Lexicographically Fair Learning: Algorithms and Generalization

Emily Diana
University of Pennsylvania, Philadelphia, PA, USA

Wesley Gill
University of Pennsylvania, Philadelphia, PA, USA

Ira Globus-Harris
University of Pennsylvania, Philadelphia, PA, USA

Michael Kearns
University of Pennsylvania, Philadelphia, PA, USA

Aaron Roth
University of Pennsylvania, Philadelphia, PA, USA

Saeed Sharifi-Malvajerdi
University of Pennsylvania, Philadelphia, PA, USA

Abstract

We extend the notion of minimax fairness in supervised learning problems to its natural conclusion: *lexicographic* minimax fairness (or *lexifairness* for short). Informally, given a collection of demographic groups of interest, minimax fairness asks that the error of the group with the *highest* error be minimized. Lexifairness goes further and asks that amongst all minimax fair solutions, the error of the group with the second highest error should be minimized, and amongst all of *those* solutions, the error of the group with the third highest error should be minimized, and so on. Despite its naturalness, correctly defining lexifairness is considerably more subtle than minimax fairness, because of inherent sensitivity to approximation error. We give a notion of approximate lexifairness that avoids this issue, and then derive oracle-efficient algorithms for finding approximately lexifair solutions in a very general setting. When the underlying empirical risk minimization problem absent fairness constraints is convex (as it is, for example, with linear and logistic regression), our algorithms are provably efficient even in the worst case. Finally, we show generalization bounds – approximate lexifairness on the training sample implies approximate lexifairness on the true distribution with high probability. Our ability to prove generalization bounds depends on our choosing definitions that avoid the instability of naive definitions.

2012 ACM Subject Classification Computing methodologies → Machine learning

Keywords and phrases Fair Learning, Lexicographic Fairness, Online Learning, Game Theory

Digital Object Identifier 10.4230/LIPIcs.FORC.2021.6

Related Version *Full Version*: `https://arxiv.org/abs/2102.08454`

Funding Supported in part by the Warren Center for Network and Data Sciences, NSF grant CCF-1763307 and the Simons Collaboration on the Theory of Algorithmic Fairness.

1 Introduction

Most notions of statistical group fairness ask that a model approximately equalize some error statistic across demographic groups. Often this is motivated as a tradeoff: the goal is to lower the error of the most disadvantaged group, and if doing so requires increasing the error on some more advantaged group, so be it – this is a cost that we are willing to pay in the name of equity. But solutions which equalize group errors do *not* in general mediate a clean tradeoff in which losses in accuracy on more advantaged groups result in increases in

accuracy on less advantaged groups: instead, generically (i.e. except in the very special case in which the Bayes optimal error is identical for all groups), a constraint of equalizing group error rates may require *artificially increasing* the error on at least one group, without any corresponding benefit to any other group.

A partial answer to this criticism of standard notions of group fairness is the classical notion of *minimax fairness*, recently studied by [23, 9] in the context of supervised learning. Minimax fairness asks for a model which minimizes the error of the group *most disadvantaged* by the model – i.e. the group with maximum group error. In doing so, it realizes the promise of equal error solutions in that it trades off higher error on populations more advantaged by the model for lower error on populations less advantaged by the model when this is possible – but without artificially increasing the error of any group when doing so. Indeed, it is not hard to see that a minimax model necessarily *weakly Pareto dominates* an equal error rate model, in the sense that group errors are only lower in the minimax solution *simultaneously for all groups*.

This narrative is most sensible if there are only two demographic groups of interest. If there are more than two groups, there may be many different minimax optimal models that have very different error profiles for groups other than the max error group. How should we choose amongst these? Prior work [9] has broken ties by optimizing for overall classification accuracy. But why should we entirely give up on the goal of optimizing for the most disadvantaged, partially enunciated in the motivation of minimax fairness, once we have fixed the error of only one of many groups?

In this paper we propose the natural continuation of this idea, which we call *lexicographic minimax fairness*. Informally speaking, this notion recurses on the idea that we wish to minimize the cost of the least well off. A model that satisfies lexicographic fairness, which we call a *lexifair* model, will minimize the maximum error γ_1 on any group, amongst all possible models (i.e. a lexifair model is a also a minimax model). Further, amongst the set of all minimax models, a lexifair model must minimize the error of the group with the second highest error γ_2. Amongst all of these models, it further minimizes the error of the group with the third highest error γ_3, and so on.[1]

1.1 Our Contributions

Our first contribution is a definition of (approximate) lexicographic minimax fairness. Correctly defining an actionable notion of lexicographic minimax fairness is surprisingly subtle. For standard computational and statistical reasons, it will not be possible to exactly match the distributional lexicographically optimal error rates $\gamma_1, \gamma_2, \gamma_3$, etc. But as we will observe, these lexicographically optimal error rates can be arbitrarily unstable, in the sense that amongst the set of models that have minimax error larger than γ_1 by even an arbitrarily small margin, the value of the optimal lexifair error on the third highest error group γ_3' can be arbitrarily larger than γ_3 (See our example in Section 2.1.1). An implication of this is that the vectors of errors γ, γ' representing exact lexifair solutions in and out of sample can be entirely incomparable and arbitrarily different from one another. Hence we need a definition of approximate lexifairness that accounts for this instability, and allows for sensible statements about approximation and generalization.

Another challenge arises in the interaction between our definitions and our (desired) algorithms. A constraint on the *highest* error amongst all groups, which arises in defining minimax error, is convex, and hence amenable to algorithmic optimization. However, naive

[1] It is easy to see that there are cases in which a lexifair model may have arbitrarily smaller errors than a minimax model on all but the worst-off group.

specifications of lexifairness involve constraints on the second highest group errors, the third highest group errors, and more generally kth highest errors. These are non-convex constraints when taken in isolation. However, as it turns out, a constraint on the second highest error becomes convex when we restrict attention to minimax optimal classifiers, and more generally, a constraint on the kth highest error becomes convex once the values of the lower order group errors are constrained to their lexifair values. We show this by giving a clearly convex variant of our lexifair definition, specified by exponentially many *linear constraints*, which replace constraints on the k'th highest error groups with constraints on the *sums* of all k-tuples of group errors. We then show that our definition of "convex lexifairness" is equivalent to our original notion of lexifairness, at least in the exact case (absent approximation). We give our formal definitions in Section 2.1.2.

With our notion of approximate lexifairness in hand and our convexified constraints, we give *oracle-efficient* algorithms for finding approximate lexifair models in both the regression and classification case. This means that our algorithms are efficient reductions to the problem of unconstrained (that is, standard non-fair) learning over the same model class. Despite the worst-case intractability of most natural learning problems even absent fairness considerations, a desirable feature of oracle-efficient algorithms is that they can be implemented using any of the common and practical heursitics for non-fair learning, often with good empirical success [20, 30, 16, 1].

Our algorithms are based on solving the corresponding constrained optimization problem by recasting it as a (Lagrangian) minmax optimization problem, and using no-regret dynamics. Because our "convexified" lexifairness constraints are exponentially numerous, the "constraint player" in our formulation has exponentially many strategies – but as we show, we can efficiently optimize over her strategy space using an efficient separation oracle. Hence the constraint player can always play according to a "best response" strategy in our simulated dynamics. When our base model class is continuous and our loss function convex (as it is with e.g. linear regression), then the "learner" in our dynamics can play gradient descent over parameter space. In this case, our oracle efficient-algorithms are in fact fully polynomial time algorithms because our reduction to weighted learning problems involves only *non-negative* weights, which preserves convexity. In the classification case, when our loss function is *non-convex*, we can convexify it by considering the set of all probability distributions over base models. Here the parameters we optimize over become the weights of the probability distribution, and our loss function (i.e. the expected loss over the choice of a random model) becomes linear in our (enormous) parameter space. In this case, we are effectively solving a linear program that has both exponentially many variables and exponentially many constraints – but we are nevertheless able to do so in an oracle-efficient manner by making appropriate use of the Follow the Perturbed Leader algorithm [18] for no-regret learning.

Finally, we prove a generalization theorem, showing that if we have a dataset S (sampled i.i.d. from an underlying distribution) that has sufficiently many samples from each group, and if we have a model that is approximately lexifair for S, then the model is also approximately lexifair on the underlying distribution. This is significantly more involved than just a standard uniform convergence argument – which would simply state that our in and out of sample errors on each group are close to one another – because approximate lexifairness additionally depends on the precise *relationship* between these group errors. Nevertheless, we show that uniform convergence is a sufficient condition to guarantee that in-sample lexifairness bounds correspond to out of sample lexifairness bounds.

1.2 Related Work

There are many notions of group or statistical fairness that are studied in the fair machine learning literature, which are generally concerned with *equalizing* various measures of error across protected groups; see e.g. [4, 24] for surveys of many such metrics.

Minimax solutions are a classical approach to fairness that have been used in many contexts including scheduling, fair division, and clustering (see e.g. [14, 3, 29, 5, 6]). A number of these works employ techniques for solving two-player zero-sum games as part of their algorithmic solution [6, 5]. This is the same general algorithmic framework that we use. More recently, minimax group error has been proposed as a fairness solution concept for classification problems in machine learning [23, 9, 22]. These works generally do not specify how to choose between multiple minimax solutions, with the exception of [9], which gives algorithms for choosing the solution with smallest overall classification error subject to the minimax constraint.

Lexicographic minimax fairness has been studied in the fair division literature for tasks such as quota allocation in mobile networks, load balancing, and network design [10, 7, 25, 33, 32, 28, 2, 27, 26]. As far as we know, we are the first to study lexicographic fairness in a learning context in which the quantities of interest must be *estimated*, and hence the first to identify the sensitivity issues that arise when defining *approximate* notions of lexicographic fairness.

An alternative approach to learning one classifier for all groups is to learn *decoupled classifiers* [11, 31], i.e. a separate classifier for each group. The decoupling of error rates across all groups eliminates tradeoffs between groups, and hence results in classifiers that are lexicographically fair (within the class of decoupled classifiers). But there are at least three important reasons one might want to learn a single classifier (the approach we take) rather than a separate classifier for each group. The first is that learning separate classifiers for each group requires that the groups be *disjoint*, which is not needed in our approach. For example, we could divide the population into groups according to race, gender, and age – despite the fact that individuals will fall into multiple groups simultaneously. In other words, our algorithms can be used to obtain *subgroup* or *intersectional* fairness [19, 20, 15, 21, 17, 13]. Second, learning separate classifiers for each group requires that protected group membership be used explicitly at classification time, which can be undesirable or illegal in important applications. Finally, learning a single classifier allows for the possibility of transfer learning, whereby a small sample from some group can be partially made up for by larger quantities of data from other (nevertheless related) groups.

