FAIRLEARN: Interpretable and Configurable Algorithmic Fairness

Ankit Kulshrestha
Ilya Safro

Computer and Information Science Department, University of Delaware, Newark, DE-19716
akulshr@udel.edu isafro@udel.edu

Abstract

The rapid growth of data in the recent years has led to the development of complex learning algorithms that are often used to make decisions in real world. While the positive impact of the algorithms has been tremendous, there is a need to mitigate any bias arising from either training samples or implicit assumptions made about the data samples. This need becomes critical when algorithms are used in automated decision making systems that can hugely impact people's lives.

Many approaches have been proposed to make learning algorithms fair by detecting and mitigating bias in different stages of optimization. However, due to a lack of a universal definition of fairness, these algorithms optimize for a particular interpretation of fairness which makes them limited for real world use. Moreover, an underlying assumption that is common to all algorithms is the apparent equivalence of achieving fairness and removing bias. In other words, there is no user defined criteria that can be incorporated into the optimization procedure for producing a fair algorithm. Motivated by these shortcomings of existing methods, we propose the FAIRLEARN procedure that produces a fair algorithm by incorporating user constraints into the optimization procedure. Furthermore, we make the process interpretable by estimating the most predictive features from data. We demonstrate the efficacy of our approach on several real world datasets using different fairness criteria.

Reproducibility: All source codes, and experimental data are available in our Github repository[1]

Introduction

The development of data driven expert artificial intelligence (AI) systems in the recent years has had a significant impact in the way we interact with the world around us. The ever evolving sophistication of expert systems has led to a tremendous efficiency in making predictions about diverse data and consequently alleviate tedious manual processing. If however, the predictions made by an expert system are applied directly to a person or a group of persons then serious consequences can arise. For instance, an expert system trained on biased racial data may give allocate better healthcare to a certain racial group (Obermeyer et al. 2019) or tag more persons belonging to a certain race as relapse criminals (Larson 2016). Moreover, if data exhibits gender imbalance then an expert system may faultily show targeted advertisement based on gender when it is not supposed to (Datta, Tschantz, and Datta 2015) or allow only a certain gender to apply for a particular job (Dastin 2018).

The above real-world examples show the importance of having a fair expert system. In many situations the data collection is not in the hand of the algorithm designer and hence algorithmic fairness techniques are required to ensure a fair expert AI system. The need for fair algorithms has led to the development of two parallel branches of work each with its own goal (Pessach and Shmueli 2020). One of the branches is concerned with the *individual* fairness, i.e., the algorithm needs to ensure that any individual belonging to a population does not get biased decisions. The other branch has the objective of ensuring fairness amongst *sub-groups* of a given population. This objective is called statistical or group fairness. Examples of different statistical fairness definitions include Disparate Impact (Chierichetti et al. 2017; Ahmadian et al. 2019, 2020; Barocas and Selbst 2016), Demographic Parity (Zafar et al.; Zliobaite 2015; Calders, Kamiran, and Pechenizkiy 2009) and Equal Opportunity (Zafar et al. 2017b; Hardt, Price, and Srebro 2016; Donini et al. 2020). In this paper, we exclusively focus on optimizing supervised learning algorithms in the context of statistical fairness by using support vector machine as an exemplar supervised learning.

The different definitions of statistical fairness lead to the evaluation of an algorithm's performance in fundamentally different ways. For instance, the disparate impact measures the *ratio* of true positive rates across groups while equal opportunity measures the *difference* between the true positive rates and false positive rates across groups with the restriction that they are less than a specified threshold. It has been shown that the different fairness metrics are often incompatible with each other (Pleiss et al. 2017; Friedler, Scheidegger, and Venkatasubramanian 2016; Dwork and Ilvento 2018) and other complementary goals of optimization. There currently does not exist a universal definition of algorithmic fairness. This lack of a universal definition of fairness has led to the development of algorithms that optimize for only a particular definition. The resulting algorithm is only useful in scenarios where the motivating definition is applica-

[1]https://github.com/aicaffeinelife/FAIRLEARN/tree/master

ble. Even when algorithms are optimizing for a particular fairness criteria, they make a simplifying assumption that the best way to achieve fairness is to set the fairness constraint to zero for any given data. However, these thresholds must be either inferred from the given data or supplied to the optimization process as a hyperparameter since we want the algorithm to be an *equitable planner* (Kleinberg et al. 2018).

