Convergent Algorithms for (Relaxed) Minimax Fairness

Emily Diana[1,2], Wesley Gill[1,2], Michael Kearns[1,2], Krishnaram Kenthapadi[2], and Aaron Roth[1,2]

[1] *University of Pennsylvania*
[2] *Amazon AWS AI*

November 9, 2020

Abstract

We consider a recently introduced framework in which fairness is measured by worst-case outcomes across groups, rather than by the more standard *difference* between group outcomes. In this framework we provide provably convergent *oracle-efficient* learning algorithms (or equivalently, reductions to non-fair learning) for *minimax group fairness*. Here the goal is that of minimizing the maximum loss across all groups, rather than equalizing group losses. Our algorithms apply to both regression and classification settings and support both overall error and false positive or false negative rates as the fairness measure of interest. They also support relaxations of the fairness constraints, thus permitting study of the tradeoff between overall accuracy and minimax fairness. We compare the experimental behavior and performance of our algorithms across a variety of fairness-sensitive data sets and show cases in which minimax fairness is strictly and strongly preferable to equal outcome notions, in the sense that equal outcomes can only be obtained by artificially inflating the harm inflicted on some groups compared to what they suffer under the minimax solution.

1 Introduction

Machine learning researchers and practitioners have often focused on achieving group fairness with respect to protected attributes such as race, gender, or ethnicity. Equality of error rates is one of the most intuitive and well-studied group fairness notions, and in enforcing it one often implicitly hopes that higher error rates on protected or "disadvantaged" groups will be reduced towards the lower error rate of the majority or "advantaged" group. But in practice, equalizing error rates and similar notions may require artificially inflating error on easier-to-predict groups — withouit necessarily decreasing the error for the harder to predict groups — and this may be undesirable for a variety of reasons.

For example, consider the many social applications of machine learning in which most or even all of the targeted population is disadvantaged, such as predicting domestic situations in which children may be at risk of physical or emotional harm [7]. While we might be interested in ensuring that our predictions are roughly equally accurate across racial groups, income levels, or geographic location, if this can only be achieved by raising lower group error rates without lowering the error for any other population, then arguably we will have only worsened overall social welfare, since this is not a setting where we can argue that we are "taking from the rich and giving to the poor." Similar arguments can be made in other high-stakes applications, such as predictive modeling for medical care. In these settings it might be preferable to consider the alternative fairness criterion of achieving *minimax group error*, in which we seek not to equalize error rates, but to minimize the largest group error rate — that is, to make sure that *the worst-off group is as well-off as possible*.

Minimax group fairness, which was recently proposed by [19] in the context of classification, has the property that any model that achieves it *Pareto dominates* (at least weakly, and possibly strictly) an equalized-error model with respect to group error rates — that is, if g_i are the group error rates of the minimax

solution for each group i, and g' is the error rate for every group in a solution that equalizes group error rates, then $g' \geq g_i$ for all i. If one or more of these inequalities is strict, it constitutes a proof that equalized errors can only be achieved by deliberately inflating the error of one or more groups in the minimax solution. Said another way, one technique for finding an error optimal solution subject to an equality of error rates constraint is to first find a minimax solution, and then to artifically inflate the error rate on any group that does not saturate the minimax constraint — an "optimal algorithm" that makes plain the deficiencies of equal error solutions.

In contrast to approaches that learn separate models for each protected group (e.g. [10, 27]), the minimax approach has two key advantages:

- The minimax approach does not require that the groups of interest be disjoint, which is a requirement for the approach of learning a different model for each group. This allows for protecting groups defined by intersectional characteristics as in [16, 17], protecting (for example) not just groups defined by race or gender alone, but also by combinations of race and gender.
- The minimax approach does not require that protected attributes be given as inputs to the trained model. This can be extremely important in domains (like credit and insurance) in which using protected attributes as features for prediction is illegal.

Our primary contributions are as follows:

1. First, we propose two algorithms: the first finds a minimax group fair model from a given statistical class and the second navigates tradeoffs between a relaxed notion of minimax fairness and overall accuracy.
2. Second, we prove that both algorithms converge and are oracle efficient — meaning that they can be viewed as efficient reductions to the problem of unconstrained learning over the same class.
3. Third, we show how our framework can be easily extended to handle different types of error rates, such as false positive and negative rates.
4. Finally, we provide a thorough experimental analysis of our two algorithms under different prediction regimes. In this section, we focus on the following:
    - We start with a demonstration of the learning process of learning a fair predictor from the class of linear regression models. This setting matches our theory exactly, because weighted least squares regression is a convex problem, and so we really do have efficient subroutines for the unconstrained learning problem.
    - We conduct an exploration of the fairness vs. accuracy tradeoff for regression and highlight an example in which our minimax algorithms provide a substantial Pareto improvement over the equality of error rates notion.
    - Next, we give an account of the difficulties encountered when our oracle assumption fails in the classification case (because there are no efficient algorithms for minimizing 0/1 classification error, and so we must rely on learning heuristics).
    - With this in mind, we again explore tradeoff curves for the classification case and finish with another comparison in which we show marked improvement over equality of error rates.

1.1 Related Work

Technically, our work uses a similar approach to [1, 2], which also reduce a "fair" classification problem to a sequence of unconstrained cost-sensitive classification problems via the introduction of a zero-sum game. Crucial to this line of work is the connection between no regret learning and equilibrium convergence, originally due to [13].

Our work builds upon the notion of minimax fairness proposed by Martinez et al. [19] in the context of classification (we note that this is a classical notion of fairness with a long history in other areas, like scheduling, fair division, and clustering — see e.g. [15, 3, 22]). We note that their algorithm, Approximate Projection onto Star Sets (APStar), shares some similarities to ours, in that it is an iterative process which alternates between finding a model that minimizes risk and updating group weightings using a variant of gradient descent. However, although their algorithm is often convergent in practice, they lack a formal proof or counterexample to analyze the convergence properties. In contrast, we provide a proof that our algorithm converges under reasonable assumptions and access to an oracle. Additionally, APStar relies on knowledge of the base distributions of the samples (while our algorithms do not) and it does not provide the capability to relax the minimax constraint and explore an error vs. fairness tradeoff.

Martinez et al. [19] analyze only the classification setting, but we provide theory and perform experiments in both classification and regression settings. Because our meta-algorithm is easily extensible, we are able to generalize to non-disjoint groups and various error types (with an emphasis on false positive and false negative rates). The difficulty with non-convex statistical hypothesis classes and loss functions is briefly discussed in [19], and we build upon this observation, carefully differentiating between the machine learning settings in which our theoretical guarantees hold and those in which our algorithm must be viewed as a principled heuristic. Finally, we note that achieving minimax fairness over a large number of groups has been proposed by [18] as a technique for achieving fairness when protected group labels are not available. Our work relates to [18] in much the same way as it relates to [19], in that [18] is purely empirical, whereas we provide a formal analysis of our algorithm, as well as a version of the algorithm that allows us to relax the minimax constraint and explore performance tradeoffs.

2 Framework and Preliminaries

We consider pairs of dependent and independent variables $(x_i, y_i)_{i=1}^n$, where x_i is a vector of d features, divided into K groups $\{G_1, \ldots, G_K\}$. We choose a class H of models (which could be either classification or regression models), with loss function L, average population error

$$\epsilon(h) = \frac{1}{n} \sum_{i=1}^{n} L(h(x_i), y_i),$$

and average group loss

$$\epsilon_k(h) = \frac{1}{|G_k|} \sum_{(x,y) \in G_k} L(h(x), y)$$

for some $h \in H$. We also admit randomized models in this paper, which can be viewed as belonging to the set ΔH of probability distributions over H. We define population loss and group loss for a distribution over models as the expected loss over a random choice of model from the distribution.