2 Model and Definitions

Let $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ be an arbitrary data domain. Each data point in our setting is a pair $z = (x, y)$ where $x \in \mathcal{X}$ is the feature vector and $y \in \mathcal{Y}$ is the response variable (i.e. the label). Let $\mathcal{X}$ consist of points belonging to K (not necessarily disjoint) groups $\mathcal{G}_1, \ldots, \mathcal{G}_K$, so we can write $\mathcal{X} = \cup_{k=1}^{K} \mathcal{G}_k$. We write $\mathcal{P}$ to denote an arbitrary distribution over $\mathcal{Z}$, and $\mathcal{P}_k$ to denote the marginal distribution induced by $\mathcal{P}$ on the kth group $\mathcal{G}_k \times \mathcal{Y}$. Let $S = \{z_i\}_{i=1}^{n}$ be a data set of size n, which for the purposes of proving generalization bounds, we will take to consist of n data points drawn i.i.d. from $\mathcal{P}$. Denote the points in S that are contained in $\mathcal{G}_k$ by $G_k \times \mathcal{Y}$, so we can write $S = \cup_{k=1}^{K} G_k$.

Let $\mathcal{H} \subseteq \{h : \mathcal{X} \to \mathcal{Y}\}$ be the model class of interest, and let $L : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$ be a loss function that takes a data point z and a model h as inputs, and outputs the loss of h on z. For instance, in the case of classification and zero-one loss, we have $L(h, z) = \mathbb{1}[h(x) \neq y]$. We will abuse notation and write $L_z(\cdot)$ for $L(\cdot, z)$ for any data point z. Throughout the paper, for any distribution $\mathcal{P}$, we write the expected loss of a model h over $\mathcal{P}$ as:

$$L_{\mathcal{P}}(h) \triangleq L(h, \mathcal{P}) \triangleq \mathbb{E}_{z \sim \mathcal{P}}[L_z(h)].$$

We slightly abuse notation and write $L_S(h)$ to denote the empirical loss on a dataset S. Here and throughout the paper when S plays the role of a distribution, we interpret that as the *uniform distribution* over the points in S, and accordingly, $z \sim S$ as a point sampled *uniformly at random* from S.

Until Section 7, we will work exclusively with sample quantities, and so for simplicity of notation, let us define $L_k(h) \triangleq L_{G_k}(h)$ to denote the *sample* loss of a model h on the k'th group. When necessary, we will write $L_k(h, \mathcal{P})$ to denote $L_{\mathcal{P}_k}(h)$, the corresponding *distributional* loss of h on the k'th group. For any model h and any data set $S = \cup_k \{G_k\}$, let $\bar{h}_S$ be the ordering induced on the groups $\{G_k\}_{k=1}^K$ by the loss of h, breaking ties arbitrarily. In other words, $\bar{h}_S : [K] \to [K]$ is any bijection such that the following condition holds: $L_{\bar{h}_S(1)}(h) \geq L_{\bar{h}_S(2)}(h) \geq \ldots \geq L_{\bar{h}_S(K)}(h)$. The corresponding distributional ordering of the groups by any model h is defined similarly: for any model h and any distribution $\mathcal{P}$ over $\mathcal{Z}$, let $\bar{h}_{\mathcal{P}} : [K] \to [K]$ be the ordering induced on the groups $\{\mathcal{G}_k\}_{k=1}^K$ by the expected loss of h, breaking ties arbitrarily. In other words, $\bar{h}_{\mathcal{P}}$ is any bijection such that the following condition holds: $L_{\bar{h}_{\mathcal{P}}(1)}(h, \mathcal{P}) \geq L_{\bar{h}_{\mathcal{P}}(2)}(h, \mathcal{P}) \geq \ldots \geq L_{\bar{h}_{\mathcal{P}}(K)}(h, \mathcal{P})$. When the distribution (data set) is clear from context, we elide the dependence on the distribution (data set) and simply write $\bar{h}$ for $\bar{h}_{\mathcal{P}}$ ($\bar{h}_S$).

Our definition of lexifairness will be given recursively. At the base level, we define $\mathcal{H}_{(0)} = \mathcal{H}$ to be the set of all models in our class. Then recursively for all $1 \leq j \leq K$, we define:

$$\gamma_j \triangleq \min_{h \in \mathcal{H}_{(j-1)}} L_{\bar{h}(j)}(h), \quad \mathcal{H}_{(j)} \triangleq \left\{ h \in \mathcal{H}_{(j-1)} : L_{\bar{h}(j)}(h) = \gamma_j \right\}.$$

In words, γ_j is the smallest error that any model in $\mathcal{H}_{(j-1)}$ obtains on the group that has the jth highest error, and $\mathcal{H}_{(j)}$ is the set of *all* models in $\mathcal{H}_{(j-1)}$ that attain this minimum – i.e. that have jth highest error equal to γ_j. Thus, γ_1 is the minimax error – i.e. the highest group error for the model that is chosen to *minimize* the maximum group error. Similarly, γ_2 is the error of the second highest group for all minimax optimal models that further minimize the error of the second highest group, and so on. With this notation in hand, we can define exact lexifairness as follows:

▶ **Definition 1** (Exact Lexicographic Fairness). *Let $1 \leq \ell \leq K$. We say a model $h \in \mathcal{H}$ satisfies level-ℓ (exact) lexicographic fairness (lexifairness) if for all $j \leq \ell$, $L_{\bar{h}(j)}(h) \leq \gamma_j$.*

Minimax fairness corresponds to level-1 lexifairness. This is a definition of *exact* lexifairness, in that it permits no approximation to the error rates – i.e. we require $L_{\bar{h}(j)}(h) \leq \gamma_j$ for all j, and hence $L_{\bar{h}(j)}(h) = \gamma_j$ for all j. For a variety of reasons, we will need definitions that tolerate approximation. For example, because we inevitably have to train on a fixed dataset, but want our guarantees to generalize to new datasets drawn from the same distribution, we will need to accommodate statistical approximation. The optimization techniques we will bring to bear will also only be able to *approximate* lexifairness, even in sample. But it turns out that defining a sensible approximate notion of lexifairness is more subtle than it first appears.

2.1 Approximate Lexifairness: Stability and Convexity

We begin with the "obvious" but ultimately flawed definition of approximate lexifairness (Definition 2), and then explain why it is lacking in stability. This will lead us to the definitions we finally adopt: Definition 3 and its *convexified* version (Definition 5), which we show is equivalent (Claim 7), and for which we can develop efficient algorithms.

2.1.1 The Challenge of Stability

The most natural seeming definition of approximate lexifairness begins with our notion of exact lexifairness (Definition 1), and adds slack to all of the inequalities contained within. In other words, we attempt to find a model that has sorted group errors $\gamma'_1, \gamma'_2, \ldots, \gamma'_K$ that pointwise approximate the optimal lexifair vector of sorted group errors $\gamma_1, \ldots, \gamma_K$.

▶ **Definition 2** (A Flawed Definition)**.** *Let $1 \leq \ell \leq K$ and $\alpha \geq 0$. We say a model $h \in \mathcal{H}$ satisfies (ℓ, α)-lexicographic fairness if for all $j \leq \ell$, $L_{\bar{h}(j)}(h) \leq \gamma_j + \alpha$.*

To see the problem with the above definition, consider a setting with three groups, and a model class $\mathcal{H}$ that contains all distributions (or randomized classifiers) over two pure classifiers $\{h_1, h_2\}$. Imagine that h_1 induces the (unsorted) vector of group error rates $\langle 0.5, 0.5, 0\rangle$, and h_2 induces the (unsorted) vector of group error rates $\langle 0.5 + 2\alpha, 0, 0.5\rangle$, for some arbitrarily small $\alpha > 0$. Note that it is easy to construct distributions over labeled instances with exactly these group error vectors by simply arranging each classifier to disagree with the labels on the specified fraction of a group. So, for simplicity we abstract away the data and directly discuss the error vectors.

The minimax group error for this model class is $\gamma_1 = 0.5$, and is achieved only by h_1 which has error 0.5 on the first and second groups. Since the largest group error of h_2 is also on the first group with value $0.5 + 2\alpha > 0.5$, any distribution over $\{h_1, h_2\}$ that places a non-zero probability on h_2 will therefore violate the (exact) minimax constraint. This in turn implies that $\mathcal{H}_{(1)} = \{h_1\}$. Therefore, the only exact lexifair model is h_1 and thus $\gamma_1 = 0.5$, $\gamma_2 = 0.5$, $\gamma_3 = 0$.

However, imagine that because of estimation error (as is inevitable if we are learning based on a finite sample) or optimization error (since we generally don't have access to exact optimization oracles in learning settings), we slightly misestimate the minimax group error γ_1 to be $\gamma'_1 = 0.5 + \alpha$. If we now optimize, allowing the largest group error to be as much as $\gamma'_1 = 0.5 + \alpha$, we may now find randomized classifiers which put weight as large as 0.5 on h_2. The uniform distribution over $\{h_1, h_2\}$ induces the unsorted vector of group errors $\langle 0.5 + \alpha, 0.25, 0.25\rangle$. The induced error on the second group (which is now also the group with second largest error) of 0.25 is considerably *smaller* than $\gamma_2 = 0.5$. So far this appears to be all right, since $\gamma'_2 < \gamma_2$. But if we now attempt to optimize the error of the third highest error γ'_3, *subject to the constraint* that the largest group error is (close to) γ'_1 and the second largest group error is (close to) γ'_2, we now find that we are forced to settle for third highest group error $\gamma'_3 \approx 0.25$, which is considerably *larger* than the value of the third highest group's error of $\gamma_3 = 0$ in the exact lexifair solution.

This example highlights a fundamental *instability* of our first (flawed) attempt at defining approximate lexifairness: even arbitrarily small estimation (or optimization) error introduced to the minimax error rate γ_1 can result in large, non-monotonic effects for later group errors – enforcing even a valid *upper bound* on γ_1 can cause γ_3 to increase substantially, and these effects compound even further if we have more than three groups.

2.1.2 A Stable and Convex Definition

With the proceeding example of the instability inherent in our (flawed) Definition 2, we now give the definition of approximate lexifairness that we begin with:

▶ **Definition 3** (Approximate Lexicographic Fairness). *Fix a distribution $\mathcal{P}$. Let $1 \leq \ell \leq K$ and $\alpha \geq 0$. For any sequence of mappings $\vec{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_\ell)$ where $\epsilon_j \in \mathbb{R}^{\mathcal{H}}$, define $\mathcal{H}^{\vec{\epsilon}}_{(0)}(\mathcal{P}) \triangleq \mathcal{H}$, and recursively for all $1 \leq j \leq \ell$ define:*

$$\mathcal{H}^{\vec{\epsilon}}_{(j)}(\mathcal{P}) \triangleq \left\{ h \in \mathcal{H}^{\vec{\epsilon}}_{(j-1)}(\mathcal{P}) : L_{\bar{h}(j)}(h, \mathcal{P}) \leq \min_{g \in \mathcal{H}^{\vec{\epsilon}}_{(j-1)}(\mathcal{P})} L_{\bar{g}(j)}(g, \mathcal{P}) + \epsilon_j(h) \right\}$$

and let $\|\vec{\epsilon}\|_\infty = \max_{1 \leq j \leq \ell} \max_{h \in \mathcal{H}} \epsilon_j(h)$. We say a model $h \in \mathcal{H}$ satisfies (ℓ, α)-lexicographic fairness ("lexifairness") with respect to $\mathcal{P}$ if there exists $\vec{\epsilon}$ with $\|\vec{\epsilon}\|_\infty \leq \alpha$ such that for all $j \leq \ell$:

$$L_{\bar{h}(j)}(h, \mathcal{P}) \leq \min_{g \in \mathcal{H}^{\vec{\epsilon}}_{(j-1)}(\mathcal{P})} L_{\bar{g}(j)}(g, \mathcal{P}) + \epsilon_j(h) + \alpha.$$

When we prove bounds on empirical lexifairness, we simply take the distribution to be the uniform distribution over the data set S. When the distribution is clear from context, we will write $\mathcal{H}^{\vec{\epsilon}}_{(j)}$ and elide the dependence on the distribution.