Our contribution The development of fair algorithms that optimize for certain definitions of fairness has hindered the development of a consistent framework for fairness. Lack of ability to parametrize and configure fairness in the existing AI expert systems represents a major gap that has to be bridged to correctly apply fairness in the real world systems. Our proposed algorithm FAIRLEARN takes a step in this direction, by decoupling the algorithm and the fairness criteria. We further address the lack of configurability in existing algorithms by introducing a tunable hyperparameter in our algorithm. More specifically we:

- Design a novel training procedure that is compatible with different types of classifiers. The procedure is configurable in the choice of criteria and the amount of fairness that can be injected into the model.
- Propose a data driven approach for estimating the critical predictive features in the data and utilizing the most predictive features for optimization. The discovered features also help in interpreting the predictions of the model in a much more principled manner.
- Demonstrate the efficacy of our method on several real world datasets that are used in expert AI systems with different fairness metrics.

The FAIRLEARN Algorithm

We begin by describing the notation that is used throughout the paper. Let $\mathcal{D}$ be a dataset consisting of n i.i.d. samples. Each sample in the dataset is a 3-tuple $(\mathbf{x}_i, s_i, y_i)$ where $\mathbf{x}_i \in \mathbb{R}^d$ is a data point consisting of d features. Associated with each data point is a label y_i and a *sensitive* feature s_i. We further assume that $y_i \in \{-1, 1\}$ and the sensitive feature is bi-valued i.e. $s_i \in \{a, b\}$. A user supplied fairness tolerance parameter f_{tol} is expected to control the amount of fairness injected into a model.

When $s_i \in \mathbf{x}_i$ then most algorithms that address fairness optimize to remove the influence of s_i over the final outcome (Zafar et al. 2017b; Donini et al. 2020; Dwork et al. 2012). However, this process is flawed for two reasons. First, it assumes that all features of the data including the sensitive feature contribute equally to the final outcome. Second, it does not account for the possibility that the sensitive feature may not *directly* affect the decision boundary but may indirectly influence it through another highly correlated feature.

We hypothesize that there must exist a set of features that are critical for prediction. In other words, if any feature belonging to this set is removed from the data, the overall performance of the algorithm will decrease significantly. We call this set as the ***critical feature set***. Depending on the data, s_i can either be present inside or outside the critical feature set. These cases give rise to different types of unfairness in a model as noted by Kamishmia *et al.* (Kamishima, Akaho, and Sakuma 2011). If the former case is true, then we have a case of direct bias and we cannot remove the sensitive feature from the data. However, if the latter case is true then there may still exist implicit bias caused by features having a high degree of correlation with the sensitive feature. Hence the removal of the sensitive feature does not guarantee fairness. If $s_i \notin \mathbf{x}_i$, i.e., the sensitive feature is not a part of the data point's feature then the same analysis can be applied and a critical feature set can be built either by concatenation of s_i to $\mathbf{x}_i$ (if s_i is deemed to be in the critical feature set) or by building a subset of critical features that are highly correlated with s_i. *One of the main features of our algorithm is that we estimate the critical feature set as the part of optimization process. The estimated critical features can be used to better interpret the underlying machine learning model.*

Algorithm 1 describes our procedure in broad strokes. It consists of three main building blocks:

- `FINDCRITFEATS`: This procedure determines the critical feature set from the given data and an unfair classifier.
- `FINDCOVARIATES`: For a given sensitive feature s, this procedure estimates the covariance matrix of the data and returns a list of the features that have a high degree of correlation with s.
- `OPTIMIZE`: Integrates the user supplied $\mathcal{F}$ into the training procedure and returns a fair classifier.

We partition $\mathcal{D}$ into a training set $\mathcal{D}_{\text{train}}$ and a test set $\mathcal{D}_{\text{test}}$. An unfair classifier $\mathbb{A}$ is used to estimate the critical feature set. We supply as inputs to our procedure $\mathcal{D}_{\text{train}}$, $\mathcal{D}_{\text{test}}$, and $\mathbb{A}$ along with a fairness criteria $\mathcal{F}$,the index of sensitive feature such that $\mathbf{x}[i] = s_i$ and a fairness threshold f_{tol}.

