Logistic Regression Penalizing Demograpic Disparities

Jiahui Zhao, Praveen Nair, Qunli Li, Yiteng Zhang

March 20, 2023

Abstract

Fairness is a rapidly growing area of concern in machine learning literature, with many arguing for algorithms to consider possible biases based on sensitive attributes (such as gender, race, nationality, etc.) when designing algorithms that could affect important decisions. One subfield of this literature seeks to develop penalties for unfairness as part of the objective functions of learning methods, such that model creators can control the tradeoff of optimizing accuracy versus maintaining group fairness. In this paper, we extend previous methods by Bechavod and Ligett (2018) to create convex penalties for fairness metrics that can be added to logistic regression in order to create more fair models. We develop a penalty to incentivize demographic parity which is able to significantly reduce disparities in a test set while preserving accuracy. We also develop a penalty that allows us to weight accuracy differently for different demographic groups, which indeed does improve accuracy for both majority and minority groups.

1 Introduction

1.1 Motivation

As machine learning algorithms are tasked with making increasingly important decisions that can affect people's everyday lives, an increasing amount of focus is being given to possible concerns with fairness and algorithmic bias in these algorithms [1, 2, 3]. Recent high-profile examples of these concerns include ProPublica's investigation of COMPAS, an algorithm by Northpointe for predicting recidivism in criminal defendants, and the discovery by Obermeyer et al. (2019) of racial biases in an algorithm for identifying high-risk medical patients in ways that would disproportionately deny Black patients care [4, 5].

There are a wide range of strategies that are used for mitigating bias in machine learning algorithms, and they can be split up into:

- pre-processing approaches, such as re-weighing and label flipping [6]
- in-processing approaches, which add penalizers to loss functions to incentivize algorithms towards fairer outcomes [7]
- post-processing approaches, which alter a model's predictions after they are made to align with desired fairness constraints [8]

For this course, in-processing approaches are the most applicable, since they allow us to modify the optimization problems inherent in learning techniques by designing additional terms to place in the objective function. Performing in-processing also allows us to tune hyperparameters that weight these different notions of fairness differently.

1.2 Previous Work

In both the philosophical sense and in the algorithmic sense, there are many varying ideas of what "fairness" might entail, each of which capture different notions of what ideal decision-making systems should and should not consider with respect to the individuals they make decisions on [9]. This is true even when we restrict our focus to group fairness, where we try to remain unbiased with respect to gender, race, and other protected attributes; there are still other notions of what it might mean to be fair, such as individual fairness [10]. This is made even more complicated by the fact that many

different fairness metrics, which may each sound reasonable on their own, are mutually exclusive with either very weak assumptions or none at all [11, 12].

There have been many previous efforts to introduce fairness to optimization problems, with a variety of approaches. Goel et al. (2018) modify logistic regression by formulating convex terms relating to historical bias present in the dataset, and to bias present in the classifier [13]. This approach yields classifiers that are Pareto optimal relative to other classifiers, in that they disadvantage any individuals only through changes that are more advantageous to others; the authors find that this method "achieves non-discrimination without significant loss in accuracy."

Zafar et al. (2017) propose a convex constraint on logistic regression that penalizes the covariance between the sensitive attributes of observations and their predicted decision from the decision boundary [14]. The bound on this covariance becomes a tunable hyperparameter, where tightening this bound on covariance more strictly enforces fairness but could have a worse impact on accuracy. They similarly exhibit a significant improvement in fairness with a modest decrease in accuracy, while also showing how the framework could be reversed to maximize fairness with accuracy constraints, as opposed to the more common setting that does the opposite.

The paper that we most directly draw on is from Bechavod and Ligett (2018), who similarly develop a penalizer for logistic regression based on distances from the decision boundary [15]. They derive this as a convex relaxation of constraining the difference between the false positive rate between different groups, which is initially a non-convex function. The penalizer they propose for false positive rate is as follows:

With $S_{a,y}$ indicating members of the dataset who are in protected group $a \in \{0, 1\}$ and whose true label is $y \in \{0, 1\}$, and with logistic regression weights θ,

$$\begin{aligned} R_{FP}^{AVD}(\theta; S) &= \left| \frac{\sum\limits_{x \in S_{00}} \theta^T x}{|S_{00}|} - \frac{\sum\limits_{x \in S_{10}} \theta^T x}{|S_{10}|} \right| \\ &= \left| \theta^T \underbrace{\left(\frac{\sum\limits_{x \in S_{00}} x}{|S_{00}|} - \frac{\sum\limits_{x \in S_{10}} x}{|S_{10}|} \right)}_{\overline{x}} \right| \\ &= \left| \theta^T \overline{x} \right| \end{aligned}$$

