learner's cumulative total utility to the total utility of the best fixed parameter in hindsight. If the learner's regret grows sublinearly with T, then their average per-round reward converges to that of the optimal fixed point in hindsight. Slower regret growth rates correspond to faster convergence.

Dispersion. Next, we introduce our main definition, dispersion, which is a constraint on a collection of piecewise Lipschitz functions. Let $u_1, \ldots, u_T$ be a collection of functions mapping a set $\mathcal{C} \subseteq \mathbb{R}^d$ to $[0, 1]$. Dispersion is a constraint on the functions $u_1, \ldots, u_T$ that limits how concentrated their discontinuities can be. We say that a partition $\mathcal{C}_1, \ldots, \mathcal{C}_K$ of $\mathcal{C}$ *splits* a set $A \subset \mathbb{R}^d$ if A intersects with at least two parts of the partition. When $\mathcal{C}_1, \ldots, \mathcal{C}_K$ is the Lipschitz partitioning of a function $u : \mathcal{C} \to [0, 1]$, a set A is split by $\mathcal{C}_1, \ldots, \mathcal{C}_K$ if it contains a discontinuity for the function u.

Definition 3. A collection of piecewise Lipschitz functions $u_1, \ldots, u_T : \mathcal{C} \to [0, 1]$ is (w, k)-*dispersed at a point* $\rho \in \mathcal{C}$ if the Euclidean ball $B(\rho, w)$ is split by the Lipschitz partition for at most k of the functions $u_1, \ldots, u_T$. We say they are (w, k)-*dispersed (globally)* if they are (w, k)-dispersed at all $x \in \mathcal{C}$, and (w, k)-*dispersed at a maximizer* if they are (w, k)-dispersed for some $x^* \in \text{argmax}_{x \in \mathcal{C}} \sum_{t=1}^{T} u_t(x)$.

Dispersion guarantees that, although each function u_t may have discontinuities, not too many can have discontinuities in any region of radius w. Every collection of T piecewise Lipschitz functions is (w, k)-dispersed for a variety of parameters. For example, since there are only T functions, they

will always be (w, T)-dispersed for any w. Our main results show that for "good" (and very reasonable) values of w and k, this condition implies sublinear regret bounds in both full information and bandit settings, and strong utility guarantees for differentially private optimization. We also provide matching lower bounds, showing that there are both online and private optimization problems where our dispersion-based bounds are unimprovable.

Online learning We prove that dispersion implies strong regret bounds in online learning. Under full information, we show that the exponentially-weighted forecaster [21] has regret bounded by $\tilde{O}(\sqrt{Td}+TLw+k)$. When $w = 1/(L\sqrt{T})$ and $k = \tilde{O}(\sqrt{T})$, this results in $\tilde{O}(\sqrt{Td})$ regret. We also prove a matching lower bound for the case of piecewise constant functions. Under bandit feedback, we show that a discretization-based algorithm achieves regret at most $\tilde{O}(\sqrt{Td(3R/w)^d}+TLw+k)$. When $w = T^{-1/(d+2)}$ and $k = \tilde{O}(T^{(d+1)/(d+2)})$, this gives a bound of $\tilde{O}(T^{(d+1)/(d+2)}(\sqrt{d(3R)^d}+L))$, matching the dependence on T of a lower bound by Kleinberg et al. [52] for (globally) Lipschitz functions.

Private batch optimization Dispersion also implies strong utility guarantees for differentially private optimization. We show that the exponential mechanism [56] outputs $\hat{\rho} \in \mathcal{C}$ such that with high probability $\frac{1}{T}\sum_{i=1}^{T} u_i(\hat{\rho}) \geq \max_{\rho\in\mathcal{C}} \frac{1}{T}\sum_{i=1}^{T} u_i(\rho) - \tilde{O}(\frac{d}{T\epsilon} + \frac{k}{T} + Lw)$ while preserving $(\epsilon, 0)$-differential privacy. When $w = 1/(L\sqrt{T})$ and $k = \tilde{O}(\sqrt{T})$, this leads to a suboptimality of $O(\frac{d}{T\epsilon} + \frac{1}{\sqrt{T}})$. As the number T of individuals in the dataset grows, the error due to differential privacy goes to zero, implying that we can nearly perfectly optimize large collections of dispersed piecewise Lipschitz functions. We also give a matching lower bound. An important property of our analysis is that the exponentially mechanism *always* preserves $(\epsilon, 0)$-differential privacy, and our dispersion condition is only necessary for meaningful utility guarantees.