First, in the pure minimax problem, our goal is to find a randomized model h^* that minimizes the maximum error rate over all groups:

$$h^* = \text{argmin}_{h \in \Delta H} \{ \max_{1 \leq k \leq K} \epsilon_k(h) \} \tag{1}$$

We let OPT_1 denote the value of the solution to the minimax problem: $\text{OPT}_1 = \max_k \epsilon_k(h^*)$. We say that a randomized model h is ϵ-approximately optimal for the minimax objective if:

$$\max_k \epsilon_k(h) \leq \text{OPT}_1 + \epsilon$$

We then describe an extension of the minimax problem: Given a target maximum group error bound $\gamma \geq \text{OPT}_1$, the goal is to find a randomized model that minimizes overall population error while staying below the specified maximum group error threshold:

$$\begin{array}{ll} \underset{h \in \Delta H}{\text{minimize}} & \epsilon(h) \\ \text{subject to} & \epsilon_k(h) \leq \gamma, k = 1, \ldots, K \end{array} \tag{2}$$

This extension has two desirable properties:

1. It has an objective function: there may in principle be many minimax optimal models that have different overall error rates. The constrained optimization problem defined above breaks ties so as to select the model with lowest overall error.

2. The constrained optimization problem allows us to trade off our maximum error bound γ with our overall error, rather than requiring us to find an exactly minimax optimal model. In some cases, this tradeoff may be worthwhile in that small increases in γ can lead to large decreases in overall error.

Given a maximum error bound γ, we write OPT_2 for the optimum value of Problem 2. We say that a randomized model h is an ϵ-approximate solution to the constrained optimization problem in 2 if $\epsilon(h) \leq \text{OPT}_2 + \epsilon$, *and* for all k, $\epsilon_k(h) \leq \gamma + \epsilon$.

In order to solve Problems 1 and 2, we pose the problems as two player games in Section 3. This will rely on several classical concepts and definitions from game theory, which we will expand upon in Section 4. We also define a weighted empirical risk minimization oracle over the class H ($WERM(H)$), which we will use as an efficient non-fair subroutine in our algorithms.

Definition 1 (Weighted Empirical Risk Minimization Oracle)**.** *A weighted empirical risk minimization oracle for a class H takes as input a set of n tuples $(x_i, y_i)_{i=1}^n$, a weighting of points w, and a loss function L, and finds a hypothesis $\hat{h} \in H$ that minimizes the weighted loss, i.e., $\hat{h} \in argmin_{h \in H} \sum_{i=1}^n w_i L(h(x_i), y_i)$.*

3 Two Player Game Formulation

Starting with Problem 1, we recast the optimization problem as a zero-sum game between a learner and a regulator in MinimaxFair. At each round t, there is a weighting over groups determined by the regulator. The learner (best) responds by computing model h_t to minimize the weighted prediction error. The regulator updates group weights using exponential weights with respect to group errors achieved by h_t. The learner's final model M is the uniform mixture of all h_t's produced. In the limit, M converges to a minimax solution with respect to group error. In particular, over T rounds, with $\eta_t \approx \frac{1}{\sqrt{t}}$, the empirical average of play forms a $\frac{1}{\sqrt{T}}$-approximate Nash equilibrium [13]. For simplicity, we describe our algorithms as if the groups G_i are disjoint, but we show in the Appendix how to formulate the problem for intersecting subgroups.

Algorithm 1: MinimaxFair

Input: $\{x_i, y_i\}_{i=1}^n$, adaptive learning rate η_t, populations G_k with relative sizes $p_k = \frac{|G_k|}{n}$, iteration count T, loss function L, model class H;

Let $\epsilon_k(h) := \frac{1}{|G_k|} \sum_{(x,y) \in G_k} L(x, y)$;

Initialize $\lambda_k := p_k \; \forall k$;

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

  Find $h_t := \text{argmin}_{h \in H} \sum_k \lambda_k \cdot \epsilon_k(h)$;

  Update each $\lambda_k := \lambda_k \cdot exp(\eta_t \cdot \epsilon_k(h_t))$;

end

Output: Uniform distribution over the set of models $h_1, ..., h_T$

As discussed in Section 2, not all minimax solutions achieve the same overall error. By setting an acceptable max group error γ, we can potentially lower overall error by solving a relaxed version of the

problem: Problem 2. Letting p_k be group proportions, and assuming that the groups are disjoint here for simplicity,[1] the Lagrangian dual function of Problem 2 is given by:

$$F(\lambda, h) = \epsilon(h) + \sum_k \lambda_k(\epsilon_k(h) - \gamma) = \sum_k ((p_k + \lambda_k)\epsilon_k(h) - \lambda_k\gamma)$$

We again cast this problem as a game in MinimaxFairRelaxed where the learner chooses h to minimize $\sum_k (p_k + \lambda_k)\epsilon_k(h)$, and the regulator adjusts λ through gradient ascent with gradient $\frac{\delta F}{\delta \lambda_k} = \epsilon_k(h) - \gamma$. As before, the empirical average of play converges to a Nash equilibrium, where an equilibrium corresponds to an optimal solution to the original constrained optimization problem.

Algorithm 2: MinimaxFairRelaxed

Input: $\{x_i, y_i\}_{i=1}^n$, adaptive learning rate η_t, populations G_j with relative sizes $p_j = \frac{|G_j|}{n}$, iteration count T, loss function L, model class H, maximal group error γ;

Let $\epsilon_j(h) := \frac{1}{|G_j|} \sum_{(x,y)\in G_j} L(x, y)$;

Initialize $\lambda_j := 0 \; \forall j$;

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

  Find $h_t := \text{argmin}_{h\in H} \sum_j (p_j + \lambda_j) \cdot \epsilon_j(h)$;

  Update each $\lambda_j := \max(\lambda_j + \eta_t \cdot (\epsilon_j(h_t) - \gamma), 0)$;

end

Output: Uniform distribution over the set of models $h_1, ..., h_T$

4 Theoretical Guarantees

In this section we derive the theoretical guarantees of our algorithms. To do so, we make two assumptions. The first is simply an assumption on how data points and losses are normalized:

Assumption 1. *We assume that the data are normalized so that each data point X_i lies in the unit ball (i.e. $||X_i|| \leq 1 \; \forall i$). We similarly assume that over this domain, our loss functions are bounded in $[0, 1]$. Note that these assumptions are without loss of generality for bounded loss functions, up to re-scaling. Without these assumptions, bounds on the maximum norm and loss would appear in our theorem statements.*

Next, we assume that learning problems (absent fairness constraints) can be solved by our algorithm as a subroutine. This is what allows us to bound the *additional* hardness of our fairness desiderata, on top of unconstrained learning:

Assumption 2. *The learner has access to a weighted empirical risk minimization oracle over the class H, $WERM(H)$, as specified in Definition 1.*

This assumption will be realized in practice whenever the objective is convex (for example, least squares linear regression). When the objective is not convex (for example, 0/1 classification error) we will employ heuristics in our experiments which are not in fact oracles, resulting in a gap between theory and practice that we investigate empirically.

4.1 MinimaxFair

Theorem 1. *After $T = \frac{\ln K}{2\epsilon^2}$ many rounds,* MinimaxFair *returns a randomized hypothesis that is an ϵ-optimal solution to Problem 1.*

[1] The derivation for overlapping groups is given in the Appendix.