In essence, minimizing this penalizer ensures that for both values of the protected class, the members whose true labels are negative are on average classified the same distance from the decision boundary; this would indicate that, given a true or false label, the value of the protected attribute a does not affect the classification of that observation. For example, if the protected attribute is sex, and our two values are men and women, this penalizer indicates that the men whose true label is negative and the women whose true label is negative are on average assigned the same score by the logistic regressor.

Bechavod and Ligett also make a squared version of this penalizer so that it is differentiable at 0,

$$R_{FP}^{SD}(\theta; S) = \left(\theta^T \overline{x} \right)^2$$

and make an analogous penalizer for false negatives by substituting in groups S_{01} and S_{11}, indicating true positives of both classes.

What is notable about this solution is that the operative term in the penalizer that allows us to consider fairness, $\overline{x}$, is pre-computable from the dataset, and therefore doesn't require tuning in the optimization step itself. This offers an incredibly simple framework for creating penalizers for logistic regression that can be easily applied to other notions of fairness.

1.3 Intended Contributions

In this paper, we aim to build off of the aforementioned frameworks for penalizing unfairness in logistic regression by developing our own simple convex penalizers that capture notions of fairness and balance that model creators might consider desirable. First, we develop a penalizer that incentivizes

demographic parity, which incentivizes similar classification of different demographic groups as a whole. Second, we create a penalizer that allows us to weight correct and incorrect predictions differently for different demographic groups, in a way that penalizes confidently incorrect predictions more highly, and vice versa for correct predictions. Then, we implement these penalizers and tune their hyperparameters on a real-world, canonical dataset in the fairness literature, to display their performance and impacts on the intended fairness metrics and accuracy.

1.4 Organization of Paper

In **Section 2**, we describe the penalizers we have formulated, and describe why they should lead to the intended incentives in our optimization problem. In **Section 3**, we formulate the full logistic regression optimization problem with our added penalizers, and using previous results on logistic regression, derive the dual optimization problem and KKT conditions. In **Section 4**, we apply this optimization problem with different values of the penalizer to a real-world dataset. In **Section 5**, we discuss our results, possible future work, and limitations of our approaches.

2 Approaches

2.1 Demographic Parity Penalizer

Demographic parity, also known as statistical parity, is a fairness metric that requires classification results to be independent of the sensitive attribute. So, in a binary classification task with sensitive attribute A having possible group values a and b, and for predictions $\hat{Y}$, demographic parity requires that:

$$P(\hat{Y} = 1|A = a) = P(\hat{Y} = 1|A = b) \text{ [2]}$$

As evident from this definition, this metric is agnostic to any of the actual features of the data, and only requires all groups to be classified positive at the same rate. So, in the very likely case that there is a correlation between the outcome and the sensitive attribute, enforcing demographic parity is inconsistent with balancing error rate between classes, and might run counter to desired behaviors even with a focus on fairness [11]. However, demographic parity is still a widely used statistic, and there are certainly areas in which it might be used as an indicator of broader issues, or as a baseline while other metrics are optimized. For example, the United States Equal Employment Opportunity Commission (EEOC) uses an "80% rule" that requires that the rate of classified positives for one group is never below 80% of the rate for another group [16].

To include demographic parity as a penalizer in the optimization problem, we can modify the squared penalizer from Bechavod and Ligett to take the average distance from the classification boundary for all members of each class, and minimize the difference between them. So, for groups $A \in a, b$ and dataset S, our penalizer for demographic parity $x_{parity} = x_P$ is as follows:

$$\begin{aligned} R_P(\theta; S) &= \left(\frac{\sum\limits_{x \in a} \theta^T x}{|A = a|} - \frac{\sum\limits_{x \in b} \theta^T x}{|A = b|} \right)^2 \\ &= \left(\theta^T \underbrace{\left(\frac{\sum\limits_{x \in a} x}{|A = a|} - \frac{\sum\limits_{x \in b} x}{|A = b|} \right)}_{x_P} \right)^2 \\ &= \left(\theta^T x_P \right)^2 \end{aligned}$$

Where $x \in a$ indicates that observation x is part of protected group a, and $|A = a|$ is the total number of observations in group a.