Application to algorithm selection. In data-driven algorithm configuration, our goal is to tune the parameters of an algorithm $\mathcal{A}$ to get the highest utility for a specific application. More formally, there is a space Π of possible problem instances or inputs to the algorithm, and a parameter space $\mathcal{C} \subset \mathbb{R}^d$. For each problem instance $x \in \Pi$ and parameter vector $\rho \in \mathcal{C}$, we let $u(x, \rho)$ denote the utility of running algorithm $\mathcal{A}$ with parameters ρ on problem instance x, where $u : \Pi \times \mathcal{C} \to [0, 1]$ is a known utility function. The connection between algorithm configuration and piecewise Lipschitz optimization is that in many algorithm configuration problems, for each fixed problem instance $x \in \Pi$, the function $\rho \mapsto u(x, \rho)$ is a piecewise Lipschitz function of the algorithm parameters. In online algorithm configuration, an adversary chooses a sequence of problem instances $x_1, \ldots, x_T$, and the learner must choose parameters for each instance to maximize utility. In the private setting, the learner obtains a collection of problem instances where each instance encodes information about one individual, and their goal is to output a parameter with high average utility for the collection while at the same time not divulging information about any individuals. These two problems are natural instances of online and private optimization of piecewise Lipschitz functions.

For example, we consider algorithm configuration for greedy subset selection algorithms for the classic knapsack problem. A knapsack problem instance is specified by a collection of n items, where item i has a value v_i and a size s_i. Given a knapsack capacity K, the algorithm's goal is to find the highest-value subset of items that fits into the knapsack. Gupta and Roughgarden [43] propose the following parameterized greedy algorithm for knapsack problems: assign a score $s(i)$ to each item and greedily add items in decreasing order of the score until the knapsack is full. They consider a

parameterized scoring rule $s_\rho(i) = v_i/s_i^\rho$, where the parameter ρ interpolates between scoring items by their value ($\rho = 0$), by their value-per-unit-size ($\rho = 1$), and scoring them by their size alone (as $\rho \to \infty$). Given a specific application domain (or source of knapsack problem instances), a natural goal is to find the value of the parameter ρ that leads to the highest value. In this case, the set Π corresponds to knapsack instances, $\mathcal{C} = [0, \infty)$, and the utility function $u(x, \rho)$ measures the total value obtained by the greedy algorithm. This utility function is piecewise constant: the items chosen by the algorithm depend only on the ordering of the items induced by the score, and the relative ordering of two items i and j only changes at the single solution to the equation $s_\rho(i) = s_\rho(j)$. Therefore, the algorithm's output is piecewise constant with at most n^2 pieces.

We show that when the knapsack problem instances $x_1, \ldots, x_T \in \Pi$ are chosen by a *smoothed adversary*, in the sense that Gaussian noise with standard deviation σ is added to each item value (and item values and sizes are constrained appropriately), then the corresponding collection of utility functions $\{u_t(\rho) = u(x_t, \rho)\}_{t=1}^T$ is (w, k)-dispersed for any $w > 0$ and $k = O\big(n^2 T w\sigma + n^2\sqrt{T\log(1/\delta)}\big)$ with probability at least $1 - \delta$. In particular, choosing $w = 1/(\sigma\sqrt{T})$ ensures (w, k)-dispersion with $k = \tilde{O}(n^2\sqrt{T})$, leading to good regret bounds in the full information setting and high utility for differentially private optimization. Choosing w appropriately, we also get regret bounds in the bandit-feedback setting. Similar arguments can be made for tuning the parameters of greedy algorithms for maximum weight independent set problems.

In our FOCS 2018 paper [15], we also study algorithm configuration for integer quadratic programs (IQPs) of the form $\max_{\vec{z}\in\{\pm 1\}^n} \vec{z}^\top A\vec{z}$, where $A \in^{n\times n}$ for some n. Many classic NP-hard problems can be formulated as IQPs, including max-cut [42], max-2SAT [42], and correlation clustering [25]. Many IQP approximation algorithms are semidefinite programming (SDP) rounding schemes; they solve the SDP relaxation of the IQP and round the resulting vectors to binary values. We study two families of SDP rounding techniques: s-linear rounding [37] and outward rotation [78], which include the Goemans-Williamson algorithm [42] as a special case. Due to these algorithms' inherent randomization, finding an optimal rounding function over T problem instances with n variables amounts to optimizing the sum of $(1/T^{1-\alpha}, \tilde{O}(nT^\alpha))$-dispersed functions for any $1/2 \le \alpha < 1$.

3.3 Related Work

Gupta and Roughgarden [43] and Balcan et al. [14] study algorithm selection in the distributional learning setting, where there is a distribution $\mathcal{P}$ over problem instances. A learning algorithm receives a set S of samples from $\mathcal{P}$. Those two works provide *uniform convergence guarantees*, which bound the difference between the average performance over S of any algorithm in a class $\mathcal{A}$ and its expected performance on $\mathcal{P}$. It is known that regret bounds imply generalization guarantees for various online-to-batch conversion algorithms [22], but in this work, we also show that dispersion can be used to explicitly provide uniform convergence guarantees via Rademacher complexity. Beyond this connection, our work is a significant departure from these works since we give guarantees for private algorithm selection and we give no regret algorithms, whereas Gupta and Roughgarden [43] only study online MWIS algorithm selection, proving their algorithm has small constant per-round regret.