Proof. From [6, 5, 14], under the conditions of Assumption 1, with loss function $L(\cdot)$, K groups, T time steps, and step size $\eta = \frac{8 \ln K}{T}$ the exponential weights update rule of the regulator yields regret:

$$\frac{R_T}{T} = \frac{1}{T}\left(\sum_{t=1}^{T} L(h_t(x), y) - min_{i \leq T} \sum_{i=1}^{T} L(h_i(x), y)\right) \leq \sqrt{\frac{\ln K}{2T}}$$

Plugging in $T = \frac{\ln K}{2\epsilon^2}$ gives $\frac{R_T}{T} \leq \sqrt{\frac{\ln K}{2\frac{\ln K}{2\epsilon^2}}} = \epsilon$.

As the learner plays a *best-response* strategy by calling $WERM(H)$, applying the following result of [13] completes the proof:

Theorem 2 (Freund and Schapire, 1996)**.** *Let $h_1, ...h_T$ be the learner's sequence of models and $w_1, ..., w_T$ be the regulator's sequence of weights. Let $\bar{h} \sum_{i=1}^{T} \frac{1}{T} h_t$ and $\bar{\lambda} = \frac{1}{T} \sum_{i=1}^{T} \lambda_i$. Then, if the regret of the regulator satisfies $\frac{R_G(T)}{T} \leq \epsilon$ and the learner best responds in each round,then $(\bar{h}, \bar{\lambda})$ is an ϵ-approximate solution.*

Therefore, the uniform distribution over $h_1, ..., h_T$ obtained by the learner in MinimaxFair is an ϵ-optimal solution to Problem 1, as desired. We note that this requires one call to $WERM(H)$ for each of the T rounds.

□

4.2 MinimaxFairRelaxed

Theorem 3. *After $T = \frac{1}{4\epsilon^2}\left(\frac{1}{\epsilon^2} + 2K\right)^2$ many rounds,* MinimaxFairRelaxed *returns a randomized hypothesis that is an ϵ-optimal solution to Problem 2.*

Proof. In MinimaxFairRelaxed the regulator plays a different strategy: Online Gradient Descent. We specify the regret bound from [28] below. Note that in the following definition, F is the set containing all values of λ (the vector updated through the gradient descent procedure), and the size of the set F is denoted as:

$$||F|| = \max_{x,y \in F} d(x, y) = \max_{x,y \in F} ||x - y||$$

In our analysis, we compute our regret to the best vector of weights such that $||\lambda|| \leq \frac{1}{\epsilon} = ||F||$. We write $||\nabla c||$ for the norm of the gradients that we feed to gradient descent. As our losses are bounded by $[0, 1]$ by Assumption 1, we have that $||\nabla c||^2 = \sum_{k=1}^{K} (\epsilon_k(h_t) - \gamma)^2 \leq K$. Then, from [28], with $\eta_t = t^{-\frac{1}{2}}$,

$$R_T \leq \frac{||F||^2 \sqrt{T}}{2} + \left(\sqrt{T} - \frac{1}{2}\right) ||\nabla c||^2$$

Plugging in the specification for our problem, we have

$$\begin{aligned} \frac{R_T}{T} &\leq \frac{1}{T}\left(\frac{\frac{1}{\epsilon}^2 \sqrt{T}}{2} + K\left(\sqrt{T} - \frac{1}{2}\right)(1 - \gamma)^2\right) \\ &< \frac{1}{T}\left(\frac{\sqrt{T}}{2\epsilon^2} + K\sqrt{T}\right) = \frac{\frac{1}{\epsilon}^2 + 2K}{2\sqrt{T}} \end{aligned}$$

Substituting $T = \frac{1}{4\epsilon^2}\left(\frac{1}{\epsilon^2} + 2K\right)^2$ yields

$$\frac{R_T}{T} \leq \frac{\frac{1}{\epsilon}^2 + 2K}{2\sqrt{\frac{1}{4\epsilon^2}\left(\frac{1}{\epsilon^2} + 2K\right)^2}} = \epsilon$$

Therefore, Theorem 2 guarantees that the value of the objective is not more than ϵ away from OPT_2, using the notation of Section 2.

Finally, we show that the learner cannot choose a model that violates a constraint by more than ϵ. Suppose the learner chose a randomized strategy h^* such that $\epsilon_k(h^*) > \gamma + \epsilon$ for some k. Then, the regulator may set $\lambda_k = \frac{1}{\epsilon}$ and $\lambda_j = 0$ for all $k \neq j$, yielding

$$\max_{\lambda}(F(\lambda, h^*) > \epsilon(h) + \frac{1}{\epsilon}\epsilon = \epsilon(h^*) + 1$$

However, because $\epsilon(h) \leq 1$ by Assumption 1, the learner is better off selecting h^0 such that $\epsilon(h^0) = \gamma$ for all groups, even if $\epsilon(h^0) = 1$. Then, $\max_\lambda(F(\lambda, h^0) = \epsilon(h^0) \leq 1$.

Thus, an ϵ-approximate equilibrium distribution h for the learner must satisfy $\epsilon_k(h) \leq \gamma + \epsilon$ to minimize the value of $\max_\lambda(F(\lambda, h)$. Therefore, we have shown that in $T = \frac{1}{4\epsilon^2}\left(\frac{1}{\epsilon^2} + 2K\right)^2$ rounds – with one call to $WERM(H)$ per round – MinimaxFairRelaxed always outputs a model h such that $\epsilon(h) \leq OPT_2 + \epsilon$ and for all k, $\epsilon_k(h) \leq \gamma + \epsilon$, or an ϵ-optimal solution to Problem 2.

□

5 Extension to False Positive and False Negative Rates

A strength of our framework is its generality. With minor alterations, MinimaxFairRelaxed can also be used to bound false negative or false positive group error in classification settings while minimizing overall population error. This is particularly useful because a trivial minimax false positive or false negative rate of zero can always be achieved by invoking a constant classifier. Therefore, it does not make sense to solve the minimax problem for false positive or false negative rates, but it does make sense to solve the constrained optimization problem.

To extend MinimaxFairRelaxed to bound false positive rates, we again consider a setting in which we are given groups $\{G_j\}_{j=1}^J$ containing points $(x_i, y_i)_{i=1}^n$. We will want to bound the error, here denoted by $\epsilon(h, (x_i, y_i) = |h(x_i) - y_i|$, on the parts of each group that contain true negatives: $G_j^0 = \{(x, y) : (x, y) \in G_j \text{ and } y = 0\}$. We want to minimize overall population error while keeping all group false positive rates below γ, and our adapted constrained optimization problem is:

$$\begin{aligned} \underset{h \in H}{\text{minimize}} \quad & \frac{1}{n}\sum_{i=1}^{n} \epsilon(h, (x_i, y_i)) \\ \text{subject to} \quad & \frac{1}{|G_j^0|}\sum_{(x_i,y_i)\in G_j^0} \epsilon(h, (x_i, y_i)) \leq \gamma, j = 1, \ldots, J \end{aligned} \tag{3}$$

The Lagrangian dual for Problem 3 is:

$$\begin{aligned} F(\lambda, h) &= \frac{1}{n}\sum_i \epsilon(h, (x_i, y_i)) + \sum_j \lambda_j \left(\frac{1}{|G_j^0|}\sum_{(x_i,y_i)\in G_j^0} \epsilon(h, (x_i, y_i)) - \gamma\right) \\ &= \sum_i \epsilon(h, (x_i, y_i)) \left(\frac{1}{n} + \sum_j \frac{\lambda_j}{|G_j^0|}\mathbb{I}\{(x_i, y_i) \in G_j^0\}\right) - \sum_j \lambda_j \gamma \\ &= \sum_i w_i \epsilon(h, (x_i, y_i)) - \sum_j \lambda_j \gamma \end{aligned}$$