Note that there are two distinctions between Definition 3 and Definition 2. First, the recursively defined sets $\mathcal{H}^{\vec{\epsilon}}_{(j)}$ now incorporate some $\epsilon_j(\cdot)$ slack in their parameterization which will help capture statistical (or optimization) error. Second (and crucially), we now call a solution (ℓ, α)-approximately lexifair if it satisfies our requirements for *some* sequence of relaxations $\vec{\epsilon}$ that is component-wise less than α for all models h. It is this second point that avoids the instability and non-monotonicity that arises from Definition 2. We observe that Definition 3 is a strict weakening of Definition 2:

▶ **Claim 4.** *Definition 3 is a relaxation of Definition 2: if a model satisfies (ℓ, α)-lexicographic fairness according to Definition 2, then it also satisfies (ℓ, α)-lexicographic fairness according to Definition 3.*

Proof. If a model satisfies (ℓ, α)-lexicographic fairness according to Definition 2, then by taking $\vec{\epsilon} = \vec{0}$, it also meets the conditions of Definition 3. ◀

We now face another definitional challenge. A priori, Definition 3 appears to be highly non-convex, because it constrains the second highest group error, the the third highest group error, etc.[2] This is in contrast to standard equal-error notions of fairness, or minimax fairness (which constrains only the highest group error) that *are* convex in the sense that a distribution over fair models remains fair. Without convexity of this sort, the algorithmic problem of finding a fair model becomes much more challenging. But in fact (at least for $\alpha = 0$), Definition 3 *does* give a convex constraint. To see this, we first introduce an alternative notion of *convex lexifairness*, and then show that it actually represents the exact same constraint as lexifairness when the approximation parameter $\alpha = 0$.

[2] E.g., if we have two groups and two models which induce group errors $(0.5, 0)$ and $(0, 0.5)$ respectively, both solutions have a second-highest error of 0 – but convex combinations have a second highest error strictly greater than 0. So absent other structure, upper bounding the second highest group error of a model corresponds to a non-convex constraint. But note that in this two-group example, the non-convexity dissapears if we restrict attention to minimax optimal models. This is what we will take advantage of more generally.

▶ **Definition 5** (Convex Lexicographic Fairness). *Fix a distribution $\mathcal{P}$. Let $1 \leq \ell \leq K$ and $\alpha \geq 0$. For any sequence of mappings $\vec{\epsilon} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_\ell)$ where $\epsilon_j \in \mathbb{R}^{\mathcal{H}}$, define $\mathcal{F}^{\vec{\epsilon}}_{(0)}(\mathcal{P}) \triangleq \mathcal{H}$, and recursively for all $1 \leq j \leq \ell$ define:*

$$\mathcal{F}^{\vec{\epsilon}}_{(j)}(\mathcal{P}) \triangleq$$
$$\left\{ h \in \mathcal{F}^{\vec{\epsilon}}_{(j-1)}(\mathcal{P}) : \max_{\{i_1,\ldots,i_j\} \subseteq [K]} \sum_{r=1}^{j} L_{i_r}(h, \mathcal{P}) \leq \min_{g \in \mathcal{F}^{\vec{\epsilon}}_{(j-1)}(\mathcal{P})} \max_{\{i_1,\ldots,i_j\}} \sum_{r=1}^{j} L_{i_r}(g, \mathcal{P}) + \epsilon_j(h) \right\}$$

and let $\|\vec{\epsilon}\|_\infty = \max_{1 \leq j \leq \ell} \max_{h \in \mathcal{H}} \epsilon_j(h)$. We say a model $h \in \mathcal{H}$ satisfies (ℓ, α)-convex lexicographic fairness with respect to $\mathcal{P}$ if there exists $\vec{\epsilon}$ with $\|\vec{\epsilon}\|_\infty \leq \alpha$ such that for all $j \leq \ell$:

$$\max_{\{i_1,\ldots,i_j\} \subseteq [K]} \sum_{r=1}^{j} L_{i_r}(h, \mathcal{P}) \leq \min_{g \in \mathcal{F}^{\vec{\epsilon}}_{(j-1)}(\mathcal{P})} \max_{\{i_1,\ldots,i_j\} \subseteq [K]} \sum_{r=1}^{j} L_{i_r}(g, \mathcal{P}) + \epsilon_j(h) + \alpha.$$

When we prove bounds on empirical convex lexifairness, we simply take the distribution to be the uniform distribution over the data set S. When the distribution is clear from context, we will write $\mathcal{F}^{\vec{\epsilon}}_{(j)}$ and elide the dependence on the distribution.

Here, we have replaced constraints on the j'th highest group error with constraints on the *sum* of group errors over all $\approx K^j$ subsets of groups of size j. This has replaced a single constraint with many constraints, but each is convex, and hence the resulting set of constraints defined by $\mathcal{F}^{\vec{\epsilon}}_{(j)}$ is convex. We will formally prove this in the following claim.

▶ **Claim 6** (Convexity of $\mathcal{F}^{\vec{\epsilon}}_{(j)}$). *Let $L_z : \mathcal{H} \to \mathbb{R}_{\geq 0}$ be a convex loss function. If the initial model class $\mathcal{H}$ is convex, then for all j and all $\vec{\epsilon}$ such that the mappings $\epsilon_j \in \mathbb{R}^{\mathcal{H}}$ are concave, the set $\mathcal{F}^{\vec{\epsilon}}_{(j)}$ is convex.*

The proof can be found in the full version of the paper ([8]), and proceeds by straightforward induction. We note that while some classes of models naturally satisfy the convexity conditions of the above claim with respect to their corresponding parameters (e.g. linear and logistic regression), this claim will apply to arbitrary classification models with zero-one loss as well. In these settings, we will convexify the class of models by considering the set of all probability distributions over deterministic models. The loss of a distribution (i.e. a randomized model) is then defined as the *expected* loss, when the model is sampled from the corresponding distribution. Hence, by linearity of expectation, our loss functions will be convex (linear) in the parameters – i.e. the weights – of these distributions.

It turns out that our notion of *convex* lexifairness is identical to our notion of lexifairness (and so our original definition in fact specified a convex set of constraints), at least when the approximation parameter $\alpha = 0$. We prove this in the following claim:

▶ **Claim 7** (Relationship between $\mathcal{F}^{\vec{\epsilon}}_{(j)}$ and $\mathcal{H}^{\vec{\epsilon}}_{(j)}$ when $\vec{\epsilon} = \vec{0}$). *For all j, and $\vec{\epsilon} = \vec{0}$, we have $\mathcal{F}^{\vec{\epsilon}}_{(j)} = \mathcal{H}^{\vec{\epsilon}}_{(j)}$.*

The intuition for the claim is the following. The sets $\mathcal{H}_{(j)}$ in Definition 3 constrain the error of the group that has the j'th highest error. In contrast, the sets $\mathcal{F}_{(j)}$ from Definition 5 constrain the *sum* of the errors for all possible j-tuples of groups. Amongst all of these constraints, the binding one will be the constraint corresponding to the j groups that have the *largest* errors. But because (inductively) the errors of the top $j-1$ error groups have already been appropriately constrained in $\mathcal{F}_{(j-1)}$, this reduces to a constraint on the j'th highest error group, as desired. These constraints are numerous, but each is convex, and so the resulting set of constraints can be seen to be convex. See the full version of the paper ([8]) for the formal proof of Claim 7, which proceeds by induction.

We emphasize that despite the complexity of our final Definition 5, what we have shown is that it is in fact a relaxation of our initial, natural definition of exact lexifairness (Definition 1) – and in particular Definitions 1, 3, and 5 coincide exactly when $\alpha = 0$. We do not know the precise relationship between our definitions of approximate lexifairness and approximate convex lexifairness for $\alpha > 0$ – but because both are smooth relaxations of the same base definition, both should be viewed as capturing the same intuition as Definition 1 (exact lexifairness) when α is small.

3 Game Theory and No-Regret Learning Preliminaries

3.1 No-Regret Dynamics

In this subsection, we briefly review the seminal result of Freund and Schapire [12]: Under certain conditions, two-player zero-sum games can be (approximately) solved by having access to a no-regret online learning algorithm for one of the players.

Suppose in this subsection that S_1 and S_2 are two vector spaces over the field of real numbers. Consider a zero-sum game with two players: a player with strategies in S_1 (the minimization player) and another player with strategies in S_2 (the maximization player). Let $U : S_1 \times S_2 \to \mathbb{R}_{\geq 0}$ be the payoff function of this game. For every strategy $s_1 \in S_1$ of player one and every strategy $s_2 \in S_2$ of player two, the first player gets utility $-U(s_1, s_2)$ and the second player gets utility $U(s_1, s_2)$.

▶ **Definition 8** (Approximate Equilibrium). *A pair of strategies $(s_1, s_2) \in S_1 \times S_2$ is said to be a ν-approximate minimax equilibrium of the game if the following conditions hold:*

$$U(s_1, s_2) - \min_{s_1' \in S_1} U(s_1', s_2) \leq \nu, \quad \max_{s_2' \in S_2} U(s_1, s_2') - U(s_1, s_2) \leq \nu$$

In other words, (s_1, s_2) is a ν-approximate equilibrium of the game if neither player can gain more than ν by deviating from their strategies.

Freund and Schapire [12] proposed an efficient framework for approximately solving the game: In an iterative fashion, have one of the players play according to a no-regret learning algorithm, and let the second player (approximately) best respond to the play of the first player. The empirical average of each player's actions over a sufficiently long sequence of such play will form an approximate equilibrium of the game. The formal statement is given in the following theorem.