Private empirical risk minimization (ERM). The goal of private ERM is to find the best machine learning model parameters based on private data. Techniques include objective and output perturbation [26], stochastic gradient descent, and the exponential mechanism [16]. These works focus on

minimizing data-dependent convex functions, so parameters near the optimum also have high utility, which is not the case in our settings.

Private algorithm configuration. Kusner et al. [53] develop private Bayesian optimization techniques for tuning algorithm parameters. Their methods implicitly assume that the utility function is differentiable. Meanwhile, the class of functions we consider have discontinuities between pieces, and it is not enough to privately optimize on each piece, since the boundaries themselves are data-dependent.

Online optimization. Prior work on online algorithm selection focuses on significantly more restricted settings. Cohen-Addad and Kanade [27] study single-dimensional piecewise constant functions under a "smoothed adversary," where the adversary chooses a distribution per boundary from which that boundary is drawn. Thus, the boundaries are independent. Moreover, each distribution must have bounded density. Gupta and Roughgarden [43] study online MWIS greedy algorithm selection under a smoothed adversary, where the adversary chooses a distribution per vertex from which its weight is drawn. Thus, the vertex weights are independent and again, each distribution must have bounded density. In contrast, we allow for more correlations among the elements of each problem instance. Our analysis also applies to the substantially more general setting of optimizing piecewise Lipschitz functions. We show several new applications of our techniques in algorithm selection for SDP rounding schemes, price setting, and auction design, none of which were covered by prior work. Furthermore, we provide differential privacy results and generalization guarantees.

Neither Cohen-Addad and Kanade [27] nor Gupta and Roughgarden [43] develop a general theory of dispersion, but we can map their analysis into our setting. In essence, Cohen-Addad and Kanade [27], who provide the tighter analysis, show that if the functions the algorithm sees map from $[0, 1]$ to $[0, 1]$ and are $(w, 0)$-dispersed at a maximizer, then the regret of their algorithm is bounded by $O(\sqrt{T \ln(1/w)})$. In this work, we show that using the more general notion of (w, k)-dispersion is essential to proving tight learning bounds for more powerful adversaries. We provide a sequence of piecewise constant functions $u_1, \ldots, u_T$ that are $(1/4, \sqrt{T})$-dispersed, which means that our regret bound is $O(\sqrt{T \log(1/w)} + k) = O(\sqrt{T})$. However, these functions are not $(w, 0)$-dispersed at a maximizer for any $w \geq 2^{-T}$, so the regret bound by Cohen-Addad and Kanade [27] is trivial, since $\sqrt{T \log(1/w)}$ with $w = 2^{-T}$ equals T. Similarly, Weed et al. [74] and Feng et al. [38] use a notion similar to $(w, 0)$-dispersion at a maximizer to prove learning guarantees for the specific problem of learning to bid, as do Rakhlin et al. [64] for learning threshold functions under a smoothed adversary.

Our online bandit results are related to those of Kleinberg [50] for the "continuum-armed bandit" problem. They consider bandit problems where the set of arms is the interval $[0, 1]$ and each payout function is uniformly locally Lipschitz. We relax this requirement, allowing each payout function to be Lipschitz with a number of discontinuities. In exchange, we require that the overall sequence of payout functions is fairly nice, in the sense that their discontinuities do not tightly concentrate. The follow-up work on Multi-armed Bandits in Metric Spaces [52] considers the stochastic bandit problem where the space of arms is an arbitrary metric space and the mean payoff function is Lipschitz. They introduce the zooming algorithm, which has better regret bounds than the discretization approach of Kleinberg [50] when either the max-min covering dimension or the (payout-dependent) zooming dimension are smaller than the covering dimension. In contrast, we consider optimization over $\mathbb{R}^d$ under the ℓ_2 metric, where this algorithm does not give improved regret in the worst case.

3.4 Proposed Work

Improved online algorithms. For full information online optimization, our current results show that the exponentially weighted forecaster has sublinear regret bounds under dispersion. However, there are many other classic algorithms that often have better running time, even in the full information setting. For example, the Follow the Perturbed Leader algorithm [20] can often be implemented extremely efficiently given efficient algorithms for solving the offline optimization problem. In contrast, we are unable to take advantage of efficient offline optimizers when implementing the exponentially weighted forecaster.

Additionally, our paper only provides proof-of-concept algorithms and regret bounds for online piecewise Lipschitz optimization under bandit feedback. Given the dispersion parameters w and k, our algorithm discretizes the feasible set $\mathcal{C}$ using a w-net and applies the EXP3 algorithm [3] to play the finite-armed bandit defined by the points in the discretization. There are a number of drawbacks to this approach: first, while the regret bound has the optimal exponent on T, the dependence on the dimension and Lipschitz constant are likely suboptimal. Second, the size of the discretization grows exponentially with the dimension of $\mathcal{C}$ and the learner must maintain statistics about each point, leading to running time and memory usage exponential in the dimension. Finally, since the algorithm must commit to the discretization before learning begins, this requires the algorithm to know the (w, k)-dispersion parameters in advance, rather than adapting to the best post-hoc parameters. One promising direction for designing better bandit algorithms is to adapt the techniques of barycentric [4] and volumetric spanners [46] developed for online linear optimization to our setting.