Where $w_i = \frac{1}{n} + \sum_j \frac{\lambda_j}{|G_j^0|}\mathbb{I}\{(x_i, y_i) \in G_j^0\}$ and

$$\frac{\delta F}{\delta \lambda_j} = \frac{1}{|G_j^0|}\sum_i \mathbb{I}\{(x_i, y_i) \in G_j^0\}\epsilon(h, (x_i, y_i)) - \gamma$$

We can then use the sample weights w in the learner's step of the constrained optimization problem. In particular, at each round t, the learner will find $h_t := \text{argmin}_{h \in H} \sum_i w_i \cdot \epsilon_i L(x_i, y_i)$ and the regulator will update the λ vector with

$$\lambda_j := \max\left(\lambda_j + \eta_t \cdot \left(\frac{1}{|G_j^0|} \sum_i \mathbb{I}\{(x_i, y_i) \in G_j^0\} \epsilon(h, (x_i, y_i)) - \gamma, 0\right)\right.$$

6 Experimental Results

We experiment with our algorithms in both regression and classification domains using real data sets. A brief summary of our findings and contributions is:

- Our minimax algorithm often admits solutions with significant Pareto improvements over equality of errors in practice.
- Unlike similar equal error algorithms, our minimax and relaxed algorithms use only non-negative sample weights, increasing performance for both regression and classification via access to better subroutines (because non-negative weights preserve the convexity of convex loss functions).
- We illustrate the tradeoff between overall error and fairness by explicitly tracing a Pareto curve of all possible models between the population error minimizing model and the model achieving minimax group fairness. We extend this to the case of false positive or false negative group fairness by using our relaxed algorithm over a range of γ values.

6.1 Methodology and Data

6.1.1 Data

We begin with a short description of each of the data sets used in our experiments:

- **Communities and Crime [21, 23, 24, 25, 26]:** Communities within the US. The data combines socio-economic data from the 1990 US Census, law enforcement data from the 1990 US LEMAS survey, and crime data from the 1995 FBI UCR.
- **Bike [12, 11, 9]:** Public bikes rented at each hour in Seoul Bike haring System with the corresponding Weather data and Holidays information.
- **COMPAS [4]:** Arrest data from Broward County, Florida, originally compiled by ProPublica.
- **Marketing [20, 9]:** Data related with direct marketing campaigns (phone calls) of a Portuguese banking institution.
- **Student [8, 9]:** Student achievement in secondary education of two Portuguese schools.

The table below outlines the data sets used in our experiments. For all data sets, categorical features were converted into one-hot encoded vectors and group labels were included in the feature set.

6.1.2 Experimental Methodology

For each of our data sets, we performed a similar set of experiments.

First, we ran MinimaxFair to find the model achieving minimax group fairness, and then we used our minimax solution to perform multiple runs of MinimaxFairRelaxed to find and plot an error versus fairness tradeoff curve. The specific process is:

Dataset	n	d	Label	Group	Task
Communities and Crime	1594	133	violent crimes per pop.	race	regression
Bike	8760	19	rented bikes normalized	season	regression
COMPAS	4904	9	two year recidivism	race, sex	classification
Marketing	45211	48	subscribes to term deposit	job	classification
Student	395	75	final grade	sex	classification

- First, we use our minimax solution from MinimaxFair to determine feasible lower and upper bounds for maximum group error for the dataset. The group error of the minimax solution is a lower bound on maximum group error, and the group error associated with the model from the first round of the game–which minimized population error–gives us an upper bound on acceptable maximum group error.
- Then, we use MinimaxFairRelaxed across many different values for γ, or maximum allowed group errors. MinimaxFairRelaxed finds a mixture model that minimizes expected population error subject to having expected maximum group error at most γ.
- Finally, we overlay the trajectory plot produced from each run of MinimaxFairRelaxed to trace a Pareto curve denoting our error versus fairness tradeoff.

In the false positive/negative case, we skip running MinimaxFair and directly use MinimaxFairRelaxed over a range of γ values above 0. This is because the minimax solution for false positive (or negative) group errors will always be zero, as a constant classifier that blindly predicts 'False' (or 'True') on all inputs.

Since each of our algorithms produces solutions in the form of probabilistic mixture models with error reported in expectation, every linear combination of these models represents another mixture model. Further, it means that every point on our Pareto curve for population error vs. maximum group error–including those falling on the line between two models–can be achieved by some actual mixture model in our class.

6.1.3 Plotting on training data

In this paper, all reported errors are on the *training* data. This ensures that our plots most accurately illustrate the behavior of our algorithms and allows us to demonstrate that our theoretical guarantees are being met in practice. Generalization issues are identical in our setting as they are in standard (unconstrained) learning problems, and can be handled with the same set of techniques, so we choose to instead focus on evaluating the novel aspects of our algorithms.

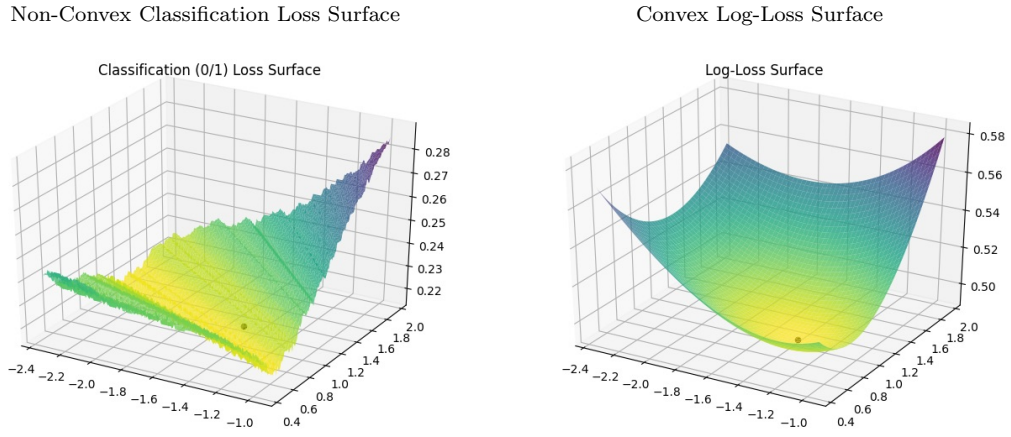
6.1.4 Regression: Finding Exact Solutions Efficiently

The solution to a weighted linear regression is *guaranteed* to be the regression function that minimizes the mean squared error across the dataset, making linear regression a pure demonstration of our theory in Sections 2 and 3. We note that to solve weighted linear regressions *efficiently*, sample weights must be non-negative, because negative weights make squared error non-convex. Our minimax and relaxed algorithms satisfy this property, giving us access to exact solutions in regression settings. In contrast, similar algorithms (e.g. those of [1, 16]) for equalizing group error rates require negative sample weights, and cannot use linear regression for exact weighted error minimization in the same way.

6.1.5 Classification: Non-convexity of 0/1 Loss

As opposed to mean squared error of linear regression, 0/1 classification loss is non-convex, so minimizing 0/1 loss locally does not imply global minimization. As a result, we cannot efficiently find solutions that

Figure 1: Comparing Convexity of 0/1 Loss vs. Log-Loss



minimize classification loss in practice. Instead, we rely on convex surrogate loss functions such as log-loss– the training objective for logistic regression–which are designed to approximate classification loss. Note that lack of exact solutions for classification loss violates Assumption 2, so the theoretical guarantees of Sections 2 and 3 may fail to hold. Our algorithm should be viewed as a principled heuristic in these settings. Note, when using logistic regression with MinimaxFair, we update sample weights based on the *log-loss* so sample weights correspond to the training objective, but we report final errors in our plots with respect to 0/1 loss.