When this penalizer equals 0, this means that the average distance from the classification boundary for members of class a and b is the same. On the one hand, this might not exactly capture demographic parity if all we care about are the resulting binary predictions, but this term might have more utility as a optimization penalizer since it also penalizes the classifier being much more confident in predictions for one class than another. Crucially, x_{parity} is precomputed from the data, so despite the somewhat complex idea we're trying to encode in the penalizer, the term we're ultimately adding to the optimization problem is very simple.

2.2 Reweighting Group Differences in Accuracy

As mentioned above, another approach to balancing classification results between demographic groups is to re-weight observations from groups differently, in such a way that weights populations that are a minority in the training set higher; otherwise, an imbalance in the prevalence of groups in the data would mean that the model could have a lower accuracy for less prevalent groups [6].

To construct this penalizer, we'll first consider a single observation (x, y) in the dataset. For this term, we'll let $y \in \{-1, 1\}$ instead of $\{0, 1\}$ as usual. The log-odds predicted by the logistic regression weights θ are $\theta^T x$. This value will be highly negative for a confident negative prediction, highly positive for a confident positive prediction, and near 0 for a prediction near the decision boundary. So, since y will have the same sign as $\theta^T x$ if the binary prediction is correct, the term $y\theta^T x$ will be positive if the prediction is correct and negative if it's incorrect. What's more, $y\theta^T x$ will be highly positive for very confident predictions and highly negative for very unconfident predictions, so this term serves as an effective higher-is-better metric of the model's predictions.

We're going to use this term as a way to reweight accuracy for different groups in our optimization problem by choosing a parameter $c \in (0, 1)$ that represents our choice of weights between the two demographic groups. Then, we will take the average of the above $y\theta^T x$ term for the two groups, multiply one by c, and the other by $(1 - c)$. So, our penalizer $R_{reweighting} = R_{RW}$ becomes:

$$\begin{aligned} R_{RW}(\theta; S) &= c \cdot \frac{\sum\limits_{x \in a} y\theta^T x}{|A = a|} - (1 - c) \cdot \frac{\sum\limits_{x \in b} y\theta^T x}{|A = b|} \\ &= \theta^T \underbrace{\left(c \cdot \frac{\sum\limits_{x \in a} yx}{|A = a|} - (1 - c) \cdot \frac{\sum\limits_{x \in b} yx}{|A = b|} \right)}_{x_{RW}} \\ &= \theta^T x_{RW} \end{aligned}$$

When c is between 0.5 and 1, we are treating this confidence metric as more important for group a, and when it is below 0.5, we are treating it as more important for group b. Note that since higher is better for this term, to use it as a penalizer in logistic regression, which is a minimization problem, we'll multiply x_{RW} by -1 in practice; in the following derivations, we'll treat x_{RW} as if this has already been applied.

3 Problem Statement

3.1 Primal Formulation

The primal formulation for vanilla logistic regression is as follows:

$$\min_{\theta} -ll(\theta; x, y)$$

Where $-ll(\theta; x, y)$ is the negative log-likelihood of the dataset's labels y given logistic regression weights θ and dataset features x, given by $\sum_i -\log(1 + e^{-y_i(\theta \cdot x_i)})$.

To create our primal optimization problem, we add on our chosen penalizers, and add a regularization constraint that keeps the squared norm of the weights θ under some scalar q:

$$\begin{aligned} \min_{\theta} \quad & -ll(\theta; x, y) + (\theta^T x_P)^2 + \theta^T x_{RW} \\ \text{s.t.} \quad & ||\theta||_2^2 \leq q \end{aligned} \tag{1}$$

(In practice, we will multiply tunable weights to our penalizers in order to vary the degree to which they affect the optimization problem; for the simplicity of this derivation, we'll omit these weights, though they would not affect the actual process of the derivation.)

3.2 Deriving a Dual Formulation

However, as we've discovered, calculating the dual of the log-likelihood function can be quite difficult, and often relies on numerical bounds that we cannot verify for our modification of the optimization problem. In discussion with TA Chester in office hours, we found a procedure by which we could more or less enumerate the dual in Keerthi et al. (2005), and we adapt our optimization problem as follows to fit their procedure [17].