Another promising algorithmic direction for the bandit setting is to adapt the *zooming algorithm* of Kleinberg et al. [51]. This algorithm was originally proposed for bandit problems where the set of arms forms a metric space, each arm has i.i.d. payouts, and the mean payout of arms is 1-Lipschitz with respect to the underlying metric. The zooming algorithm uses an adaptive discretization of the set of arms that gradually adds additional discretization points in regions of the arm-space that appear to have high expected payout. We plan to extend the techniques used in the zooming algorithm in order to maintain an adaptive discretization used for online optimization of piecewise Lipschitz functions. This will lead to algorithms that do not need to know the (w, k)-parameters in advance. The main technical challenge is that the granularity required in a region of the parameter space to guarantee that a discretization closely approximates all nearby points depends not only on the Lipschitz constant L, but also on the concentration of discontinuities in that region. Given that the learner does not directly observe the discontinuity locations under bandit feedback, this may require more elaborate discretization refinement conditions than in the original zooming algorithm. Alternatively, we can explore learning under a richer form of feedback that includes information about nearby discontinuities (in many algorithm configuration applications, this information is readily available).

Semi-bandit feedback. Many online piecewise Lipschitz optimization problems are between the full information and bandit feedback settings, with the learner observing some portion of the utility function u_t containing more points than their choice ρ_t. For example, in many algorithm configuration problems, it turns out that after running the algorithm with a given parameter ρ_t, we learn not only the scalar utility $u_t(\rho_t)$, but also the entire Lipschitz piece of u_t containing ρ_t. A similar situation for finite armed bandits is explored by Alon et al. [2]. They suppose that there is a graph whose nodes are the finite set of available arms and after playing an arm, the learner observes the payouts for that arm and its neighbors. They design algorithms whose regret scales with the size of the smallest

dominating set for the feedback graph. We plan to extend their techniques to the semi-bandit feedback setting in online piecewise Lipschitz optimization. We require adaptations of their methods to handle cases where there are infinitely many available actions, and when the graph feedback structure is determined by the sequence of functions chosen by the adversary.

Application to linkage-based clustering. We also to apply (w, k)-dispersion to online optimization of the parameters for α-linkage clustering algorithms. Since these algorithms are deterministic, in order for (w, k)-dispersion to be satisfied, we need to assume some small amount of randomness in the input problem instances. One very mild assumption is to suppose that the adversary is smoothed, in the sense that they choose the locations of the points in the clustering instance, but then nature corrupts those points by adding a small amount of independent Gaussian noise to each point. The locations of the discontinuities in the clustering utility function will depend on the added noise, causing them to be "spread out" or dispersed. The main challenge in carrying out a smoothed analysis of the linkage algorithms is that a single run of the algorithm on n points involves performing $n - 1$ merges, and every merge depends on the noise added to every point. This implies that there will be significant correlation across the rounds, and understanding these correlations on some level will be required for the dispersion analysis. One possibly simplifying factor is that we only need to analyze the probability that any discontinuity of the utility function lands in a given interval I, and this may not require a complete characterization of the correlations.

Thorough empirical studies. In addition to our current set of formal guarantees, I plan to carry out thorough experiments investigating dispersion-based algorithms for online and private algorithm configuration. These experiments will cover a range of different algorithms including linkage-based clustering and Lloyd's method. An important empirical question I hope to answer is whether or not bandit learning algorithms can be used to more efficiently find optimal parameters for a collection of problem instances, despite their slower rate of convergence in theory.

Chapter 4

Envy-free Classification

4.1 Introduction

The study of fairness in machine learning is driven by an abundance of examples where learning algorithms were perceived as discriminating against protected groups [69, 31]. Addressing this problem requires a conceptual — perhaps even philosophical — understanding of what fairness means in this context. In other words, the million dollar question is this: What are the formal constraints that fairness imposes on learning algorithms?

Most of the answers proposed so far [55, 35, 77, 45, 48, 76] focus on situations where each individual is either assigned a favorable outcome (e.g., being released on bail or being given a loan) or not. The goal is to ensure that certain statistical properties of the outcome distribution are approximately equal across the protected subgroups. For example, equal opportunity requires the favorable outcome to be assigned at the same rate to each protected group as it is on the whole population.

In this this work, we consider situations where there is a diverse set of possible outcomes and individuals have heterogeneous preferences for those outcomes. Our proposed fairness notion draws on an extensive body of work on rigorous approaches to fairness, which has not been tapped by machine learning researchers: the literature on *fair division* [19, 59]. The most prominent notion is that of *envy-freeness* [39, 73], which, in the context of the allocation of goods, requires that the utility of each individual for his allocation be at least as high as his utility for the allocation of any other individual; this is the gold standard of fairness for problems such as cake cutting [65, 63] and rent division [68, 41]. In the classification setting, envy-freeness would simply mean that the utility of each individual for his distribution over outcomes is at least as high as his utility for the distribution over outcomes assigned to any other individual.