In Figure 1, we illustrate the difference between convex and non-convex loss surfaces with an example. The top image shows the non-convex 0/1 loss surface corresponding to two-dimensional linear models over a synthetic dataset, and the bottom image shows the (convex) log-loss surface over the same space.

6.1.6 Paired Regression Classifier

Due to the heuristic nature of finding models to minimize 0/1 loss, we also experiment with a second classification model: the paired regression classifier (PRC) used by [1, 16], detailed below. For the sake of our experiments, the PRC serves as an alternative model class in cases for which logistic regression behaves strangely. Another important benefit of the PRC is that it accepts negative sample weights, a necessary feature for use with our equal error rates algorithm which we use to benchmark our minimax algorithm.

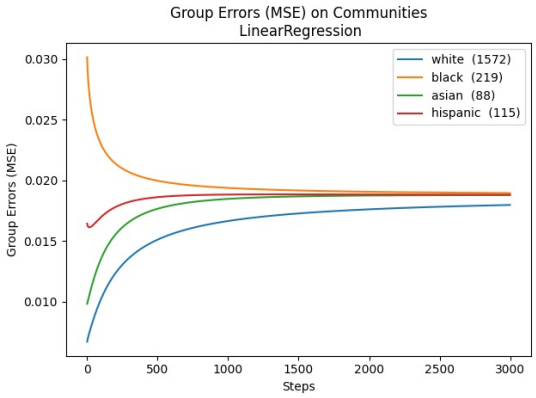
Definition 2 (Paired Regression Classifier)**.** *The paired regression classifier, used in [16, 1], allows us to use linear regression for classification and works in the following way. We form two weight vectors, z^0 and z^1, where z_i^k corresponds to the penalty assigned to sample i in the event that it is labeled k. For the correct labeling of x_i, the penalty is 0. For the incorrect labeling, the penalty is the current sample weight of the point, w_i. We fit two linear regression models h^0 and h^1 to predict z^0 and z^1, respectively, on all samples. Then, given a new point x, we calculate $h^0(x)$ and $h^1(x)$ and output $h(x) = argmin_{k\in\{0,1\}} h^k(x)$.*

6.2 Linear Regression Experiments

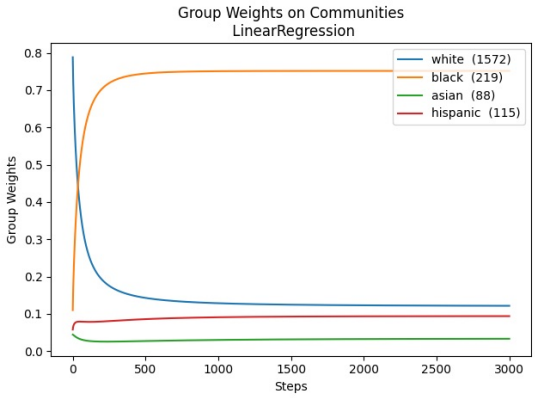
As explained above, linear regression is an exact setting for demonstrating the properties of our algorithms, as we can solve weighted linear regression problems exactly and efficiently in practice. Our first results, shown in Figure 2, illustrate the behavior of MinimaxFair on the *Communities* dataset. The first and second plot illustrate the errors and weights of the various groups. The third plot denotes the "trajectory" of our mixture model over time, showing how we update our mixture from our initial population-error minimizing model (labeled by a large pink dot) to our final model achieving minimax group fairness (colored yellow).

Figure 2: Minimax Solution on Communities Dataset

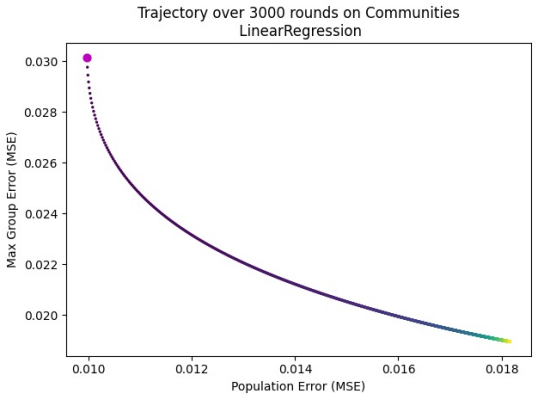
Group Errors



Group Weights



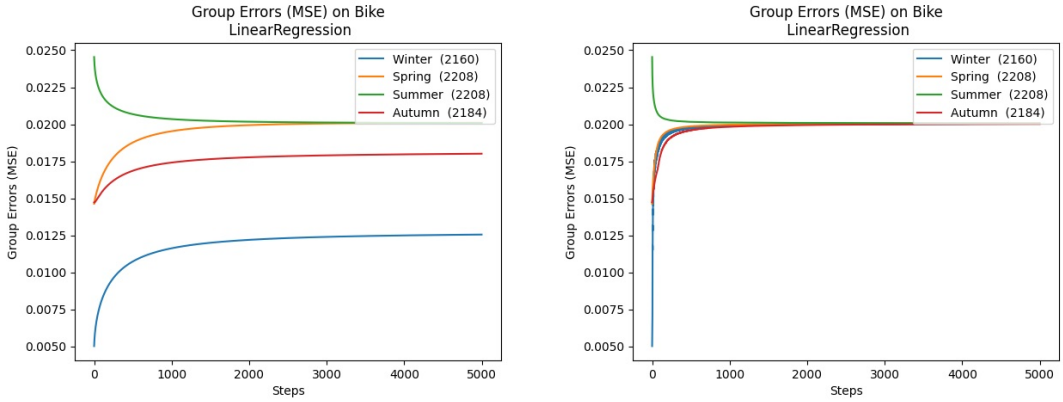
Learning Trajectory



Looking at the first two plots, we see that the plurality black communities–denoted by an orange line–begin with the largest error of any group with an MSE of 0.3, while the plurality white communities–denoted by a blue line–have the lowest MSE at a value 0.005. Our algorithm responds to this disparity by up-weighting samples representing plurality black communities and by down-weighting those representing plurality white communities. After many rounds, an approximate minimax solution for group-fairness is reached, with maximum group error has value slightly under 0.02. Moreover, we observe that our minimax solution nearly equalized errors on three of our groups, with one group (white communities) having error slightly below the minimax value.

Near equalization of the highest group errors in minimax solutions as seen in this example is frequent and well-explained. In any error optimal solution, the only way to decrease the error of one group is to increase the error of another. Hence, whenever the loss landscape is continuous over our class of models (as it is in our case, because we allow for distributions over classifiers), minimax optimal solutions will always equalize the error of at least two groups. But as we see in our examples, it does not require equalizing error across all groups when there are more than two.

Figure 3: Minimax (left) vs. Equal Error (right) Solutions on Bike Dataset



6.2.1 Comparing Minimax to Equality

Next, we provide a comparison between our minimax algorithm and the equal error rates formulation of [1]. We note that, while linear regression is an excellent fit for our minimax algorithm, it poses difficulties in the equal error rates framework. In particular, similar primal/dual algorithms for equalizing error rates across groups require the use of negative sample weights, because the dual solution to a linear program with equality constraints generically requires negative variables. Negative sample weights destroy convexity for objective functions like squared error that are convex with non-negative weights. For this reason, we can only use our equal-error algorithm in a meaningful way for linear regression in settings in which the sample weights (by luck) never become negative. On the Bike dataset, we meet this condition, and are therefore able to provide a meaningful comparison between the solutions produced by the two algorithms which is illustrated in Figure 3. We observe that the only difference between the two solutions, is that error in Winter and Autumn increases to the minimax value when we move from minimax to equality. This highlights an important point: enforcing equality of group errors may significantly hinder our performance on members of low-error groups, without providing benefit to those of higher-error groups. Though the bike dataset itself is not naturally fairness-sensitive, the properties illustrated in this example can occur in any dataset.