Our procedure (FAIRLEARN) begins by estimating the critical feature set $\mathcal{C}$ and a set of correlated features ψ from the given data (Lines 2 and Lines 3). The set $\mathcal{C}$ is sorted in descending order based on the magnitude of impact a given feature has on the accuracy. The set ψ is sorted in descending order based on the magnitude of covariance with the protected feature. If s_i is within $\mathcal{C}$, then we run the `OPTIMIZE` procedure with the objective of producing a fairer classifier $\mathcal{A}$ from $\mathbb{A}$ (Line 5). If s_i is not found with $\mathcal{C}$, then we cannot assume the model is fair. This is because, the sensitive feature can have a highly corelated feature which may be a part of the critical feature set. Hence, we check if any of its covariates are present in $\mathcal{C}$. We build a new critical feature set $\mathcal{C}'$ and then run `OPTIMIZE` (Line 8). If however, neither case is true then we can conclude the source of bias is *extrinsic* to the model and fairness can be achieved with data preprocessing techniques. We describe the different subroutines in the following subsections.

Finding Critical Features and Covariates

The first step in FAIRLEARN is to determine the critical features and the sensitive features covariates. We show our approach for this step in Algorithms 2 and Algorithm 3.

The main idea in the `FINDCRITFEATS` procedure is to leverage *permutation feature importance* (Breiman 2001) for estimating the overall importance of a particular feature. To effectively test the importance of a feature, we use the

Algorithm 1: The FAIRLEARN Algorithm

1: **procedure** FAIRLEARN($\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}}, \mathbb{A}, s, \mathcal{F}, f_{tol}$)
2: $\quad \mathcal{C} \leftarrow$ FINDCRITFEATS($\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}}, \mathbb{A}$)
3: $\quad \psi \leftarrow$ FINDCOVARIATES($\mathcal{D}_{\text{train}}, s$)
4: $\quad$ **if** $s \in \mathcal{C}$ **then**
5: $\quad\quad \mathcal{A} \leftarrow$ OPTIMIZE($\mathcal{D}_{\text{train}}, \mathcal{C}, \mathcal{F}, f_{tol}$)
6: $\quad$ **else if** $\psi \neq \emptyset$ & $\texttt{Any}(\psi) \in \mathcal{C}$ **then**
7: $\quad\quad \mathcal{C}' \leftarrow \psi \cap \mathcal{C}$ $\quad \triangleright$ intersection of covariates and critical features.
8: $\quad\quad \mathcal{A} \leftarrow$ OPTIMIZE($\mathcal{D}_{\text{train}}, \mathcal{C}', \mathcal{F}, f_{tol}$)
9: $\quad$ **else**
10: $\quad\quad \mathcal{A} \leftarrow \mathbb{A}$
11: $\quad$ **end if**
12: $\quad$ **return** $\mathcal{A}$
13: **end procedure**

accuracy on the test set as a measure of performance for the given classifier. However, we stress that nothing prevents using other performance measures that might be more suitable for specific tasks and data. We accept a parameter K that determines the number of permutations to perform per feature. This parameter must be chosen carefully since too many permutations can impact the running time of the algorithm, especially for large datasets. For each feature j we permute it K times and measure the score on the test set (Line 11). A permutation of feature j is defined as a shuffling of data points that breaks the association between feature j and the observed outcome (Line 10). The importance of a feature j is then given by $\Delta - \frac{1}{K}\sum_{i=1}^{K} \delta_k$, where Δ is the accuracy of the model when no features were permuted, δ_k is the accuracy of the model at the k^{th} permutation step. At the end of the procedure we sort the indices in the descending order of importance and return the indices to the main algorithm (Line 16).

In order to find the set of features that are correlated with s from data, we adopt a Maximum Likelihood approach that aims to fit an estimator to determine the covariance matrix. We assume that all examples are i.i.d and the number of samples is much larger than the number of features in the data. Once the covariance matrix ε is determined (Line 2) we compute the sensitive feature covariates ε_s. We reject the features that are negatively correlated. We further reject the sensitive feature and return the indices of the remaining covariates sorted in descending order (Line 5).