▶ **Theorem 9** (No-Regret Dynamics [12]). *Let S_1 and S_2 be convex, and suppose the utility function U is convex-concave: $U(\cdot, s_2) : S_1 \to \mathbb{R}_{\geq 0}$ is convex for all $s_2 \in S_2$, and $U(s_1, \cdot) : S_2 \to \mathbb{R}_{\geq 0}$ is concave for all $s_1 \in S_1$. Let $(s_1^1, s_1^2, \ldots, s_1^T)$ be the sequence of play for the first player, and let $(s_2^1, s_2^2, \ldots, s_2^T)$ be the sequence of play for the second player. Suppose for $\nu_1, \nu_2 \geq 0$, the regret of the players jointly satisfies*

$$\sum_{t=1}^{T} U(s_1^t, s_2^t) - \min_{s_1 \in S_1} \sum_{t=1}^{T} U(s_1, s_2^t) \leq \nu_1 T, \quad \max_{s_2 \in S_2} \sum_{t=1}^{T} U(s_1^t, s_2) - \sum_{t=1}^{T} U(s_1^t, s_2^t) \leq \nu_2 T$$

Let $\bar{s}_1 = \frac{1}{T}\sum_{t=1}^{T} s_1^t \in S_1$ and $\bar{s}_2 = \frac{1}{T}\sum_{t=1}^{T} s_2^t \in S_2$ be the empirical average play of the players. We have that the pair $(\bar{s}_1, \bar{s}_2)$ is a $(\nu_1 + \nu_2)$-approximate equilibrium of the game.

No regret online learning algorithms are algorithms that can guarantee the conditions of Theorem 9 against arbitrary adversaries. We will use two no-regret online learning algorithms: *Online Projected Gradient Descent*, which we will use in regression settings in which models

are represented by parameters in a Euclidean space, and *Follow the Perturbed Leader (FTPL)*, which we will use in binary classification settings ([8]). We will make use of these no-regret learning algorithms in our proposed algorithm for learning a lexifair model; full explanations and pseudocode are in Appendix C.

4 Finding Lexifair Models

In this section we focus on developing the tools required to prove the following (informally stated) theorem. Formal claims appear in Theorems 13 (regression) and 14 (classification).

▶ **Theorem 10** (Informal). *Suppose the model class $\mathcal{H}$ is convex and compact, and that the loss function $L_z : \mathcal{H} \to \mathbb{R}_{\geq 0}$ is convex for all data points $z \in \mathcal{Z}$. There exists an efficient algorithm that returns a model which is (ℓ, α)-convex lexicographic fair (according to Definition 5), for any given ℓ and α.*

We will propose algorithms for both classification and regression settings. The algorithms we propose proceed inductively to solve the minimax problems defined recursively by our convex lexifair definition. The first minimax problem is the one that minimizes the maximum group error rate: $\min_{h\in\mathcal{H}} \max_{k\in[K]} L_k(h)$. Let us denote the estimated value (computed by the first phase of our algorithm) for this minimax problem by η_1. The second minimax problem is minimizing the maximum sum of any two group error rates subject to the constraint that all group error rates are at most η_1: the estimated value for this minimax problem is called η_2. The rest of the minimax problems are defined in a similar inductive fashion: suppose at round $j \leq \ell$, we are given some estimates $(\eta_1, \ldots, \eta_{j-1})$ for the first $j-1$ minimax values. Now using these estimates, the new minimax problem for the sum of any j group error rates can be stated as follows.

$$\min_{\substack{h\in\mathcal{H}: \\ \forall r\leq j-1,\, \forall\{i_1,\ldots,i_r\}\subseteq[K] \\ L_{i_1}(h)+\ldots+L_{i_r}(h)\leq\eta_r}} \left\{ \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h) \right\}. \tag{1}$$

We can reformulate this problem by calling the objective $\max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h) := \eta_j$ and introducing a new set of constraints which require that any sum of j group error rates must be at most η_j. Note that this new formulation introduces a new variable, η_j, to the optimization problem. We therefore have that the optimization problem (1) is equivalent to

$$\min_{\substack{h\in\mathcal{H},\eta_j\in[0,j\cdot L_M]: \\ \forall r\leq j,\, \forall\{i_1,\ldots,i_r\}\subseteq[K] \\ L_{i_1}(h)+\ldots+L_{i_r}(h)\leq\eta_r}} \eta_j \triangleq \mathrm{OPT}_j\left(\eta_1, \ldots, \eta_{j-1}\right) \tag{2}$$

which is a constrained convex optimization problem given that the model class $\mathcal{H}$ and the loss function L are convex. Here $L_M = \max_{z,h} L_z(h)$ is an upper bound on the loss function which identifies the range of feasible values for η_j: $[0, j\cdot L_M]$. Recall that in this round, $(\eta_1, \ldots, \eta_{j-1})$ are given from the previous rounds, and η_j is a variable in the optimization problem. We denote the optimal value of the optimization problem (2) by $\mathrm{OPT}_j\left(\eta_1, \ldots, \eta_{j-1}\right)$.

4.1 Formulation as a Two-Player Zero-Sum Game

Optimization problem (2) is written as a constrained optimization problem, but we can express it equally well as an unconstrained minimax problem via Lagrangian duality. The corresponding Lagrangian can be written as:

$$\mathcal{L}_j\left((h,\eta_j),\lambda\right) = \eta_j + \sum_{r=1}^{j} \sum_{\{i_1,\ldots,i_r\}\subseteq[K]} \lambda_{\{i_1,i_2,\ldots,i_r\}} \cdot \left(L_{i_1}(h) + \ldots + L_{i_r}(h) - \eta_r\right) \tag{3}$$

where we introduce one dual variable λ for every inequality constraint in the optimization problem (2), and index the dual variables by their corresponding constraint. Therefore, there are $q_j = \sum_{r=1}^{j} \binom{K}{r}$ dual variables in this round. Solving optimization problem (2) is equivalent to solving the following minimax problem:

$$\min_{h\in\mathcal{H},\eta_j\in[0,j\cdot L_M]} \max_{\lambda\in\mathbb{R}^{q_j}_{\geq 0}} \mathcal{L}_j\left((h,\eta_j),\lambda\right) = \max_{\lambda\in\mathbb{R}^{q_j}_{\geq 0}} \min_{h\in\mathcal{H},\eta_j\in[0,j\cdot L_M]} \mathcal{L}_j\left((h,\eta_j),\lambda\right) \tag{4}$$

where the minimax theorem holds because 1) the range of the primal variables, i.e. $\mathcal{H}$ and $[0, j \cdot L_M]$, is convex and compact, the range for the dual variable ($\mathbb{R}^{q}_{\geq 0}$) is convex, and 2) $\mathcal{L}_j\left((h,\eta_j),\lambda\right)$ is convex in its primal variables (h,η_j) and concave in the dual variable λ. Therefore we focus on solving the minimax problem (4) which can be seen as solving a two-player zero-sum game with payoff function $\mathcal{L}_j\left((h,\eta_j),\lambda\right)$. Using the no-regret dynamics of [12] (see Section 3.1), we will have the primal player (or *Learner*) with strategies $(h,\eta_j) \in \mathcal{H}\times[0, j\cdot L_M]$ play a no-regret learning algorithm and let the dual player (or *Auditor*) with strategies $\lambda \in \Lambda_j = \{\lambda \in \mathbb{R}^{q_j}_{\geq 0} : \|\lambda\|_1 \leq B\}$ best respond. Here we place an upper bound B on the ℓ_1-norm of the dual variable to guarantee convergence of our algorithms. This nuisance parameter will be set optimally in our algorithms, and we note that the minimax theorem continues to hold in the presence of this upper bound on λ. We will first analyze the best response problem for both players – i.e. the problem of optimizing the Lagrangian for one of the players *fixing* the strategy of the other player.

4.2 The Auditor's Best Response

Fixing the (h,η_j) variables of the Learner and the estimated values $(\eta_1,\ldots,\eta_{j-1})$ from previous rounds, the Auditor can best respond by solving

$$\operatorname*{argmax}_{\lambda\in\Lambda_j} \mathcal{L}_j\left((h,\eta_j),\lambda\right) \equiv \operatorname*{argmax}_{\lambda\in\Lambda_j} \sum_{r=1}^{j} \sum_{\{i_1,\ldots,i_r\}\subseteq[K]} \lambda_{\{i_1,i_2,\ldots,i_r\}} \cdot \left(L_{i_1}(h) + \ldots + L_{i_r}(h) - \eta_r\right).$$

Since the objective is linear in the dual variables λ, the Auditor can without loss of generality best respond by putting all its mass B on the variable $\lambda_{\{i_1,i_2,\ldots,i_r\}}$ corresponding to the most violated constraint, if one exists. In particular, given any model $h \in \mathcal{H}$ and any ordering $\bar{h}$ induced by h on the groups, we have that the Auditor's best response $\lambda_{\text{best}}(h,\eta_j)$ is

$$\lambda_{\text{best}}(h,\eta_j) = \begin{cases} 0 \in \mathbb{R}^{q_j} & \text{if } \forall r \leq j:\ L_{\bar{h}(1)}(h) + \ldots + L_{\bar{h}(r)}(h) \leq \eta_r \\ \lambda^\star \in \mathbb{R}^{q_j} & \text{if } \exists r \leq j:\ L_{\bar{h}(1)}(h) + \ldots + L_{\bar{h}(r)}(h) > \eta_r \end{cases}$$

where the entries of $\lambda^\star$ are defined as follows.

$$\lambda^\star_{\{i_1,i_2,\ldots,i_r\}} = \begin{cases} B & \text{if } \{i_1,i_2,\ldots,i_r\} = \{\bar{h}(1),\bar{h}(2),\ldots,\bar{h}(r^\star)\} \\ 0 & \text{Otherwise} \end{cases} \tag{5}$$

where $r^\star \in \operatorname{argmax}_{r\leq j}\left(L_{\bar{h}(1)}(h) + \ldots + L_{\bar{h}(r)}(h) - \eta_r\right)$.

Note that the Auditor's best response can be computed efficiently because it only requires sorting the vector of error rates across K groups. We summarize the best response algorithm for the Auditor in Algorithm 1.

Algorithm 1 The Auditor's Best Response (λ_{best}): jth round.