6.2.2 Relaxing Fairness Constraints

Finally, we investigate the "cost" of fairness with respect to overall accuracy. To do this, we trace the Pareto curve of population error vs. group fairness as described in Section 6.1.2. Figure 4 shows an overlay of many trajectories and the associated Pareto curve (denoted in red, dashed line) on the *Communities* dataset. We observe that the relationship between expected population error and maximum group error is decreasing and convex, illustrating a clear tradeoff between the two objectives. (The point labeled $\gamma = 0.0$ corresponds to the minimax solution for MinimaxFair, where $\gamma = \gamma^*$.)

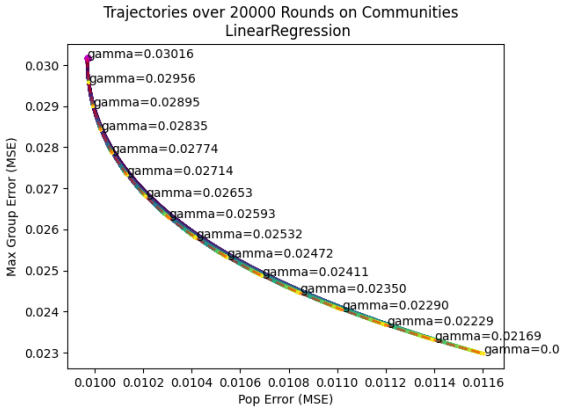
6.3 Classification Experiments

6.3.1 Comparing Minimax to Equality

First, we exhibit several experiments showing marked improvement by our meta-algorithm compared to the related equal error rates algorithm of [1]. In order to handle negative sample weights, we use solely the PRC heuristic for the equal errors algorithm. For the minimax algorithm, which requires only non-negative sample weights, we can freely choose between logistic regression and PRC for learning each rounds models, and we include plots for both for reference.

In Figure 5 we compare the performance of MinimaxFair with both logistic regression and PRC to its

Figure 4: Fairness Accuracy Tradeoff (Linear Regression)



equal error variant using the PRC on the Marketing dataset, which divides instances into 12 employment-based groups. Comparing the minimax algorithm with logistic regression to the equal error algorithm (which is forced to use the PRC), we see that our minimax solution strongly Pareto dominates the equal error one. The minimax solution has maximum group error of 0.22 with the errors of other groups varying below[2] that and taking on values as low as 0.07. Alternatively, in the equal-error solution, we see that group errors equalize at a value around 0.29, meaning that we inflated the error on not only the low error groups but also on the highest error group. Looking at the minimax solution using the PRC, we see that the final solution reached is nearly identical to that of the equal-error solution, with the group errors of *all* groups inflated to equalize at a value near 0.29, indicating that the poor performance of the equal error algorithm in this setting may be due to failures of the PRC as a classification heuristic. Thus, we attribute the improved performance of the minimax algorithm in this setting to its ability to access a richer set of learning algorithms, specifically those requiring non-negative sample weights.

In Figure 6, we perform a similar comparison over ProPublica's COMPAS Recidivism Risk Score Data and Analysis dataset [4]. We examine error in predictions of criminal recidivism, and see that using either logistic regression or PRC for our minimax algorithm leads to a solution Pareto dominating that for the equal errors case, though to a lesser degree than before.

6.3.2 Relaxation and Pareto Curves

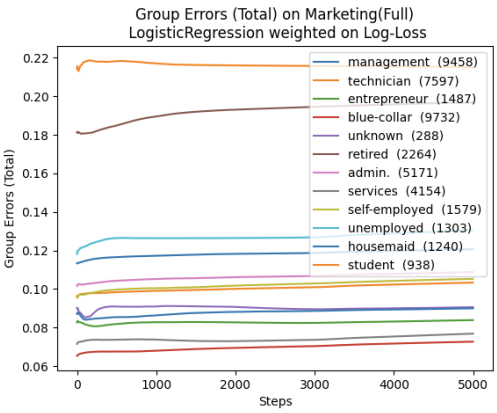
With the potential issues of non-convexity in mind, we move to an experimental analysis of the error versus fairness tradeoff curves produced by MinimaxFairRelaxed in the classification setting. In Figure 7, we predict whether or not an individual will subscribe a term deposit in a Portuguese bank using the Marketing dataset of [20, 9]. In this experiment, we train on log-loss, using it as the error metric for the updates of both the learner and regulator. As the theory dictates, by convexity of the log-loss, we observe an excellent convex error fairness tradeoff across different values of γ. When we examine the resulting tradeoff of the model with respect to classification loss – shown in the lower part of the figure – we see that the shape is similar, indicating that, for this dataset, log-loss is a good surrogate for 0/1 loss, and our fairness guarantees may be realized in practice.

Turning to a second dataset, the Student dataset of [8, 9], we perform the same analysis, modeling student performance in a Portuguese elementary school. We observe that the log-loss trajectory shown at the top of Figure 8 is once again convex, yet the behavior of the classification loss trajectory is more sporadic and often the population error and maximum group error decrease simultaneously. This behavior indicates that

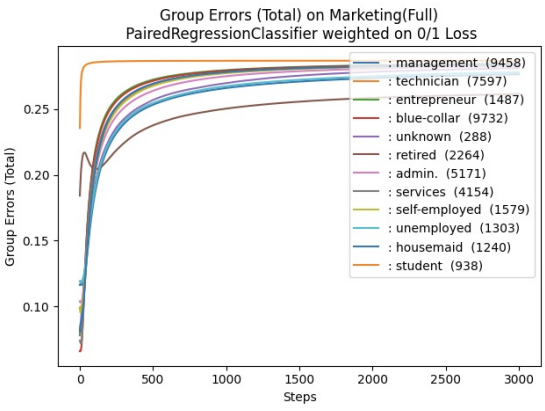
[2]Note that during training we find the minimax solution with respect to the log-loss, but we plot the 0/1 classification error of the resulting model. This is why we do not see an equalization of error between the two groups with the highest error rates at the minimax solution.