The OPTIMIZE Procedure

The `OPTIMIZE` accepts the training set $\mathcal{D}_{\text{train}}$, the critical feature set $\mathcal{C}$, a fairness criteria $\mathcal{F}$ and a tolerance parameter f_{tol}. The main goal of this procedure is to provide a consistent framework for finding a fairer algorithm $\mathcal{A}$ by optimizing for any $\mathcal{F}$. It is a general procedure in the sense that, a user may supply their own `OPTIMIZE` procedure depending upon the choice of unfair algorithm $\mathbb{A}$. The only restriction is that the supplied procedure must be parametrized in a similar manner to the one shown in Algorithm 1. In such a case our algorithm will still find the best features for the data and return a fairer algorithm according to the provided optimization procedure. For instance, if one wishes to optimize

Algorithm 2: The FINDCRITFEATS Procedure

1: **procedure** FINDCRITFEATS($\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{test}}, \mathbb{A}, K$)
2: $\quad \mathbb{A} \leftarrow \text{TRAIN}((X, Y) \in \mathcal{D}_{\text{train}})$
3: $\quad \mathcal{N} \leftarrow \text{DIM}(\mathcal{D}_{\text{train}})$
4: $\quad (X_{\text{test}}, Y_{\text{test}}) \leftarrow \mathcal{D}_{\text{test}}$
5: $\quad I \leftarrow \{0, 0, \ldots, 0\}$ $\quad \triangleright$ Feature importance list for $\mathcal{N}$ features
6: $\quad \Delta \leftarrow \text{ACCURACY}(\mathbb{A}, X_{\text{test}}, Y_{\text{test}})$
7: $\quad$ **for** $j \in \mathcal{N}$ **do**
8: $\quad\quad$ Scores $\leftarrow \emptyset$
9: $\quad\quad$ **for** $k \in RANGE(K)$ **do**
10: $\quad\quad\quad \hat{X}_{\text{test}} \leftarrow \text{RANDOMSHUFFLE}(X_{\text{test}}, j)$
11: $\quad\quad\quad \delta_k \leftarrow \text{ACCURACY}(\mathbb{A}, \hat{X}_{\text{test}}, Y_{\text{test}})$
12: $\quad\quad\quad$ Scores $\leftarrow$ Scores $\cup\, \delta_k$
13: $\quad\quad$ **end for**
14: $\quad\quad I_j \leftarrow \text{IMPORTANCE}(\Delta, \text{Scores})$
15: $\quad$ **end for**
16: $\quad$ **return** $\text{ARGSORT}(I)$
17: **end procedure**

Algorithm 3: The FINDCOVARIATES Procedure

1: **procedure** FINDCOVARIATES($\mathcal{D}_{\text{train}}, s$)
2: $\quad \varepsilon \leftarrow \text{MLE\_ESTIMATOR}(\mathcal{D}_{\text{train}})$
3: $\quad \varepsilon_s \leftarrow \varepsilon[:, s]$
4: $\quad C \leftarrow \varepsilon_s > 0.0$
5: $\quad$ **return** $\text{ARGSORT}(C)[0{:}]$ $\quad \triangleright$ Return all covariate features except the first.
6: **end procedure**

a multi-layer perceptron on such fairness metric as the Equal Opportunity, she can provide an `OPTIMIZE` procedure that integrates the fairness constraint as a regularization term in the loss function.

Optimizing with Support Vector Machines We develop an instance of `OPTIMIZE` procedure for a support vector machine (SVM) with the Equal Opportunity (EO) fairness metric. We largely adopt the framework of Donini *et al.* (Donini et al. 2020) in the design of the optimization procedure with fairness constraints with some important modifications in the formulation. We translate the definition of the EO metric, i.e.,

$$\Big|P[D=1|\mathcal{S}=a,\mathcal{Y}=1]-P[D=1|\mathcal{S}=b,\mathcal{Y}=1]\Big| \leq \epsilon, \tag{1}$$

where D is the predicted outcome, into a convex constraint:

$$\Big|\mathbb{E}\left[l^a(f(\mathbf{x}),y_+)-l^b(f(\mathbf{x},y_+)\right]\Big| \leq \epsilon. \tag{2}$$

The convex and smooth loss function $l^g(f,y_+)$ in Equation 2 measures the performance of a family of functions f in predicting positively labeled samples y_+ belonging to a group $g=\{a,b\}$. The overall goal of optimization is to find

$$\min\left\{f\in\mathcal{H};\text{s.t. }\mathbb{E}[l^a(f,y_+)-l^b(f,y_+)]\right]\leq\epsilon\right\}, \tag{3}$$

where $\mathcal{H}$ is the hypothesis space of functions.