For example, consider a system responsible for displaying credit card advertisements to individuals. There are many credit cards with different eligibility requirements, annual rates, and reward programs. An individual's utility for seeing a card's advertisement will depend on their eligibility and their benefit from the rewards programs and potentially other factors. It may well be the case that an envy-free advertisement assignment shows Bob advertisements for a card with worse annual rates than those shown to Alice, but this outcome is not unfair if bob is genuinely more interested in the card offered to him. Such rich utility functions are also evident in the context of job advertisements [31]: people generally want higher paying jobs, but would presumably have higher utility for seeing advertisements for jobs that better fit their qualifications and interests.

It is worth noting that the classification setting is different from classic fair division problems in

that the "goods" (outcomes) are non-excludable. In fact, one envy-free solution simply assigns each individual to his favorite outcome; but when the loss function disagrees with the utility functions, it may be possible to achieve smaller loss without violating the envy-freeness constraint.

A second appealing property of envy-freeness is that its fairness guarantee binds at the level of individuals. Fairness notions can be coarsely characterized as being either individual notions, or group notions, depending on whether they provide guarantees to specific individuals, or only on average to a protected subgroup. The majority of work on fairness in machine learning focuses on group fairness.

The best known example of individual fairness is the influential fair classification model of Dwork et al. [35]. The model involves a set of individuals and a set of outcomes. The centerpiece of the model is a *similarity metric* on the space of individuals; it is specific to the classification task at hand, and ideally captures the ethical ground truth about relevant attributes. For example, a man and a woman who are similar in every other way should be considered similar for the purpose of credit card offerings, but perhaps not for lingerie advertisements. Assuming such a metric is available, fairness can be naturally formalized as a Lipschitz constraint, which requires that individuals who are close according to the similarity metric be mapped to distributions over outcomes that are close according to some standard metric (such as total variation). The algorithmic problem is then to find a classifier that minimizes loss, subject to the Lipschitz constraint.

As attractive as this model is, it has one clear weakness from a practical viewpoint: the availability of a similarity metric. Dwork et al. [35] are well aware of this issue; they write that justifying this assumption is "one of the most challenging aspects" of their approach. They add that "in reality the metric used will most likely only be society's current best approximation to the truth." But, despite recent progress on automating ethical decisions in certain domains [61, 40], the task-specific nature of the similarity metric makes even a credible approximation thereof seem unrealistic. In particular, if one wanted to learn a similarity metric, it is unclear what type of examples a relevant dataset would consist of.

In place of a metric, envy-freeness requires access to individuals' utility functions, but — in stark contrast — we do not view this assumption as a barrier to implementation. Indeed, there are a variety of techniques for learning utility functions [23, 60, 11]. Moreover, in our running example of advertising, one can even think of standard measures like expected click-through rate (CTR) as an excellent proxy for utility.

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

This project is joint work with Maria-Florina Balcan, Ariel Procaccia, and Ritesh Noothigattu.

4.2 Preliminaries and Current Results

We assume that there is a space $\mathcal{X}$ of individuals, a finite space $\mathcal{Y}$ of outcomes, and a utility function $u : \mathcal{X} \times \mathcal{Y} \to [0, 1]$ encoding the preferences of each individual for the outcomes in $\mathcal{Y}$. In the advertising example, individuals are users, outcomes are advertisements, and the utility function reflects the benefit an individual derives from being shown a particular advertisement. For any distribution $p \in \Delta(\mathcal{Y})$ (where $\Delta(\mathcal{Y})$ is the set of distributions over $\mathcal{Y}$) we let $u(x, p) = \mathbb{E}_{y\sim p}[u(x, y)]$ denote individual x's expected utility for an outcome sampled from p.

Our goal is to learn a classifier h that assigns individuals to outcomes. It turns out that envy-freeness is a very strong constraint on deterministic classifers, so we allow the assignment output by

h to be randomized. That is, our goal is to find a map $h : \mathcal{X} \to \Delta(\mathcal{Y})$ that assigns a distribution over outcomes to each individual. We still refer this function as a classifier.

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

Definition 4. A classifier $h : \mathcal{X} \to \Delta(\mathcal{Y})$ is *envy free (EF)* on a set S of individuals if $u(x, h(x)) \geq u(x, h(x'))$ for all $x, x' \in S$. Similarly, h is (α, β)-*EF* with respect to a distribution P on $\mathcal{X}$ if

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

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

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

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

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

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

Current results. The technical challenge we face is that the space of individuals is potentially huge, yet we seek to provide universal envy-freeness guarantees. To this end, we are given a sample consisting of individuals drawn from an unknown distribution. We are interested in learning algorithms that minimize loss, subject to satisfying the envy-freeness constraint, *on the sample*. Our primary technical question is that of generalizability, that is, *given a classifier that is envy free on a sample, is it approximately envy free on the underlying distribution?*