Let $-ll(\theta; x, y)$ equal $\sum_i g(\xi_i)$, where $g(\xi) = \log(1 + e^{\xi})$ and $\xi_i = -y_i(\theta \cdot x_i)$. (Note that we are augmenting all observation vectors x_i with a leading 1 to account for an intercept parameter.) This is, of course, an equivalent formulation of the log-likelihood as before, but it will make it easier for us to compute the gradient with respect to θ of the Lagrangian in a few steps by removing θ from the log-likelihood.

This allows us to reformulate the primal formulation as such:

$$\begin{aligned} \min_{\theta, \xi} \quad & \sum_i g(\xi_i) + (\theta^T x_P)^2 + \theta^T x_{RW} \\ \text{s.t.} \quad & ||\theta||_2^2 \leq q; \\ & \xi_i = -y_i(\theta \cdot x_i) \quad \forall i \end{aligned} \tag{2}$$

Now, we can formulate the Lagrangian function, with Lagrange multipliers λ for the regularization term, and v_i for each value of ξ_i:

$$L(\theta, \lambda, v, \xi) = \sum_i g(\xi_i) + (\theta^T x_P)^2 + \theta^T x_{RW} + \lambda(||\theta||_2^2 - q) + \sum_i v_i \left(-\xi_i - y_i(\theta \cdot x_i)\right) \tag{3}$$

So, the Lagrange dual becomes:

$$g(\lambda, v, \xi) = \inf_{\theta, \xi} L(\theta, \lambda, v, \xi) \tag{4}$$

We can solve for this by finding the value of θ where $\nabla_\theta L = 0$, and the value of ξ where $\nabla_\xi L = 0$. It should be clear now why we chose to use Keerthi et al.'s formulation of the problem, since this will make the calculation of $\nabla_\theta L$ more tractable, while allowing us to cite previous work for $\nabla_\xi L = 0$.

3.2.1 Solving for Gradient w.r.t. θ

We know take the derivative of L with respect to θ.

$$\nabla_\theta L = 2(\theta^T x_P)x_P + x_{RW} + 2\lambda\theta - \sum_i v_i y_i x_i = 0 \tag{5}$$

$$\nabla_\theta L = 2(\theta^T x_P)x_P + 2\lambda\theta = (\sum_i v_i y_i x_i) - x_{RW} \tag{6}$$

We cannot solve for θ outright, but we can formulate a series of d equations, one for each dimension of x or θ. For element k of x and θ, where $k \in \{1, \ldots, d\}$:

$$2(\theta^T x_P)x_{P_k} + 2\lambda\theta_k = (\sum_i v_i y_i x_{i_k}) - x_{RW_k} \tag{7}$$

$$2(\sum_j \theta_j x_{P_j})x_{P_k} + 2\lambda\theta_k = (\sum_i v_i y_i x_{i_k}) - x_{RW_k} \tag{8}$$

Let $\sum_{j \setminus k}$ indicate the sum over all values of j (which are dimensions of x or θ) other than k:

$$2(\sum_{j\setminus k}\theta_j x_{P_j} + \theta_k x_{P_k})x_{P_k} + 2\lambda\theta_k = (\sum_i v_i y_i x_{i_k}) - x_{RW_k} \tag{9}$$

$$2x_{P_k}\sum_{j\setminus k}\theta_j x_{P_j} + 2\theta_k x_{P_k}^2 + 2\lambda\theta_k = (\sum_i v_i y_i x_{i_k}) - x_{RW_k} \tag{10}$$

$$2\theta_k x_{P_k}^2 + 2\lambda\theta_k = (\sum_i v_i y_i x_{i_k}) - x_{RW_k} - 2x_{P_k}\sum_{j\setminus k}\theta_j x_{P_j} \tag{11}$$

$$\theta_k(2x_{P_k}^2 + 2\lambda) = (\sum_i v_i y_i x_{i_k}) - x_{RW_k} - 2x_{P_k}\sum_{j\setminus k}\theta_j x_{P_j} \tag{12}$$

$$\theta_k = \frac{1}{(2x_{P_k}^2 + 2\lambda)}\left((\sum_i v_i y_i x_{i_k}) - x_{RW_k} - 2x_{P_k}\sum_{j\setminus k}\theta_j x_{P_j}\right) \tag{13}$$

While we cannot solve directly, this defines a system of d equations, in which each element of θ is defined in terms of all the other elements. When we solve this system of equations, we get θ^*, the value of θ for which $\nabla_\theta L = 0$.