Figure 5: Minimax vs. Equal Errors on Marketing Dataset

Minimax with Logistic Regression



Minimax with PRC



Equal Errors with PRC

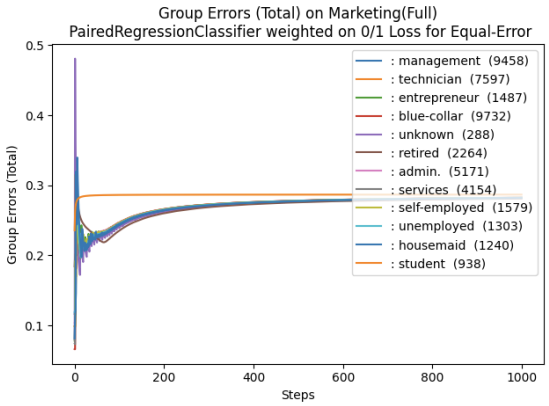
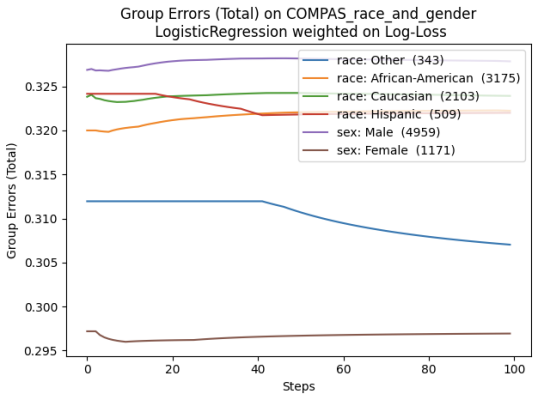
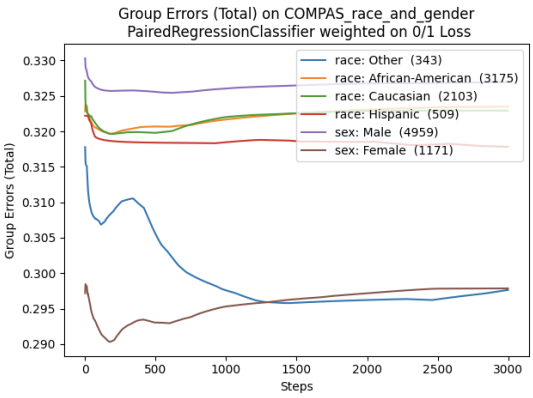


Figure 6: Minimax vs. Equal Errors on COMPAS Dataset

Minimax with Logistic Regression



Minimax with PRC



Equal Errors with PRC

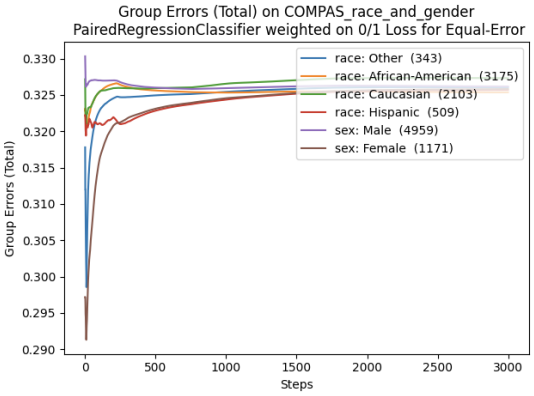
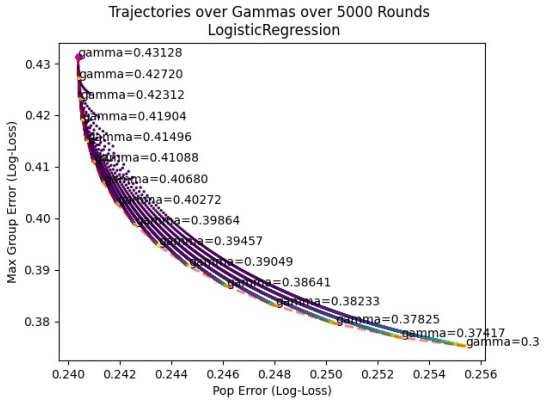


Figure 7: Fairness Accuracy Tradeoff on Marketing Dataset for Log-Loss (left) and Classification Loss (right)



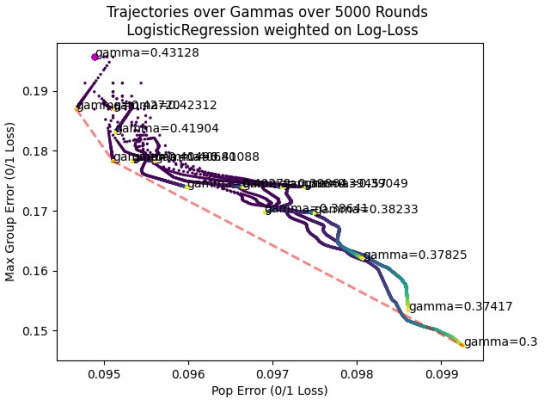
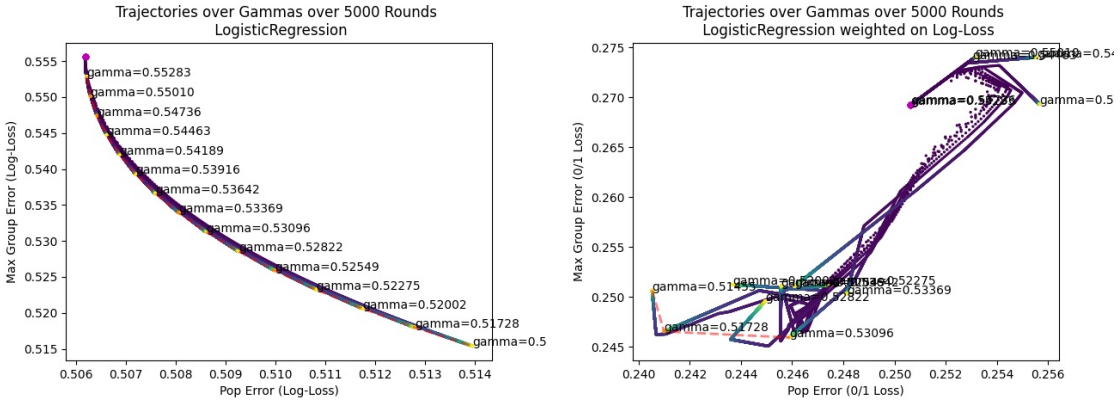


Figure 8: Fairness Accuracy Tradeoff on Student Dataset for Log-Loss (left) and Classification Loss (right)



the weighted ERM solutions we are finding with respect to classification error are only heuristic.

6.3.3 False Positive and False Negative Rates

In some contexts, we may care more about false positive or false negative rates than overall error rates. For example, when predicting criminal recidivism (using, for example, the popular *COMPAS* dataset), we may consider a false positive to be more harmful to an individual than a false negative, as it can lead to mistaken incarceration. As mentioned previously, one can trivially achieve a false positive rate of zero using a constant classifier that blindly predicts 'False' on all inputs (minimizing false negatives is analogous). Therefore, achieving minimax group false positive or false negative rates alone is uninteresting. Instead, we care about the tradeoff between overall error and group false positive or false negative rates.

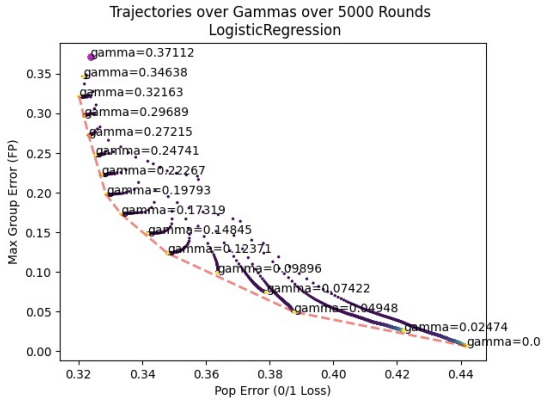
One distinct advantage of our relaxed algorithm is the ability to explore this tradeoff. Unlike the minimax algorithm, which ignores population error completely, the relaxed algorithm prioritizes minimizing population error subject to meeting its fairness constraints. Because of this, we can apply it in the false positive/negative case and examine the tradeoff between lowering group false positive/negative rates and consequently increasing population error.