Motivated by classic generalization results, we focus on structured families of classifiers. On a high level, our goal is to relate the combinatorial richness of the family to generalization guarantees. One obstacle is that standard notions of dimension do not extend to the analysis of randomized classifiers, whose range is *distributions* over outcomes (equivalently, real vectors). We circumvent this obstacle by considering mixtures of *deterministic* classifiers that belong to a family of bounded Natarajan dimension (an extension of the well-known VC dimension to multi-class classification). For any class $\mathcal{G} = \{g : \mathcal{X} \to \mathcal{Y}\}$ of deterministic classifiers, we let $\mathcal{H}(\mathcal{G}, m)$ denote the set of randomized classifiers that can be written as a random combination of m classifiers from $\mathcal{G}$. Each $h \in \mathcal{H}(\mathcal{G}, m)$ is described by a collection of functions $g_1, \ldots, g_m$ together with mixing weights $\alpha_1, \ldots, \alpha_m \in [0, 1]$ with $\sum_i \alpha_i = 1$. For an individual $x \in \mathcal{X}$, the classifier h chooses function g_i with probability α_i

and outputs $g_i(x)$. Our main technical result asserts that envy-freeness on a sample does generalize to the underlying distribution for classifiers belonging to the class $\mathcal{H}(\mathcal{G}, m)$.

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

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

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

4.3 Related Work

Conceptually, our work is most closely related to work by Zafar et al. [76]. They are interested in group notions of fairness, and advocate preference-based notions instead of parity-based notions. In particular, they assume that each group has a utility function for *classifiers*, and define the *preferred treatment* property, which requires that the utility of each group for its own classifier be at least its utility for the classifier assigned to any other group. Their model and results focus on the case of binary classification where there is a desirable outcome and an undesirable outcome, so the utility of a group for a classifier is simply the fraction of its members that are mapped to the desirable outcome. Although, at first glance, this notion seems similar to envy-freeness, it is actually fundamentally different. Our paper is also completely different from that of Zafar et al. in terms of technical results; theirs are purely empirical in nature, and focus on the increase in accuracy obtained when parity-based notions of fairness are replaced with preference-based ones.

Very recent, concurrent work by Rothblum and Yona [66] provides generalization guarantees for the metric notion of individual fairness introduced by Dwork et al. [35], or, more precisely, for an approximate version thereof. There are two main differences compared to our work: first, we propose envy-freeness as an alternative notion of fairness that circumvents the need for a similarity metric. Second, they focus on randomized *binary* classification, which amounts to learning a real-valued function, and so are able to make use of standard Rademacher complexity results to show generalization. By contrast, standard tools do not directly apply in our setting. It is worth noting that several other papers provide generalization guarantees for notions of group fairness, but these are more distantly related to our work [77, 75, 33, 49, 47].

4.4 Proposed Work

Improved dependence on m. One undesirable property of the sample complexity provided by Theorem 1 is the dependence on m^2, where m is the number of deterministic functions we are allowed to mix. A classic result for Rademacher complexity shows that for any function class $\mathcal{F}$, the class consisting of convex combinations of functions in $\mathcal{F}$ has exactly the same Rademacher comlexity. Since our randomized classifiers are convex combinations of deterministic classifiers, it seems that the dependence on m should not appear in our bounds.

Learning algorithms with guarantees. An important and interesting research direction is to design learning algorithms for producing envy-free classifiers with low loss that can be expressed as random mixtures of deterministic classifiers. One approach would be to apply the techniques of Cotter et al. [28], where the constrained optimization problem is formulated as a two-player game, where the players choose primal and dual variables to minimize/maximize the Lagrangian, respectively. They show that if the primal player minimizes regret and the dual player minimizes the swap-regret, then they will converge to a mixed equilibria such that the corresponding randomized mixture of primal variables is approximately optimal and feasible. Since the resulting classifier is already expressed as a randomized mixture of some base classifiers, this is a promising approach for learning envy-free mixtures of deterministic classifiers.

Bibliography

[1] E. Allwein, R. Schapire, and Y. Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. In *Journal of Machine Learning Research*, 2000.

[2] Noga Alon, Nicolo Cesa-Bianchi, Claudio Gentile, Shie Mannor, Yishay Mansour, and Ohad Shamir. Nonstochastic multi-armed bandits with graph-structured feedback. *SIAM Journal on Computing*, 46(6):1785–1826, 2017.

[3] Peter Auer, Nicol'o Cesa-Bianchi, Yoav Freund, and Robert Shapire. The nonstochastic multi-armed bandit problem. In *SIAM Journal on Computing*, 2003.

[4] Baruch Awerbuch and Robert Kleinberg. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences*, 74(1):97–114, 2008.

[5] K. Balasubramanian, P. Donmez, and G. Lebanon. Unsupervised supervised learning ii: Margin-based classification without labels. In *AISTATS*, pages 137–145, 2011.

[6] K. Balasubramanian, P. Donmez, and G. Lebanon. Unsupervised supervised learning ii: Margin-based classification without labels. In *Journal of Machine Learning Research*, volume 12, pages 3119–3145, 2011.

[7] M-F. Balcan and A. Blum. A discriminative model for semi-supervised learning. In *Journal of the ACM*, 2010.