3.2.2 Solving for Gradient w.r.t. ξ

When we take the gradient of L with respect to ξ_i, we get:

$$\frac{\partial L}{\partial \xi_i} = g'(\xi_i) - v_i = 0$$

Citing Remark 1 of Keerthi et al. (2005) – or just taking the derivative of g – we get [17]:

$$g'(\xi_i) = \frac{e^{\xi_i}}{1 + e^{\xi_i}}$$

So, to solve for ξ_i such that $\nabla_{\xi_i} L = 0$:

$$g'(\xi_i) = v_i$$

$$\frac{e^{\xi_i}}{1 + e^{\xi_i}} = v_i$$

$$e^{\xi_i} = v_i + v_i e^{\xi_i}$$

$$e^{\xi_i} - v_i e^{\xi_i} = v_i$$

$$e^{\xi_i}(1 - v_i) = v_i$$

$$e^{\xi_i} = \frac{v_i}{(1 - v_i)}$$

$$\xi_i = \log(\frac{v_i}{1 - v_i})$$

3.2.3 Dual Problem

Finally, we can get the Lagrange dual function by substituting the values of θ and ξ such that $\nabla L = 0$ into the Lagrangian function. Since we don't have a closed-form solution for θ, we'll define the optimal solution as θ^*.

$$g(\lambda, v) = \left(\sum_i g(\log(\frac{v_i}{1-v_i})) + (\theta^{*T} x_P)^2 + \theta^{*T} x_{RW} + \lambda(||\theta^*||_2^2 - q) + \sum_i v_i \left(-\log(\frac{v_i}{1-v_i}) - y_i(\theta^* \cdot x_i) \right) \right) \tag{14}$$

And so, the dual problem is as follows:

$$\begin{aligned} \max_{\lambda, v} \quad & g(\lambda, v) \\ \text{s.t.} \quad & \lambda \in \mathbb{R}^+ \\ & v \in \mathbb{R}_d \end{aligned} \tag{15}$$

3.3 KKT Conditions

The KKT conditions for the solution are as follows:

Primal Constraints:

$$||\theta||_2^2 - q \leq 0$$

$$\xi_i + y_i(\theta \cdot x_i) = 0 \quad \forall i$$

Dual Constraints:

$$\lambda \in \mathbb{R}^+, v \in \mathbb{R}_d$$

Complementary Slackness:

$$\lambda(||\theta||_2^2 - q) = 0$$

Gradient/Stationarity:

$$\nabla_{\theta,\xi} \left(\sum_i g(\xi_i) + (\theta^T x_P)^2 + \theta^T x_{RW} \right) + \lambda \nabla_{\theta,\xi}(||\theta||_2^2 - q) + \sum_i v_i \nabla_{\theta,\xi} \left(\xi_i + y_i(\theta \cdot x_i) \right) = 0$$

4 Results

To test our approach, we applied our data to an incredibly popular, canonical dataset in the machine learning space, used in many of the pre-cited papers, the "Adult" Census dataset [18]. The Adult dataset is derived from 1994 data from the US Census Bureau, and includes information about a respondent's employment, education, race, and more. The binary attribute we try to predict is whether or not the respondent's income is above or below $50,000, and the protected demographic attribute is the respondent's gender. In our analysis, this dataset has 46,033 observations, each with 46 attributes, most of which are one-hot encoded from categorical selections in the dataset.

For the purposes of our analysis, we use a 30/70 train-test split, replicating the one used by Bechavod and Ligett, and run the above optimization problem using the convex optimization package CVXPY [19, 20]. We use similar settings for CVXPY as Bechavod and Ligett, limiting the ECOS solver to 1000 iterations, and we handle any possible solver errors by using the SCS solver as a backup for 1500 iterations. Throughout the following results, we set the regularization parameter $q = 1$, meaning that the squared L2 norm of weights $||\theta||_2^2 = 1$. In addition, we only run the optimizer with one penalizer active at a time (to reduce the computational costs of a full grid search).