Input: Learner's play (h, η_j), previous estimates $(\eta_1, \ldots, \eta_{j-1})$
Compute $L_k(h)$ for all groups $k \in [K]$;
Find the top j elements of vector $(L_1(h), \ldots, L_K(h))$ and call them:
$L_{\bar{h}(1)}(h) \geq \ldots \geq L_{\bar{h}(j)}(h)$;
if $\forall r \leq j : L_{\bar{h}(1)}(h) + \ldots + L_{\bar{h}(r)}(h) \leq \eta_r$ **then** $\lambda_{out} = 0$;
else Let $r^\star \in \operatorname{argmax}_{r \leq j} \left(L_{\bar{h}(1)}(h) + \ldots + L_{\bar{h}(r)}(h) - \eta_r \right)$, $\lambda_{out} = \lambda^\star$ as in Equation (5) ;
Output: $\lambda_{out} \in \Lambda_j$

4.3 The Learner's Best Response

Given dual weights $\lambda \in \Lambda_j$ chosen by the Auditor, the Learner can best respond by solving

$$\operatorname*{argmin}_{h \in \mathcal{H}, \eta_j \in [0, j \cdot L_M]} \mathcal{L}_j\left((h, \eta_j), \lambda\right).$$

We note that the objective function $\mathcal{L}_j\left((h, \eta_j), \lambda\right)$ can be decomposed into three terms: one that depends only on the model h, another that depends only on η_j, and finally one that is constant (with respect to (h, η_j)). Therefore, this optimization problem is separable for the Learner – the decomposition is formally described below.

$$\mathcal{L}_j\left((h, \eta_j), \lambda\right) = \mathcal{L}_j^1\left(h, \lambda\right) + \mathcal{L}_j^2\left(\eta_j, \lambda\right) + C_j\left(\lambda\right) \tag{6}$$

where

$$\mathcal{L}_j^1\left(h, \lambda\right) \triangleq \sum_{r=1}^{K} w_r(\lambda) L_r(h), \text{ where } w_r(\lambda) \triangleq \sum_{s=0}^{j-1} \sum_{\{i_2, \ldots, i_s\} \subseteq [K] \setminus \{r\}} \lambda_{\{r, i_2, \ldots, i_s\}} \tag{7}$$

$$\mathcal{L}_j^2\left(\eta_j, \lambda\right) \triangleq \left(1 - \sum_{\{i_1, \ldots, i_j\} \subseteq [K]} \lambda_{\{i_1, i_2, \ldots, i_j\}}\right) \eta_j \tag{8}$$

$$C_j\left(\lambda\right) \triangleq -\sum_{r=1}^{j-1} \sum_{\{i_1, \ldots, i_r\} \subseteq [K]} \lambda_{\{i_1, i_2, \ldots, i_r\}} \cdot \eta_r \tag{9}$$

Given this decomposition of the Lagrangian, the best response (h, η_j) of the Learner to the variables λ of the Auditor is as follows:

$$(h, \eta_j) = \left(\operatorname*{argmin}_{h \in \mathcal{H}} \mathcal{L}_j^1\left(h, \lambda\right), \operatorname*{argmin}_{\eta_j \in [0, j \cdot L_M]} \mathcal{L}_j^2\left(\eta_j, \lambda\right) \right).$$

Note that the first optimization problem is a weighted minimization problem over the class $\mathcal{H}$, and the second one is a simple minimization of a linear function. Furthermore, even though in general computing the sums in Equations (7) and (8) can be computationally hard (because they are sums over exponentially many terms), *when the Auditor is best responding (which will be the case in our algorithms), these sums can be computed efficiently.* We formally state this claim in Fact 11.

▶ **Fact 11.** *When the Auditor is using its best response algorithm (Algorithm 1) to respond to the Learner, the Auditor will either output zero or identify a single subset C of groups ($|C| \leq j$) on which the constraints are violated maximally. In the former case, $w_r(\lambda) = 0$ for all r and $1 - \sum_{\{i_1,\ldots,i_j\} \subseteq [K]} \lambda_{\{i_1,i_2,\ldots,i_j\}} = 1$. In the latter case, we have*

$$w_r(\lambda) = B \cdot \mathbb{1}\left[r \in C\right], \quad 1 - \sum_{\{i_1,\ldots,i_j\} \subseteq [K]} \lambda_{\{i_1,i_2,\ldots,i_j\}} = 1 - B \cdot \mathbb{1}\left[|C| = j\right].$$

4.4 Solving the Game with No-Regret Dynamics

Having analyzed the best response problem for both players, we now focus on developing efficient algorithms to approximately solve the two-player zero-sum game defined above, which corresponds to finding an approximate convex lexifair model. The algorithms we propose use no-regret dynamics (see Section 3.1) in which the Learner plays a no-regret learning algorithm and the Auditor best responds according to Algorithm 1. As a consequence, we get that the empirical average of the played strategies $((\hat{h}, \hat{\eta}_j), \hat{\lambda})$ of the players over the course of the iterative algorithms will form a ν-approximate equilibrium of the game for some small value of $\nu \geq 0$ (according to Definition 8). Then, by the following theorem, we can turn these equilibrium guarantees into the fairness guarantees of the output model $\hat{h}$. Its proof can be found in [8].

We remark that what we mean by the empirical average will depend on the setting. If we are in a setting in which the loss function is convex in the model parameters (e.g. logistic or linear regression), then we can actually average the model parameters, and output a single deterministic model. Alternately, if we are in a classification setting in which the loss function (e.g. zero-one loss) is non-convex in the model parameters, then by averaging, we mean using the randomized model that corresponds to the uniform distribution over the empirical play history.

▶ **Theorem 12.** *At round j, let $(\hat{\eta}_1, \ldots, \hat{\eta}_{j-1})$ be any given estimated minimax values from the previous rounds and let the strategies $((\hat{h}, \hat{\eta}_j), \hat{\lambda})$ form a ν-approximate equilibrium of the game for this round, i.e.,*

$$\mathcal{L}_j\left((\hat{h}, \hat{\eta}_j), \hat{\lambda}\right) \leq \min_{h \in \mathcal{H}, \eta_j \in [0, j \cdot L_M]} \mathcal{L}_j\left((h, \eta_j), \hat{\lambda}\right) + \nu, \quad \mathcal{L}_j\left((\hat{h}, \hat{\eta}_j), \hat{\lambda}\right) \geq \max_{\lambda \in \Lambda_j} \mathcal{L}_j\left((\hat{h}, \hat{\eta}_j), \lambda\right) - \nu.$$

We have that $\hat{\eta}_j \leq OPT_j\left(\hat{\eta}_1, \ldots, \hat{\eta}_{j-1}\right) + 2\nu$, and for all $r \leq j$,

$$\max_{\{i_1,\ldots,i_r\} \subseteq [K]} \sum_{s=1}^{r} L_{i_r}(\hat{h}) \leq \hat{\eta}_r + \frac{jL_M + 2\nu}{B}.$$

We will next instantiate this general result to give concrete algorithms for learning convex lexifair models in the regression and classification settings respectively.

5 Finding Lexifair Regression Models

Suppose in this section that $\mathcal{Y} \subseteq \mathbb{R}$ and $\mathcal{H}$ is a class of models in which each model is parametrized by some d-dimensional vector in $\mathbb{R}^d$: $\mathcal{H} = \{h_\theta : \theta \in \Theta\}$ where $\Theta \subseteq \mathbb{R}^d$. In this parametric setting we can think of each parameter $\theta \in \Theta$ as a model and write the loss function as a function of θ. Suppose the loss function $L_z : \Theta \to \mathbb{R}_{\geq 0}$ is differentiable for all z.[3] We will have the Learner play according to the Online Projected Gradient Descent

[3] If it is not differentiable we can use sub-gradients instead of gradients.

Algorithm 2 `LexiFairReg`: Finding a Lexifair Regression Model.

Input: $S = \cup_{k=1}^{K} G_k$ data set consisting of K groups, (ℓ, α) desired fairness parameters, loss function parameters L_M and G, diameter D of the model class Θ

for $j = 1, 2, \ldots, \ell$ **do**

Set $T_j = \frac{4j^2(GD+L_M)^2(2\alpha+jL_M)^2}{\alpha^4}$;

Set $B_j = \frac{\alpha+jL_M}{\alpha}$;

$(\hat{\theta}_j, \hat{\eta}_j) = \texttt{RegNR}(T_j, B_j; \hat{\eta}_1, \ldots, \hat{\eta}_{j-1})$ (Calling Algorithm 3)

end

Output: (ℓ, α)-convex lexifair model $\hat{\theta}_\ell$

algorithm (see Appendix C.1) where the gradients of the corresponding loss function of the game for the Learner (i.e. $\mathcal{L}_j((\theta, \eta_j), \lambda)$) can be computed using Equations (7) and (8), and the decomposition given in (6):

$$\nabla_\theta \mathcal{L}_j((\theta, \eta_j), \lambda) = \nabla_\theta \mathcal{L}_j^1(\theta, \lambda) = \sum_{r=1}^{K} w_r(\lambda) \nabla_\theta L_r(\theta), \tag{10}$$

$$\nabla_{\eta_j} \mathcal{L}_j((\theta, \eta_j), \lambda) = \nabla_{\eta_j} \mathcal{L}_j^2(\eta_j, \lambda) = 1 - \sum_{\{i_1, \ldots, i_j\} \subseteq [K]} \lambda_{\{i_1, i_2, \ldots, i_j\}}. \tag{11}$$

The algorithm for this setting is given as Algorithm 2, which makes calls to a subroutine (Algorithm 3) that solves the two-player zero-sum games defined above by having the Learner play Online Projected Gradient Descent (see Appendix C) and the Auditor best respond using Algorithm 1. Note that since the Auditor is best responding, computing the sums in Equations (10) and (11) can be done efficiently per Fact 11.

▶ **Theorem 13** (Lexifairness for Regression). *Suppose $\Theta \subseteq \mathbb{R}^d$ is convex, compact, and bounded with diameter D: $\sup_{\theta, \theta' \in \Theta} \|\theta - \theta'\|_2 \leq D$. Suppose the loss function $L_z : \Theta \to \mathbb{R}_{\geq 0}$ is convex and that there exists constants L_M and G such that $L_z(\cdot) \leq L_M$ and $\|\nabla_\theta L_z(\cdot)\|_2 \leq G$, for all data points $z \in \mathcal{Z}$. We have that for any $\ell \leq K$ and any $\alpha \geq 0$, the model $\hat{\theta}_\ell \in \Theta$ output by Algorithm 2 is (ℓ, α)-convex lexicographic fair.*

The proof of this theorem (which can be found in Appendix A) involves bounding the regret of each player, and then appealing to Theorem 12.

6 Finding Lexifair Classification Models

In this section we briefly discuss how we can find lexifair models in a classification setting. All details including our algorithm for this setting and its analysis can be found in [8]. Suppose in this section that $\mathcal{Y} = \{0, 1\}$ and our model class $\mathcal{H}$ is the probability simplex over a class of deterministic binary classifiers. We slightly abuse notation and write $\mathcal{H}$ for the given class of deterministic classifiers and write $\Delta\mathcal{H} \triangleq \{p : p \text{ is a distribution over } \mathcal{H}\}$ for the probability simplex, and work with $\Delta\mathcal{H}$ as our model class. Let the loss function be zero-one loss: for any $h \in \mathcal{H}$: $L_z(h) = \mathbb{1}\{h(x) \neq y\}$. The loss of any randomized model p on data point z is defined as the *expected loss* of h on z when h is sampled from $\mathcal{H}$ according to the distribution p. In other words,

$$L_z(p) \triangleq \mathbb{E}_{h \sim p}[L_z(h)]$$

Algorithm 3 RegNR: jth round.