[8] M-F. Balcan and R. Urner. Active learning. In *Survey in the Encyclopedia of Algorithms*, 2015.

[9] M-F. Balcan, A. Blum, and K. Yang. Co-training and expansion: Towards bridging theory and practice. In *NIPS*, 2004.

[10] M-F. Balcan, A. Beygelzimer, and J. Lanford. Agnostic active learing. In *ICML*, 2006.

[11] M.-F. Balcan, F. Constantin, S. Iwata, and L. Wang. Learning valuation functions. In *25thProceedings of the Conference on Learning Theory (COLT)*, pages 4.1–4.24, 2012.

[12] M.-F. Balcan, A. Blum, and Y. Mansour. Exploiting ontology structures and unlabeled data for learning. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pages 1112–1120, 2013.

[13] M-F. Balcan, T. Dick, and Y. Mansour. Label efficient learning by exploiting multi-class output codes. In *AAAI*, 2017.

[14] Maria-Florina Balcan, Vaishnavh Nagarajan, Ellen Vitercik, and Colin White. Learning-theoretic foundations of algorithm configuration for combinatorial partitioning problems. *Proceedings of the Conference on Learning Theory (COLT)*, 2017.

[15] Maria-Florina Balcan, Travis Dick, and Ellen Vitercik. Dispersion for data-driven algorithm design, online learning, and private optimization. In *FOCS*, 2018.

[16] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Differentially private empirical risk minimization: Efficient algorithms and tight error bounds. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, 2014.

[17] A. Beygelzimer, J. Langford, and P. Ravikumar. Solving multiclass learning problems via error-correcting output codes. *ALT*, 2009.

[18] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, 1998.

[19] S. J. Brams and A. D. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996.

[20] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[21] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

[22] Nicoló Cesa-Bianchi, Alex Conconi, and Claudio Gentile. On the generalization ability of on-line learning algorithms. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 359–366, 2002.

[23] U. Chajewska, D. Koller, and D. Ormoneit. Learning an agent's utility function by observing behavior. In *18thProceedings of the International Conference on Machine Learning (ICML)*, pages 35–42, 2001.

[24] O. Chapelle, B. Schlkopf, and A. Zien. *Semi-Supervised Learning*. The MIT Press, 1st edition, 2010. ISBN 0262514125, 9780262514125.

[25] Moses Charikar and Anthony Wirth. Maximizing quadratic programs: extending Grothendieck's inequality. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, 2004.

[26] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

[27] Vincent Cohen-Addad and Varun Kanade. Online Optimization of Smoothed Piecewise Constant Functions. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[28] Andrew Cotter, Heinrich Jiang, Serena Wang, Taman Narayan, Maya Gupta, Seungil You, and Karthik Sridharan. Optimization with non-differentiable constraints with applications to fairness, recall, churn, and other goals, 2018.

[29] A. Daniely, M. Schapira, and G. Shahaf. Multiclass learning approaches: A theoretical comparison with implications. In *NIPS*, 2012.

[30] S. Dasgupta. Two faces of active learning. In *Theoretical Computer Science*, 2011.

[31] A. Datta, M. C. Tschantz, and A. Datta. Automated experiments on ad privacy settings: A tale of opacity, choice, and discrimination. In *15th*, pages 92–112, 2015.

[32] T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, pages 263–286, 1995.

[33] M. Donini, L. Oneto, S. Ben-David, J. Shawe-Taylor, and M. Pontil. Empirical Risk Minimization under Fairness Constraints. arXiv:1802.08626, 2018.

[34] P. Donmez, G. Lebanon, and K. Balasubramanian. Unsupervised supervised learning i: Estimating classification and regression errors without labels. In *Journal of Machine Learning Research*, volume 11, pages 1323–1351, 2010.

[35] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. S. Zemel. Fairness through awareness. In *3rdProceedings of the ACM Conference on Innovations in Theoretical Computer Science (ITCS)*, pages 214–226, 2012.

[36] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Theory of Cryptography Conference (TCC)*, pages 265–284. Springer, 2006.

[37] Uriel Feige and Michael Langberg. The RPR^2 rounding technique for semidefinite programs. *Journal of Algorithms*, 60(1):1–23, 2006.

[38] Zhe Feng, Chara Podimata, and Vasilis Syrgkanis. Learning to bid without knowing your value. *Proceedings of the ACM Conference on Economics and Computation (EC)*, 2018.

[39] D. Foley. Resource allocation and the public sector. *Yale Economics Essays*, 7:45–98, 1967.

[40] R. Freedman, J. Schaich Borg, W. Sinnott-Armstrong, J. P. Dickerson, and V. Conitzer. Adapting a kidney exchange algorithm to align with human values. In *32ndProceedings of the AAAI Conference on Artificial Intelligence*, pages 1636–1645, 2018.

[41] Y. Gal, M. Mash, A. D. Procaccia, and Y. Zick. Which is the fairest (rent division) of them all? *Journal of the ACM*, 64(6): article 39, 2017.