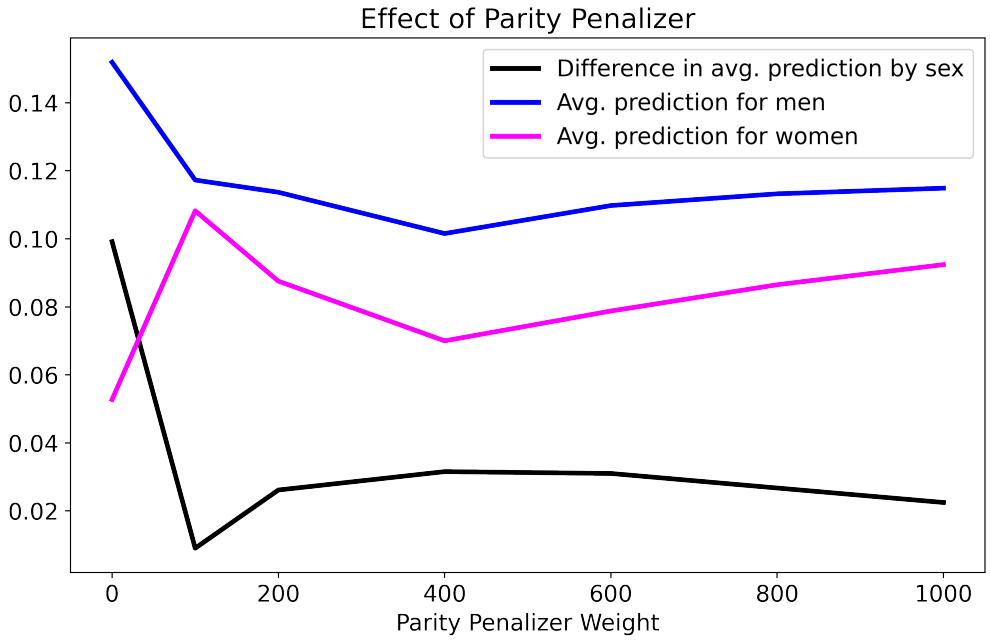

Figure 1: Effect of Parity Penalizer on Logistic Regression Test Results

Parity Weight	Avg. Prediction for Men	Avg. Prediction for Women	Difference in Average Prediction	Total Accuracy
0	0.1519	0.0527	0.0992	0.7959
100	0.1173	0.1083	0.0090	0.7852
200	0.1137	0.0876	0.0261	0.7938
400	0.1016	0.0700	0.0316	0.8040
600	0.1098	0.0788	0.0310	0.7983
800	0.1132	0.0865	0.0267	0.7944
1000	0.1149	0.0924	0.0225	0.7908

Table 1: Results for Parity Penalizer on Average Predictions by Sex

4.1 Parity Penalizer

As can be seen in the above figure and table, introducing x_{parity}, which penalizes differences in the proportion of predicted positives between men and women, does indeed seem to have the intended effect in practice. When the penalizer is not in effect at all, with weight = 0, 15.19% of men in the test set are predicted as positives (in this case, as having an income above $50,000), but only 5.27% of the women are predicted as positives.

When we instead have a weight of 100 (meaning we multiply $\left(\theta^T x_P\right)^2$ by 100 in the optimization problem), then we predict 11.73% of men and 10.83% of women as positives, bringing us much closer to our penalizer's goal of demographic parity. The difference in average prediction goes from 9.92% to 0.9%, an 11-fold improvement. This behavior is similar for larger values of the penalizer. (The reason that the parity difference does not monotonically go down as the weight is increased is because we're evaluating this penalizer on the test set, not the training set.)

We can also observe from the above table that the initial improvement in demographic parity is accompanied by a modest, if not negligible change in total classifier accuracy. Indeed, for parity weight = 400, we are able to lower parity significantly from the unpenalized classifier while having *higher* accuracy.