Input: Number of rounds T, dual variable upper bound B, previous estimates $(\eta_1, \ldots, \eta_{j-1})$

Set learning rates $\eta = \frac{D}{jBG\sqrt{T}}$ and $\eta' = \frac{jL_M}{(1+B)\sqrt{T}}$;

Initialize the Learner: $\theta^1 \in \Theta, \eta_j^1 \in [0, j \cdot L_M]$;

for $t = 1, 2, \ldots, T$ **do**

Learner plays (θ^t, η_j^t);

Auditor best responds: $\lambda^t = \lambda_{\text{best}}(\theta^t, \eta_j^t; (\eta_1, \ldots, \eta_{j-1}))$ using Algorithm 1;

Learner updates its actions using Projected Gradient Descent:

$$\theta^{t+1} = \text{Proj}_{\Theta}\left(\theta^t - \eta \cdot \nabla_\theta \mathcal{L}_j(\theta^t, \eta_j^t, \lambda^t)\right)$$

$$\eta_j^{t+1} = \text{Proj}_{[0, j \cdot L_M]}\left(\eta_j^t - \eta' \cdot \nabla_{\eta_j} \mathcal{L}_j(\theta^t, \eta_j^t, \lambda^t)\right)$$

where the gradients are given in Equations (10) and (11).

end

Output: the average play $\hat{\theta} = \frac{1}{T}\sum_{t=1}^{T} \theta^t \in \Theta$, and $\hat{\eta}_j = \frac{1}{T}\sum_{t=1}^{T} \eta_j^t \in [0, j \cdot L_M]$.

which is convex (linear) in the model p (weights of the distribution). We will also assume that the model class $\mathcal{H}$ has finite VC dimension. Sauer's Lemma will then imply that for any finite dataset, $\mathcal{H}$ induces only finitely many labelings. This will serve two purposes. First, it allows us to write the optimization problem as a linear program with *finitely many* variables (probability weights over the set of all possible induced labelings), and therefore appeal to strong duality. Second, it allows us to pose the Learner's best response problem as an n-dimensional *linear optimization* problem, over the only exponentially many labelings of the n data points. This is what will allow us to apply Follow the Perturbed Leader and obtain oracle-efficient no-regret learning guarantees for the Learner. Here we are following an approach similar to that of [19]. The final algorithm will then have the Learner play according to Follow the Perturbed Leader (given access to a *Cost Sensitive Classification Oracle* for the function class $\mathcal{H}$), and have the Auditor best respond.

▶ **Theorem 14** (Lexifairness for Classification). *Let $\mathcal{H}$ be any class of binary classifiers with finite VC dimension, and let $L_z(p) = \mathbb{E}_{h\sim p}[L_z(h)]$ for any randomized model $p \in \Delta\mathcal{H}$ where $L_z(h) = \mathbb{1}\{h(x) \neq y\}$ is the zero-one loss. Fix any $\ell \leq K$ and any $\alpha \geq 0$. There exists an oracle-efficient algorithm (see [8]) such that for any $\delta > 0$, with probability at least $1 - \delta$, its output model is (ℓ, α)-convex lexicographic fair.*

7 Generalization

In this section, we turn our attention to out of sample bounds. Standard uniform convergence statements would tell us that if we have enough samples from every group, then our in-sample group errors are good estimates of our out of sample group errors. However, this alone does not directly imply that we satisfy approximate lexifairness out of sample. We prove this is the case below. Our ability to prove out of sample bounds crucially relies on our definitional choices that removed the instability of the naive Definition 2. Specifically, we show that if:

1. Our base class $\mathcal{H}$ satisfies a standard uniform convergence bound across every group (so that we can control the maximum gap between in and out of sample error across every $h \in \mathcal{H}$, within each group k), and

2. We have a model that is approximately convex lexifair on our dataset $S \sim \mathcal{P}^n$, then then our model is also appropriately convex lexifair on the underlying distribution (with some loss in the approximation parameter).

▶ **Theorem 15** (Generalization for Convex Lexifairness). *Fix any distribution $\mathcal{P}$. Suppose for every $\delta > 0$, there exists $\beta(\delta)$ such that the following uniform convergence bound holds.*

$$\Pr_S \left[\max_{h \in \mathcal{H}, k \in [K]} |L_k(h, S) - L_k(h, \mathcal{P})| > \beta(\delta) \right] < \delta$$

where S is a data set sampled i.i.d. from $\mathcal{P}$. We have that for every data set S sampled i.i.d. from $\mathcal{P}$, if a model h satisfies (ℓ, α)-convex lexicographic fairness with respect to S, then with probability at least $1 - \delta$ it also satisfies (ℓ, α')-convex lexicographic fairness with respect to $\mathcal{P}$ for $\alpha' = \alpha + 2\ell\beta(\delta)$.

The proof of the theorem is given in Appendix B. We can now instantiate the above theorem in a classification setting in which we have VC-type convergence bounds. A corollary that we get by applying standard uniform convergence bounds for finite VC classes is the following:

▶ **Corollary 16** (Generalization for Convex Lexifairness: Classification Setting). *Suppose $\mathcal{H}$ is a class of binary classifiers with VC dimension $d_{\mathcal{H}}$ and let $L_z(p) = \mathbb{E}_{h \sim p}[L_z(h)]$ for any randomized model $p \in \Delta\mathcal{H}$ where $L_z(h) = \mathbb{1}\{h(x) \neq y\}$ is the zero-one loss. We have that for every $\mathcal{P}$, every data set $S \equiv \{G_k\}_k$ of size n sampled i.i.d. from $\mathcal{P}$, if a model $p \in \Delta\mathcal{H}$ satisfies (ℓ, α)-convex lexicographic fairness with respect to S, then with probability at least $1 - \delta$ it also satisfies $(\ell, 2\alpha)$-convex lexicographic fairness with respect to $\mathcal{P}$ provided that*

$$\min_{1 \leq k \leq K} |G_k| = \Omega\left(\frac{l^2 \left(d_{\mathcal{H}} \log(n) + \log(K/\delta) \right)}{\alpha^2} \right).$$

We have here proven a generalization theorem for convex lexifairness (Definition 5) which is the definition that our algorithms satisfy. We also prove a generalization theorem for lexifairness (Definition 3) which can be found in [8].

References

1 Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

2 M. Allalouf and Y. Shavitt. Centralized and distributed algorithms for routing and weighted max-min fair bandwidth allocation. *IEEE/ACM Transactions on Networking*, 16(5):1015–1024, 2008. doi:10.1109/TNET.2007.905605.

3 Arash Asadpour and Amin Saberi. An approximation algorithm for max-min fair allocation of indivisible goods. *SIAM Journal on Computing*, 39(7):2970–2989, 2010.

4 Richard Berk, Hoda Heidari, Shahin Jabbari, Michael Kearns, and Aaron Roth. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, page 0049124118782533, 2018.

5 Robert S. Chen, Brendan Lucier, Yaron Singer, and Vasilis Syrgkanis. Robust optimization for non-convex objectives. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, pages 4705–4714. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/10c66082c124f8afe3df4886f5e516e0-Paper.pdf.

6 Andrew Cotter, Heinrich Jiang, and Karthik Sridharan. Two-player games for efficient non-convex constrained optimization. In Aurélien Garivier and Satyen Kale, editors, *Proceedings of the 30th International Conference on Algorithmic Learning Theory*, volume 98 of *Proceedings of Machine Learning Research*, pages 300–332, Chicago, Illinois, 22–24 March 2019. PMLR. URL: http://proceedings.mlr.press/v98/cotter19a.html.

7 Emilie Danna, Avinatan Hassidim, Haim Kaplan, Alok Kumar, Yishay Mansour, Danny Raz, and Michal Segalov. Upward max-min fairness. *J. ACM*, 64(1), 2017. doi:10.1145/3011282.

8 Emily Diana, Wesley Gill, Ira Globus-Harris, Michael Kearns, Aaron Roth, and Saeed Sharifi-Malvajerdi. Lexicographically fair learning: Algorithms and generalization. *arXiv preprint*, 2021. arXiv:2102.08454.

9 Emily Diana, Wesley Gill, Michael Kearns, Krishnaram Kenthapadi, and Aaron Roth. Convergent algorithms for (relaxed) minimax fairness. *arXiv preprint*, 2020. arXiv:2011.03108.

10 Dongliang Xie, Xin Wang, and Linhui Ma. Lexicographical order max-min fair source quota allocation in mobile delay-tolerant networks. In *2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)*, pages 1–6, 2016. doi:10.1109/IWQoS.2016.7590424.

11 Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Max Leiserson. Decoupled classifiers for group-fair and efficient machine learning. In *Conference on Fairness, Accountability and Transparency*, pages 119–133, 2018.

12 Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, COLT '96, page 325–332, New York, NY, USA, 1996. Association for Computing Machinery. doi:10.1145/238061.238163.

13 Varun Gupta, Christopher Jung, Georgy Noarov, Mallesh M Pai, and Aaron Roth. Online multivalid learning: Means, moments, and prediction intervals. *arXiv preprint*, 2021. arXiv:2101.01739.

14 Ellen L. Hahne. Round-robin scheduling for max-min fairness in data networks. *IEEE Journal on Selected Areas in communications*, 9(7):1024–1039, 1991.

15 Ursula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1939–1948. PMLR, 2018.

16 C. Jung, S. Neel, A. Roth, L. Stapleton, and S. Wu. An algorithmic framework for fairness elicitation. *Preprint*, 2020.

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

18 Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005. Learning Theory 2003. doi:10.1016/j.jcss.2004.10.016.

19 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. PMLR, 2018.

20 Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. An empirical study of rich subgroup fairness for machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 100–109, 2019.

21 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.

22 Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed H. Chi. Fairness without demographics through adversarially reweighted learning, 2020. arXiv:2006.13114.

23 Natalie Martinez, Martin Bertran, and Guillermo Sapiro. Minimax Pareto fairness: A multi objective perspective. In *Proceedings of the 37th International Conference on Machine Learning*. Vienna, Austria, PMLR 119, 2020.

24 Shira Mitchell, Eric Potash, Solon Barocas, Alexander D'Amour, and Kristian Lum. Algorithmic fairness: Choices, assumptions, and definitions. *Annual Review of Statistics and Its Application*, 8, 2020.

25 D. Nace and M. Pióro. Max-min fairness and its applications to routing and load-balancing in communication networks: a tutorial. *IEEE Communications Surveys and Tutorials*, 10, 2008.

26 W. Ogryczak and Warsaw. Lexicographic max-min optimization for efficient and fair bandwidth allocation. *International network optimization conference (INOC)*, January 2007.

27 Wlodzimierz Ogryczak, Hanan Luss, Dritan Nace, and Michał Pióro. Fair Optimization and Networks: Models, Algorithms, and Applications. *Journal of Applied Mathematics*, September 2014. doi:10.1155/2014/340913.

28 B. Radunovic and J. Le Boudec. A unified framework for max-min and min-max fairness with applications. *IEEE/ACM Transactions on Networking*, 15(5):1073–1083, 2007. doi:10.1109/TNET.2007.896231.

29 Samira Samadi, Uthaipon Tantipongpipat, Jamie H Morgenstern, Mohit Singh, and Santosh Vempala. The price of fair PCA: One extra dimension. In *Advances in Neural Information Processing Systems*, pages 10976–10987, 2018.

30 Saeed Sharifi-Malvajerdi, Michael Kearns, and Aaron Roth. Average individual fairness: Algorithms, generalization and experiments. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/0e1feae55e360ff05fef58199b3fa521-Paper.pdf.

31 Berk Ustun, Yang Liu, and David Parkes. Fairness without harm: Decoupled classifiers with preference guarantees. In *International Conference on Machine Learning*, pages 6373–6382, 2019.