In Figure 9 we examine this tradeoff on the *COMPAS* dataset with reference to false positive rates. When population error is minimized (indicated with a pink dot), the maximum false positive rate of any group is around 0.37. As we decrease γ, we see a gradual increase in population error tracing a nearly-convex tradeoff curve. When the maximum group false positive rate reaches 0, the population error achieved by our algorithm is 0.44. Importantly, we are not restricted to picking from these extremal points on the curve. Suppose we believed that a maximum group false positive rate of 0.05 was acceptable. Our tradeoff curve shows that we can meet this constraint while achieving a population error of only 0.39.

7 Conclusion

We have provided a provably convergent, practical algorithm for solving minimax group fairness problems and error minimization problems subject to group-dependent error upper bound constraints. Our meta-algorithm supports any statistical model class making use of sample weights, and as the theory suggests, performance is excellent when we can exactly solve weighted empirical risk minimization problems (as is the case with linear regression and other problems with convex objectives). We provide a thorough discussion of instances in which this assumption is infeasible, markedly when a convex surrogate loss function is used in place of classification loss. In these instances, our algorithm must be viewed as a principled heuristic. Finally, in regression and classification settings, we demonstrate that our algorithms for optimizing minimax group error results in overall error that is no worse than the error that can be obtained when attempting

Figure 9: Fairness Accuracy Tradeoff for FP on COMPAS



to equalize error across groups and can be markedly better for some groups compared to the equal error solution. In high stakes settings, this Pareto improvement may be highly desirable, in that it avoids harming any group more than is necessary to reduce the error of the highest error group.

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] Alekh Agarwal, Miroslav Dudik, and Zhiwei Steven Wu. Fair regression: Quantitative definitions and reduction-based algorithms. volume 97 of *Proceedings of Machine Learning Research*, pages 120–129, Long Beach, California, USA, 09–15 Jun 2019. PMLR. URL `http://proceedings.mlr.press/v97/agarwal19d.html`.

[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] *COMPAS Recidivism Risk Score Data and Analysis*. Broward County Clerk's Office, Broward County Sherrif's Office, Florida Department of Corrections, ProPublica, September 2020. URL `https://www.propublica.org/datastore/dataset/compas-recidivism-risk-score-data-and-analysis`.

[5] Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, 2006. doi: 10.1017/CBO9780511546921.

[6] Nicolò Cesa-Bianchi and Gábor Lugosi. On prediction of individual sequences. *Ann. Statist.*, 27(6): 1865–1895, 12 1999. doi: 10.1214/aos/1017939242. URL `https://doi.org/10.1214/aos/1017939242`.

[7] Alexandra Chouldechova, Diana Benavides-Prado, Oleksandr Fialko, and Rhema Vaithianathan. A case study of algorithm-assisted decision making in child maltreatment hotline screening decisions. In *Conference on Fairness, Accountability and Transparency*, pages 134–148, 2018.

[8] Paulo Cortez and Alice Silva. Using data mining to predict secondary school student performance. *EUROSIS*, 01 2008.

[9] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

[10] 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.

[11] Sathishkumar V E and Yongyun Cho. A rule-based model for Seoul bike sharing demand prediction using weather data. *European Journal of Remote Sensing*, pages 1–18, 2020. doi: 10.1080/22797254.2020.1725789.

[12] Sathishkumar V E, Jangwoo Park, and Yongyun Cho. Using data mining techniques for bike sharing demand prediction in metropolitan city. *Computer Communications*, 153:353 – 366, 2020. ISSN 0140-3664. doi: https://doi.org/10.1016/j.comcom.2020.02.007. URL `http://www.sciencedirect.com/science/article/pii/S0140366419318997`.

[13] Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, 1996.

[14] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139, August 1997. ISSN 0022-0000. doi: 10.1006/jcss.1997.1504. URL `https://doi.org/10.1006/jcss.1997.1504`.

[15] 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.

[16] Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *Proceedings of the 35th International Conference on Machine Learning*. Stockholm, Sweden, PMLR 80, 2018.

[17] 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.

[18] 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.

[19] 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.

[20] S. Moro, P. Cortez, and P. Rita. A data-driven approach to predict the success of bank telemarketing. *Decis. Support Syst.*, 62:22–31, 2014.

[21] Michael Redmond and Alok Baveja. A data-driven software tool for enabling cooperative information sharing among police departments, 2002. ISSN 0377-2217. URL `http://www.sciencedirect.com/science/article/pii/S0377221701002648`.

[22] 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.

[23] Bureau of the Census U. S. Department of Commerce. Census of population and housing 1990 United States: Summary tape file 1a & 3a (computer files), 1900.

[24] Bureau Of The Census Producer U.S. Department Of Commerce, 1992.

[25] Bureau of Justice Statistics U.S. Department of Justice, 1992.

[26] Federal Bureau of Investigation U.S. Department of Justice. Crime in the United States (computer file), 1995.

[27] 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.

[28] Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning*. Washington, DC, 2003.

A Update Rules in MinimaxFairRelaxed for Overlapping Groups

In the original presentation of MinimaxFairRelaxed, we assumed that the groups were disjoint for ease of presentation. Here we provide a more general derivation of the update rules for the learner and regulator when groups are allowed to be overlapping. As before, we are given groups $G_j{}_{j=1}^{J}$ (not necessarily disjoint) containing points $(x_i, y_i)_{i=1}^n$, and we want to minimize population error while bounding the error of each group by γ. So, our constrained optimization problem is:

$$\begin{aligned} &\underset{h \in H}{\text{minimize}} && \epsilon(h) \\ &\text{subject to} && \epsilon_j(h) \leq \gamma, j = 1, \ldots, J \end{aligned} \tag{4}$$

Then, the corresponding Lagrangian dual is:

$$\begin{aligned} F(\lambda, h) &= \epsilon(h) + \sum_j \lambda_j(\epsilon_j(h) - \gamma) \\ &= \frac{1}{n}\sum_{i=1}^n L(h(x_i), y_i) + \sum_j \lambda_j \left(\frac{1}{|G_j|} \sum_{i=1}^n L(h(x_i), y_i) \mathbb{I}\{(x_i, y_i) \in G_j\} - \gamma \right) \\ &= \sum_{i=1}^n \left(\frac{1}{n} L(h(x_i), y_i)) + \sum_{j=1}^J \frac{\lambda_j}{|G_j|} L(h(x_i), y_i))) \mathbb{I}\{(x_i, y_i) \in G_j\} \right) - \sum_{j=1}^J \lambda_j \gamma \\ &= \sum_{i=1}^n L(h(x_i), y_i) \left(\frac{1}{n} + \sum_{j=1}^J \frac{\lambda_j}{|G_j|} \mathbb{I}\{(x_i, y_i) \in G_j\} \right) - \sum_{j=1}^J \lambda_j \gamma \\ &= \sum_{i=1}^n w_i L(h(x_i), y_i) - \sum_{j=1}^J \lambda_j \gamma \end{aligned}$$

Where $w_i = \frac{1}{n} + \sum_j \frac{\lambda_j}{|G_j|} \mathbb{I}\{(x_i, y_i) \in G_j\}$ and

$$\frac{\delta F}{\delta \lambda_j} = \frac{1}{|G_j|} \sum_i \mathbb{I}\{(x_i, y_i) \in G_j\} L(h(x_i), y_i)) - \gamma$$

We can then use the sample weights w in the learner's step of the constrained optimization problem. In particular, at each round t, the learner will find $h_t := \text{argmin}_{h \in H} \sum_i w_i \cdot \epsilon_i L(x_i, y_i)$ and the regulator will update the λ vector with

$$\lambda_j := \max\left(\lambda_j + \eta_t \cdot \left(\frac{1}{|G_j|} \sum_i \mathbb{I}\{(x_i, y_i) \in G_j\} L(h(x_i), y_i)\right) - \gamma, 0\right)$$