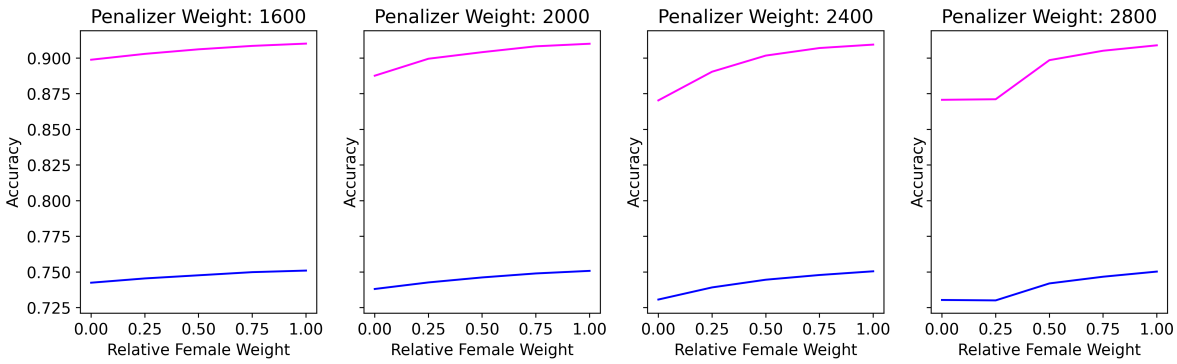

Figure 2: Effect of Reweighting Penalizer: Each plot represents a different weight of the penalizer, and in each one, we vary the relative importance given to male and female accuracy when optimizing on the training set.

Penalizer Weight	Relative Female Weight	Male Accuracy	Female Accuracy
1600	**0**	0.7425	0.8988
	0.25	0.7455	0.9029
	0.5	0.7477	0.9061
	0.75	0.7499	0.9085
	1	0.751	0.9101
2000	**0**	0.7381	0.8876
	0.25	0.7427	0.8995
	0.5	0.7462	0.9041
	0.75	0.749	0.9082
	1	0.7508	0.91

Penalizer Weight	Relative Female Weight	Male Accuracy	Female Accuracy
2400	**0**	0.7307	0.8703
	0.25	0.7392	0.8904
	0.5	0.7446	0.9017
	0.75	0.7479	0.907
	1	0.7505	0.9094
2800	**0**	0.7304	0.8707
	0.25	0.7301	0.8711
	0.5	0.742	0.8985
	0.75	0.7467	0.9051
	1	0.7503	0.9089

Table 2: Accuracy of Classifier by Sex for Different Penalizer Weights and Reweightings

4.2 Accuracy Re-weighting

In the above figure and table, we report the results of applying our penalizer that assigns a penalty to incorrect predictions corresponding to prediction confidence, reweighted by the sensitive attribute. Specifically, for different weights (1600, 2000, 2400, and 2800), which control the strength of the penalizer, we ran the optimization problem with a different set of relative weights for men and women (0, 0.25, 0.5, 0.75, and 1), and collected the resulting accuracy for each men and women on the test set.

We find, as expected, modest improvements in accuracy for women as we increase the penalizer's relative weight for female observations, with improvements in accuracy of around 3% from the case in which we penalize inaccuracy only in men to the case in which we penalize inaccuracy only in women.

However, as the above graphs make clear, we also yield the unexpected result that the accuracy for *men* also increases as we increase the relative weight of observations for *women*, across all four of the above penalizer weights. This is the opposite of what we might expect. Remember that we are evaluating these results on a test set, not the training set for the optimizer. So one reason for this result might be that since women are a minority in the dataset, reweighting their observations might actually be making our classifier more robust in general, and so we are preventing overfitting in the data to the existing male observations. Since reweighting is already an existing strategy for reducing overfitting in classifiers in general, by creating a model that reweights a minority population in our dataset, we may have unwittingly replicated that in our own models [21].

5 Discussion

Above, we showed that the effect of our penalizers on a logistic regression optimization problem trained on a real-world, canonical dataset from the machine learning fairness space. Our demographic parity penalizer achieves a large reduction in disparities in average prediction between men and women on the test set, with little to no reduction in classification accuracy. Our penalizer for reweighting accuracy by sex, although it did not act exactly how we expected, managed to improve accuracy for both men

and women, possibly through the unintended but positive effects of using reweighting to prioritize observations of minority groups.

5.1 Limitations

There are, obviously, some limitations to the approaches we use here. The most obvious is that, like loss functions for classifiers themselves, our fairness penalties are subject to overfitting. This is because they are computed entirely based on the observed averages within demographic groups in the training set, and so if the test data differs from these observed patterns, then our penalizers are no longer achieving their stated goals.