32 X. Wang, K. Kar, and J. Pang. Lexicographic max-min fair rate allocation in random access wireless networks. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 1294–1300, 2006. doi:10.1109/CDC.2006.377233.

33 Congzhou Zhou and N. F. Maxemchuk. Scalable max-min fairness in wireless ad hoc networks. In Jun Zheng, Shiwen Mao, Scott F. Midkiff, and Hua Zhu, editors, *Ad Hoc Networks*, pages 79–93, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

34 Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*, ICML'03, page 928–935. AAAI Press, 2003.

A Proofs from Section 5

▶ **Theorem 13** (Lexifairness for Regression). *Suppose $\Theta \subseteq \mathbb{R}^d$ is convex, compact, and bounded with diameter D: $\sup_{\theta,\theta' \in \Theta} \|\theta - \theta'\|_2 \leq D$. Suppose the loss function $L_z : \Theta \to \mathbb{R}_{\geq 0}$ is convex and that there exists constants L_M and G such that $L_z(\cdot) \leq L_M$ and $\|\nabla_\theta L_z(\cdot)\|_2 \leq G$, for all data points $z \in \mathcal{Z}$. We have that for any $\ell \leq K$ and any $\alpha \geq 0$, the model $\hat{\theta}_\ell \in \Theta$ output by Algorithm 2 is (ℓ, α)-convex lexicographic fair.*

Proof. We will show that for every round j, the model $\hat{\theta}_j$ computed by our algorithm is (j, α)-convex lexicographic fair, and as a consequence, the very last model $(\hat{\theta}_\ell)$ is (ℓ, α)-convex lexicographic fair. Fix any round $j \leq \ell$. Let $(\theta^t, \eta_j^t, \lambda^t)_{t=1}^T$ be the sequence of plays in the no-regret dynamics of Algorithm 3 in this round. First, note that by the decomposition given in Equation (6), we have

$$\sum_{t=1}^{T} \mathcal{L}_j\left((\theta^t, \eta_j^t), \lambda^t\right) - \min_{\theta\in\Theta, \eta_j\in[0, j\cdot L_M]} \sum_{t=1}^{T} \mathcal{L}_j\left((\theta, \eta_j), \lambda^t\right)$$
$$= \left\{\sum_{t=1}^{T} \mathcal{L}_j^1\left(\theta^t, \lambda^t\right) - \min_{\theta\in\Theta} \sum_{t=1}^{T} \mathcal{L}_j^1\left(\theta, \lambda^t\right)\right\} + \left\{\sum_{t=1}^{T} \mathcal{L}_j^2\left(\eta_j^t, \lambda^t\right) - \min_{\eta_j\in[0, j\cdot L_M]} \sum_{t=1}^{T} \mathcal{L}_j^2\left(\eta_j, \lambda^t\right)\right\}.$$

In other words, we can decompose the regret of the Learner into two terms: one is the regret of gradient descent plays corresponding to θ, and the other one is the corresponding regret of gradient descent plays for η_j. Note that by Equations (10) and (11) we have the following bounds on the norm of gradients for the Learner. We also use the fact that when the Auditor is best responding, $w_r(\lambda^t)$ can be simplified as in Fact 11.

$$\left\|\nabla_\theta \mathcal{L}_j\left((\theta, \eta_j), \lambda^t\right)\right\|_2 \leq \sum_{r=1}^{K} \left|w_r(\lambda^t)\right| \cdot \left\|\nabla_\theta L_r(\theta)\right\|_2 \leq jBG$$

$$\left\|\nabla_{\eta_j} \mathcal{L}_j\left((\theta, \eta_j), \lambda^t\right)\right\|_2 = \left|1 - \sum_{\{i_1,\ldots,i_j\}\subseteq[K]} \lambda^t_{\{i_1,i_2,\ldots,i_j\}}\right| \leq 1 + B$$

Now letting $\eta = \frac{D}{jBG\sqrt{T}}$ and $\eta' = \frac{jL_M}{(1+B)\sqrt{T}}$ in Algorithm 3 and using the regret bound of Online Projected Gradient Desccent (Theorem 18), we have

$$\sum_{t=1}^{T} \mathcal{L}_j^1\left(\theta^t, \lambda^t\right) - \min_{\theta\in\Theta} \sum_{t=1}^{T} \mathcal{L}_j^1\left(\theta, \lambda^t\right) \leq jBGD\sqrt{T}$$

$$\sum_{t=1}^{T} \mathcal{L}_j^2\left(\eta_j^t, \lambda^t\right) - \min_{\eta_j\in[0, j\cdot L_M]} \sum_{t=1}^{T} \mathcal{L}_j^2\left(\eta_j, \lambda^t\right) \leq j(B+1)L_M\sqrt{T}$$

and therefore the regret of the Learner can be bounded by

$$\sum_{t=1}^{T} \mathcal{L}_j\left((\theta^t, \eta_j^t), \lambda^t\right) - \min_{\theta\in\Theta, \eta_j\in[0, j\cdot L_M]} \sum_{t=1}^{T} \mathcal{L}_j\left((\theta, \eta_j), \lambda^t\right) \leq j(GD + L_M)(B+1)\sqrt{T} := \nu_j T.$$

Let $\nu_j \triangleq j(GD + L_M)(B+1)/\sqrt{T}$. Now using the guarantees of the no-regret dynamics (Theorem 9), the average play of the players $(\hat{\theta}, \hat{\eta}_j, \hat{\lambda})$ forms a ν_j-approximate equilibrium of the game in the sense that

$$\mathcal{L}_j\left((\hat{\theta}, \hat{\eta}_j), \hat{\lambda}\right) \leq \min_{\theta\in\Theta, \eta_j\in[0, j\cdot L_M]} \mathcal{L}_j\left((\theta, \eta_j), \hat{\lambda}\right) + \nu_j, \quad \mathcal{L}_j\left((\hat{\theta}, \hat{\eta}_j), \hat{\lambda}\right) \geq \max_{\lambda\in\Lambda_j} \mathcal{L}_j\left((\hat{\theta}, \hat{\eta}_j), \lambda\right) - \nu_j.$$

Finally, using Theorem 12 we can turn these into the following guarantees. First,

$$\hat{\eta}_j \leq OPT_j\left(\hat{\eta}_1, \ldots, \hat{\eta}_{j-1}\right) + 2\nu_j \tag{12}$$

and second, for all $r \leq j$,

$$\max_{\{i_1,\ldots,i_r\}\subseteq[K]} \sum_{s=1}^{r} L_{i_r}(\hat{\theta}_j) \leq \hat{\eta}_r + \frac{jL_M + 2\nu_j}{B}. \tag{13}$$

Define $\epsilon_r \triangleq \hat{\eta}_r - OPT_r\left(\hat{\eta}_1, \ldots, \hat{\eta}_{r-1}\right)$ for all $r \leq j$ (ϵ's here are basically *constant* mappings in $\mathbb{R}^{\mathcal{H}}$). We immediately have from Equation (12) that: $\epsilon_r \leq 2\nu_r$, for all $r \leq j$. Now let $\vec{\epsilon} = (\epsilon_1, \ldots, \epsilon_j)$, and let $\mathcal{F}^{\vec{\epsilon}}_{(0)} = \Theta$ be the initial model class. Note that according to Definition 5 and given the defined $\vec{\epsilon}$, we have for every $r \leq j$,

$$\min_{\theta \in \mathcal{F}^{\vec{\epsilon}}_{(r-1)}} \max_{\{i_1,\ldots,i_r\} \subseteq [K]} \sum_{s=1}^{r} L_{i_r}(\theta) \equiv OPT_r\left(\hat{\eta}_1, \ldots, \hat{\eta}_{r-1}\right).$$

And therefore, by Equation (13), for all $r \leq j$:

$$\begin{aligned}
\max_{\{i_1,\ldots,i_r\} \subseteq [K]} \sum_{s=1}^{r} L_{i_r}(\hat{\theta}_j) &\leq \hat{\eta}_r + \frac{jL_M + 2\nu_r}{B} \\
&= OPT_r\left(\hat{\eta}_1, \ldots, \hat{\eta}_{r-1}\right) + \epsilon_r + \frac{jL_M + 2\nu_r}{B} \\
&= \min_{g \in \mathcal{F}^{\vec{\epsilon}}_{(r-1)}} \max_{\{i_1,\ldots,i_r\} \subseteq [k]} \sum_{s=1}^{r} L_{i_r}(g) + \epsilon_r + \frac{jL_M + 2\nu_r}{B}
\end{aligned}$$

which completes the proof by the choice of $\nu_r = \frac{\alpha}{2}$ for all $r \leq j$ (to guarantee that $\|\vec{\epsilon}\|_\infty \leq \alpha$), and $B = \frac{\alpha + jL_M}{\alpha}$. Note that this setting of parameters, together with $\nu_j = j(GD + L_M)(B+1)/\sqrt{T}$, implies that

$$T = \frac{4j^2(GD + L_M)^2(2\alpha + jL_M)^2}{\alpha^4}.$$ ◀

B Proofs from Section 7

▶ **Theorem 15** (Generalization for Convex Lexifairness). *Fix any distribution $\mathcal{P}$. Suppose for every $\delta > 0$, there exists $\beta(\delta)$ such that the following uniform convergence bound holds.*

$$\Pr_S\left[\max_{h \in \mathcal{H}, k \in [K]} \left|L_k\left(h, S\right) - L_k\left(h, \mathcal{P}\right)\right| > \beta(\delta)\right] < \delta$$

where S is a data set sampled i.i.d. from $\mathcal{P}$. We have that for every data set S sampled i.i.d. from $\mathcal{P}$, if a model h satisfies (ℓ, α)-convex lexicographic fairness with respect to S, then with probability at least $1 - \delta$ it also satisfies (ℓ, α')-convex lexicographic fairness with respect to $\mathcal{P}$ for $\alpha' = \alpha + 2\ell\beta(\delta)$.