[42] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[43] Rishi Gupta and Tim Roughgarden. A PAC approach to application-specific algorithm selection. *SIAM Journal on Computing*, 46(3):992–1017, 2017.

[44] S. Hanneke. Theory of active learning. *Foundations and Trends in Machine Learning*, 7(2–3), 2014.

[45] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. In *30thProceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 3315–3323, 2016.

[46] Elad Hazan and Zohar Karnin. Volumetric spanners: an efficient exploration basis for learning. *The Journal of Machine Learning Research*, 17(1):4062–4095, 2016.

[47] Ú. Hébert-Johnson, M. P. Kim, O. Reingold, and G. N. Rothblum. Calibration for the (computationally-identifiable) masses. In *35thProceedings of the International Conference on Machine Learning (ICML)*, 2018. Forthcoming.

[48] M. Joseph, M. Kearns, J. Morgenstern, and A. Roth. Fairness in learning: Classic and contextual bandits. In *30thProceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 325–333, 2016.

[49] M. Kearns, S. Neel, A. Roth, and S. Wu. Computing parametric ranking models via rank-breaking. In *35thProceedings of the International Conference on Machine Learning (ICML)*, 2018.

[50] Robert Kleinberg. Nearly tight bounds for the continuum-armed bandit problem. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 2004.

[51] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*. ACM, 2008.

[52] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandits in metric spaces. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, 2008.

[53] Matt Kusner, Jacob Gardner, Roman Garnett, and Kilian Weinberger. Differentially private Bayesian optimization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 918–927, 2015.

[54] J. Langford and A. Beygelzimer. Sensitive error correcting output codes. *COLT*, 2005.

[55] B. T. Luong, S. Ruggieri, and F. Turini. k-NN as an implementation of situation testing for discrimination discovery and prevention. In *17thProceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 502–510, 2011.

[56] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 94–103, 2007.

[57] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling. Never-ending learning. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*, 2015.

[58] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. MIT press, 2012.

[59] H. Moulin. *Fair Division and Collective Welfare*. MIT Press, 2003.

[60] T. D. Nielsen and F. V. Jensen. Learning a decision maker's utility function from (possibly) inconsistent behavior. *Artificial Intelligence*, 160(1–2):53–78, 2004.

[61] R. Noothigattu, S. S. Gaikwad, E. Awad, S. Dsouza, I. Rahwan, P. Ravikumar, and A. D. Procaccia. A voting-based system for ethical decision making. In *32ndProceedings of the AAAI Conference on Artificial Intelligence*, pages 1587–1594, 2018.

[62] M. Palatucci, D. Pomerleau, G. Hinton, and T. Mitchell. Zero-shot learning with semantic output codes. In *NIPS*, 2009.

[63] A. D. Procaccia. Cake cutting: Not just child's play. *Communications of the ACM*, 56(7):78–87, 2013.

[64] Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning: Stochastic, constrained, and smoothed adversaries. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*. 2011.

[65] J. M. Robertson and W. A. Webb. *Cake Cutting Algorithms: Be Fair If You Can*. A. K. Peters, 1998.

[66] G. N. Rothblum and G. Yona. Probably approximately metric-fair learning. arXiv:1803.03242, 2018.

[67] J. Steinhardt and P. Liang. Unsupervised risk estimation with only structural assumptions. 2016. (Preprint).

[68] F. E. Su. Rental harmony: Sperner's lemma in fair division. *American Mathematical Monthly*, 106(10):930–942, 1999.

[69] L. Sweeney. Discrimination in online ad delivery. *Communications of the ACM*, 56(5):44–54, 2013.

[70] S. Thrun. *Explanation-Based Neural Network Learning: A Lifelong Learning Approach*. Kluwer Academic Publishers, Boston, MA, 1996.

[71] S. Thrun and T. Mitchell. Learning one more thing. In *Proc. 14th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1217–1225, 1995.

[72] Sebastian Thrun and Tom M. Mitchell. Lifelong robot learning. *Robotics and Autonomous Systems*, 15(1-2):25–46, 1995.

[73] H. Varian. Equity, envy and efficiency. *Journal of Economic Theory*, 9:63–91, 1974.

[74] Jonathan Weed, Vianney Perchet, and Philippe Rigollet. Online learning in repeated auctions. In *Proceedings of the Conference on Learning Theory (COLT)*, pages 1562–1583, 2016.

[75] B. Woodworth, S. Gunasekar, M. I. Ohannessian, and N. Srebro. Learning non-discriminatory predictors. In *30thProceedings of the Conference on Learning Theory (COLT)*, pages 1920–1953, 2017.

[76] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, K. P. Gummadi, and A. Weller. From parity to preference-based notions of fairness in classification. In *31stProceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 228–238, 2017.

[77] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. Learning fair representations. In *30thProceedings of the International Conference on Machine Learning (ICML)*, pages 325–333, 2013.

[78] Uri Zwick. Outward rotations: a tool for rounding solutions of semidefinite programming relaxations, with applications to max cut and other problems. In *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, 1999.