At the same time, though, our penalizers are no more subject to overfitting than are any learned model from a training set. Whenever we learn a model on a training set, we assume that the distribution of observations in our training set more or less matches that in the test set and in practice; by that same assumption, the marginal distributions of the other features for different values of the protected group should also remain similar, so our penalizers should still have the desired effect.

Another issue is that our penalizers add significant complexity to the logistic regression problem, by requiring their own hyperparameter tuning for the penalizer weight (and in the example of the accuracy-reweighting penalizer, there's an additional parameter that can vary).

5.2 Future Work

As we've alluded to above, there are still a great number more fairness metrics that different model creators might value, and many of them likely have their own corresponding penalties that could be applied to logistic regression to improve fairness on those metrics. The relaxation method proposed by Bechavod and Ligett is quite powerful, in that as long as we can factor out the logistic regression weights from the penalizer, we can insert the penalizer as a simple linear term in the objective function.

Another avenue for future work directly from our paper would be to prove whether optimizing our penalizer truly is equivalent to optimizing demographic parity or re-weighted accuracy. Bechavod and Ligett develop their penalizers based on statistical guarantees given in Woodworth et al. (2017) that prove that minimizing their penalizers for false positive rate and false negative rate do indeed yield their desired fairness metrics on datasets drawn from the same distribution as the training set [22]. However, we did not derive or discover any similar guarantees for our penalizers, which are simply based on our intuitions of how these terms might work. If it is possible to prove that our penalizers do indeed guarantee fairness on some metrics when optimized, this would give us a much stronger guarantee that they will perform as intended when properly tuned.

References

[1] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6), jul 2021.

[2] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning: Limitations and Opportunities*. fairmlbook.org, 2019. http://www.fairmlbook.org.

[3] Alexandra Chouldechova and Aaron Roth. A snapshot of the frontiers of fairness in machine learning. *Communications of the ACM*, 63(5):82–89, 2020.

[4] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. In *Ethics of Data and Analytics*, pages 254–264. Auerbach Publications, March 2022.

[5] Ziad Obermeyer, Brian Powers, Christine Vogeli, and Sendhil Mullainathan. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464):447–453, 2019.

[6] Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. Building classifiers with independency constraints. In *2009 IEEE international conference on data mining workshops*, pages 13–18. IEEE, 2009.

[7] Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. Fairness-aware learning through regularization approach. In *2011 IEEE 11th International Conference on Data Mining Workshops*, pages 643–650. IEEE, 2011.

[8] Felix Petersen, Debarghya Mukherjee, Yuekai Sun, and Mikhail Yurochkin. Post-processing for individual fairness, 2021.

[9] Arvind Narayanan. Translation tutorial: 21 fairness definitions and their politics. In *Proc. conf. fairness accountability transp., new york, usa*, volume 1170, page 3, 2018.

[10] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Rich Zemel. Fairness through awareness, 2011.

[11] Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 5(2):153–163, 2017. PMID: 28632438.

[12] Jon Kleinberg, Sendhil Mullainathan, and Manish Raghavan. Inherent trade-offs in the fair determination of risk scores, 2016.

[13] Naman Goel, Mohammad Yaghini, and Boi Faltings. Non-discriminatory machine learning through convex fairness criteria. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[14] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez Rogriguez, and Krishna P Gummadi. Fairness constraints: Mechanisms for fair classification. In *Artificial intelligence and statistics*, pages 962–970. PMLR, 2017.

[15] Yahav Bechavod and Katrina Ligett. Penalizing unfairness in binary classification, 2017.

[16] Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 259–268, New York, NY, USA, 2015. Association for Computing Machinery.

[17] S. S. Keerthi, K. B. Duan, S. K. Shevade, and A. N. Poo. A fast dual algorithm for kernel logistic regression. *Machine Learning*, 61(1-3):151–165, July 2005.

[18] Ron Kohavi. Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, page to appear, 1996.

[19] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[20] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

[21] Preethi Lahoti, Alex Beutel, Jilin Chen, Kang Lee, Flavien Prost, Nithum Thain, Xuezhi Wang, and Ed Chi. Fairness without demographics through adversarially reweighted learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 728–740. Curran Associates, Inc., 2020.

[22] Blake Woodworth, Suriya Gunasekar, Mesrob I Ohannessian, and Nathan Srebro. Learning non-discriminatory predictors. In *Conference on Learning Theory*, pages 1920–1953. PMLR, 2017.