Proof. Fix a distribution $\mathcal{P}$ and a data set S sampled *i.i.d.* from $\mathcal{P}$. Suppose h satisfies (ℓ, α)-convex lexicographic fairness with respect to S. Therefore, according to our convex lexifairness definition, there exists a sequence of mappings $\vec{\epsilon} = (\epsilon_1, \ldots, \epsilon_\ell)$ where $\epsilon_j \in \mathbb{R}^{\mathcal{H}}$, and a sequence of function classes $\{\mathcal{F}^{\vec{\epsilon}}_{(j)}(S)\}_j$ such that

$$\max_{1 \leq j \leq \ell} \left\{\max_{h' \in \mathcal{H}} \epsilon_j(h')\right\} \leq \alpha$$

and that for all $j \leq \ell$:

$$\max_{\{i_1,\ldots,i_j\} \subseteq [K]} \sum_{r=1}^{j} L_{i_r}(h, S) \leq \min_{g \in \mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S)} \max_{\{i_1,\ldots,i_j\} \subseteq [K]} \sum_{r=1}^{j} L_{i_r}(g, S) + \epsilon_j(h) + \alpha \tag{14}$$

where recall that $\mathcal{F}^{\vec{\epsilon}}_{(0)}(S) = \mathcal{H}$ and that for all $j \in [\ell]$,

$$\mathcal{F}^{\vec{\epsilon}}_{(j)}(S) = \left\{h' \in \mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S) : \max_{\{i_1,\ldots,i_j\} \subseteq [K]} \sum_{r=1}^{j} L_{i_r}(h', S) \leq \min_{g \in \mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S)} \max_{\{i_1,\ldots,i_j\} \subseteq [K]} \sum_{r=1}^{j} L_{i_r}(g, S) + \epsilon_j(h')\right\}.$$

Let us define a mapping $\nu_j^1 : \mathcal{H} \to \mathbb{R}$ such that for every $h' \in \mathcal{H}$,

$$\nu_j^1(h') \triangleq \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h', \mathcal{P}) - \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h', S)$$

and let

$$\nu_j^2 \triangleq \min_{g\in\mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S)} \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(g, S) - \min_{g\in\mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S)} \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(g, \mathcal{P})$$

Now define for every $h' \in \mathcal{H}$, $\tau_j(h') \triangleq \epsilon_j(h') + \nu_j^1(h') + \nu_j^2$ and let $\mathcal{F}^{\vec{\tau}}_{(j)}(\mathcal{P})$ be defined according to our convex lexifairness definition with the sequence of mappings defined by $\vec{\tau} = (\tau_1, \ldots, \tau_\ell)$. In other words, $\mathcal{F}^{\vec{\tau}}_{(0)}(\mathcal{P}) = \mathcal{H}$, and for all $j \in [\ell]$,

$$\mathcal{F}^{\vec{\tau}}_{(j)}(\mathcal{P}) = \left\{ h' \in \mathcal{F}^{\vec{\tau}}_{(j-1)}(\mathcal{P}) : \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h', \mathcal{P}) \le \min_{g\in\mathcal{F}^{\vec{\tau}}_{(j-1)}(\mathcal{P})} \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(g, \mathcal{P}) + \tau_j(h') \right\}.$$

▶ **Claim 17.** *For all j, $\mathcal{F}^{\vec{\tau}}_{(j)}(\mathcal{P}) = \mathcal{F}^{\vec{\epsilon}}_{(j)}(S)$.*

Proof. We use induction on j. For $j = 0$, we have $\mathcal{F}^{\vec{\tau}}_{(0)}(\mathcal{P}) = \mathcal{F}^{\vec{\epsilon}}_{(0)}(S) = \mathcal{H}$. For $j \ge 1$, we have

$$\begin{aligned}
&h' \in \mathcal{F}^{\vec{\tau}}_{(j)}(\mathcal{P}) \\
&\iff h' \in \mathcal{F}^{\vec{\tau}}_{(j-1)}(\mathcal{P}), \\
&\qquad \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h', \mathcal{P}) \le \min_{g\in\mathcal{F}^{\vec{\tau}}_{(j-1)}(\mathcal{P})} \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(g, \mathcal{P}) + \tau_j(h') \\
&\iff h' \in \mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S), \\
&\qquad \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h', \mathcal{P}) \le \min_{g\in\mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S)} \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(g, \mathcal{P}) + \tau_j(h') \\
&\iff h' \in \mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S), \\
&\qquad \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h', S) \le \min_{g\in\mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S)} \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(g, S) + \epsilon_j(h') \\
&\iff h' \in \mathcal{F}^{\vec{\epsilon}}_{(j)}(S)
\end{aligned}$$

where the second line follows from the induction assumption ($\mathcal{F}^{\vec{\tau}}_{(j-1)}(\mathcal{P}) = \mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S)$) and the third line follows from the definition of τ_j. This establishes our claim. ◀

We have that for all $j \le \ell$, the model h satisfies

$$\max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h, \mathcal{P}) = \max_{\{i_1,\ldots,i_j\}\subseteq[K]} \sum_{r=1}^{j} L_{i_r}(h, S) + \nu_j^1(h) \le \ldots$$

$$\begin{aligned}
\ldots &\leq \min_{g \in \mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S)} \max_{\{i_1,\ldots,i_j\} \subseteq [K]} \sum_{r=1}^{j} L_{i_r}(g, S) + \epsilon_j(h) + \alpha + \nu_j^1(h) \\
&= \min_{g \in \mathcal{F}^{\vec{\epsilon}}_{(j-1)}(S)} \max_{\{i_1,\ldots,i_j\} \subseteq [K]} \sum_{r=1}^{j} L_{i_r}(g, \mathcal{P}) + \nu_j^2 + \epsilon_j(h) + \alpha + \nu_j^1(h) \\
&= \min_{g \in \mathcal{F}^{\vec{\tau}}_{(j-1)}(\mathcal{P})} \max_{\{i_1,\ldots,i_j\} \subseteq [K]} \sum_{r=1}^{j} L_{i_r}(g, \mathcal{P}) + \tau_j(h) + \alpha
\end{aligned}$$

where the first inequality follows from Equation (14). The third line follows from the definition of ν_j^2. The last equality follows from Claim 17 and the fact that $\tau_j(h) = \epsilon_j(h) + \nu_j^1(h) + \nu_j^2$. The proof is complete by the uniform convergence bound provided in the theorem statement. With probability at least $1-\delta$ over the random draws of the data set S, we have $\max_{h' \in \mathcal{H}} |\nu_j^1(h')| \leq j\beta(\delta)$ and $|\nu_j^2| \leq j\beta(\delta)$, and hence for all $j \leq \ell$,

$$\begin{aligned}
\|\tau\|_\infty &= \max_{1 \leq j \leq \ell} \left\{ \max_{h' \in \mathcal{H}} \tau_j(h') \right\} \\
&\leq \max_{1 \leq j \leq \ell} \left\{ \max_{h' \in \mathcal{H}} \epsilon_j(h') \right\} + \max_{1 \leq j \leq \ell} \left\{ \max_{h' \in \mathcal{H}} |\nu_j^1(h')| + |\nu_j^2| \right\} \\
&\leq \alpha + 2l\beta(\delta).
\end{aligned}$$

◀

C No-Regret Learning Algorithms

C.1 Online Projected Gradient Descent

Consider an online setting where a learner is playing against an adversary. The learner's action space is some Euclidean subspace $\Theta \subseteq \mathbb{R}^d$ which is equipped with the ℓ_2 norm denoted by $\|\cdot\|_2$. At every round t of the interaction between the learner and the adversary, the learner picks an action $\theta^t \in \Theta$ and the adversary chooses a loss function $\ell^t : \Theta \to \mathbb{R}_{\geq 0}$. The learner then incurs a loss of $\ell^t(\theta^t)$ at that round. Suppose the learner is using some algorithm $\mathcal{A}$ to update its actions from round to round. The goal for the learner is that the regret of $\mathcal{A}$ defined as

$$R_{\mathcal{A}}(T) \triangleq \sum_{t=1}^{T} \ell^t(\theta^t) - \min_{\theta \in \Theta} \sum_{t=1}^{T} \ell^t(\theta)$$

grows sublinearly in T. When Θ and the loss functions played by the adversary are convex, a standard choice of algorithm to use for the learner is *Online Projected Gradient Descent* (Algorithm 4), where in each round, the algorithm updates its action θ^{t+1} for the next round by taking a step in the opposite direction of the gradient of the loss function evaluated at the action of that round: $\nabla \ell^t(\theta^t)$. The updated action is then projected onto the feasible action space Θ: $\text{Proj}_\Theta(\theta) \triangleq \text{argmin}_{\theta' \in \Theta} \|\theta - \theta'\|_2$. Note if the loss functions are not differentiable, we can use subgradients (which are defined given the convexity of the loss functions) instead of gradients and the guarantees will remain.

▶ **Theorem 18** (Regret for Online Projected Gradient Descent [34]). *Suppose $\Theta \subseteq \mathbb{R}^d$ is convex, compact and has bounded diameter D: $\sup_{\theta, \theta' \in \Theta} \|\theta - \theta'\|_2 \leq D$. Suppose for all t, the loss functions ℓ^t are convex and that there exists some G such that $\|\nabla \ell^t(\cdot)\|_2 \leq G$. Let $\mathcal{A}$ be Algorithm 4 run with learning rate $\eta = D/(G\sqrt{T})$. We have that for every sequence of loss functions $(\ell^1, \ell^2, \ldots, \ell^T)$ played by the adversary, $R_{\mathcal{A}}(T) \leq GD\sqrt{T}$.*

Algorithm 4 Online Projected Gradient Descent.

Input: learning rate η
Initialize the learner $\theta^1 \in \Theta$;
for $t = 1, 2, \ldots$ **do**
  Learner plays action θ^t;
  Adversary plays loss function ℓ^t;
  Learner incurs loss of $\ell^t(\theta^t)$;
  Learner updates its action:

$$\theta^{t+1} = \text{Proj}_\Theta\left(\theta^t - \eta \nabla \ell^t(\theta^t)\right)$$

end

Algorithm 5 Follow the Perturbed Leader (FTPL).

Input: learning rate η
Initialize the learner $a^1 \in A$;
for t = *1,2, ...* **do**
  Learner plays action a^t;
  Adversary plays loss vector ℓ^t;
  Learner incurs loss of $\langle \ell^t, a^t \rangle$. Learner updates its action:

$$a^{t+1} = \operatorname*{argmin}_{a \in A} \left\{ \left\langle \sum_{s \le t} \ell^s, a \right\rangle + \frac{1}{\eta} \langle \xi^t, a \rangle \right\}$$

  where $\xi^t \sim Uniform\left([0,1]^d\right)$, independent of every other randomness.
end

C.2 Follow the Perturbed Leader

Here assume the learner's action space is $A \subseteq \{0,1\}^d$. At every round t, the learner chooses an action $a^t \in A$ and then the adversary plays a loss vector $\ell^t \in \mathbb{R}^d$. The learner then incurs a loss of $\langle \ell^t, a^t \rangle$ which is the inner product if a^t and ℓ^t. Suppose the learner is using some algorithm $\mathcal{A}$ to pick its actions in every round. The goal for the learner is to ensure that the regret of $\mathcal{A}$ defined as $R_{\mathcal{A}}(T) \triangleq \sum_{t=1}^T \langle \ell^t, a^t \rangle - \min_{a \in \mathcal{A}} \sum_{t=1}^T \langle \ell^t, a \rangle$ grows sublinearly in T. *Follow the Perturbed Leader (FTPL)* ([18]), which is described in Algorithm 5, can provide guarantees in this setting.

▶ **Theorem 19** (Regret of FTPL [18]). *Suppose for all t, $\ell^t \in [-M, M]^d$. Let $\mathcal{A}$ be Algorithm 5 run with learning rate $\eta = 1/(M\sqrt{dT})$. We have that for every sequence of loss vectors $(\ell^1, \ell^2, \ldots, \ell^T)$ played by the adversary, $\mathbb{E}[R_{\mathcal{A}}(T)] \le 2Md^{3/2}\sqrt{T}$, where expectation is taken with respect to the randomness in $\mathcal{A}$.*