As a representative example, we choose $f(\mathbf{x})=\langle\mathbf{w},\phi(\mathbf{x})\rangle$ and assume that the underlying hypothesis space $\mathcal{H}$ is a Reproducing Kernel Hilbert Space (RKHS). This assumption allows us to define a kernel mapping in high dimension space $K(\mathbf{X},\mathbf{X}')=\langle\phi(\mathbf{X}),\phi(\mathbf{X}')\rangle$. In (Donini et al. 2020), a parameter $\mathbf{u}_g$ is introduced that measures the barycenter of the induced feature mapping as:

$$\mathbf{u}_g=\frac{1}{N_g^+}\sum_{j\in I_g^+}\phi(x_j) \tag{4}$$

In Equation 4, N_g^+ is the number of positively labeled points belonging to the sensitive group g and I_g^+ is the set of indices of such points. Based on the definition of $\mathbf{u}_g$ in Equation 4, the fairness constraint was specified as $\langle\mathbf{w},|\mathbf{u}_a-\mathbf{u}_b|\rangle\leq\epsilon$. The constraint can be understood as a way to minimize the effect of the sensitive group clusters on the overall decision boundary. This formulation is effective in injecting fairness in model, but only matches the first order moments of the clusters. We improve upon this approach by taking into account the *spread* of the respective clusters. Specifically, we redefine $\mathbf{u}_g$ as:

$$\hat{\mathbf{u}}_g=\frac{1}{\sigma_g}\left[\frac{1}{N_g^+}\sum_{j\in I_g^+}\phi(x_j)\right] \tag{5}$$

In Equation 5, $\sigma_{\hat{g}}$ is the square root of inter-cluster variance:

$$\sigma_g^2=\frac{1}{N_g^+-1}\sum_{i=1}^{n}\left[\phi(x_i)_g^+-\mu_{\hat{g}}\right]^2 \tag{6}$$

In Equation 6, $\mu_{\hat{g}}$ is the mean of the positively labeled points in a group different from the current one. In our experiments we found that this scaling significantly improved the fairness of our model. Based on this modification, the constrained optimization objective is stated as follows:

$$\begin{aligned}&\min_{\gamma\in\mathbb{R}^n}\sum_{i=1}^{n} l\left(\sum_{j=1}^{n}K_{ij}\gamma_i,y_i\right)+\lambda\sum_{(i,j)=1}^{n}\gamma_i\gamma_jK_{ij}\\ &\text{s.t.}\left|\sum_{i=1}^{n}\gamma_i[\frac{1}{\sigma_a}(\frac{1}{N_a^+}\sum_{j\in I_a^+}K_{ij})-\frac{1}{\sigma_b}(\frac{1}{N_b^+}\sum_{j\in I_b^+}K_{ij})]\right|\leq f_{tol}\end{aligned} \tag{7}$$

In Equation 7, we make use of the Representer theorem (Kimeldorf and Wahba 1971) to express $\mathbf{w}=\sum_{i=1}^{n}\gamma_i\phi(x_i)$, where $\gamma\in\mathbb{R}^n$ and solve the Tikhonov regularization problem.

The entire setup effectively translates to solving the dual of a soft-margin SVM with the fairness constraint. *It is important to mention that in contrast with the approach in (Donini et al. 2020), we do not restrict ourselves to the case where $f_{tol}=0$ and allow it to be a user defined parameter which is extremely important for the real-world applications.*

Algorithm 4 shows the procedure for solving the dual of the optimization problem in Equation 7. The variables ϑ_a,ϑ_b are selected based on the fairness criteria and can be a set of either positively or negatively labeled points. The matrices $\mathbf{G},\mathbf{h}$ encode the box constraint $0\leq\vec{\alpha}\leq C$. For simplicity, we add the fairness criteria to the affine constraint in a quadratic program. We make use of an off-the-shelf quadratic solver (e.g., CVXOPT or MOSEK (ApS 2019)). The resulting vector of solutions $\vec{\alpha}$ is used to get the support vectors and make predictions via the `GET_ALG` function.

Algorithm 4: OPTIMIZE with SVM Objective

procedure OPTIMIZE($\mathcal{D}_{\text{train}},\mathcal{C},\mathcal{F},f_{tol},s$)
  $s\leftarrow$ FINDINDEX($\mathcal{C},s$)
  $\mathbf{X},\mathbf{y}\leftarrow\mathcal{D}_{\text{train}}$
  $\mathbf{X}\leftarrow\mathbf{X}[:,\mathcal{C}]$
  $N\leftarrow|\mathcal{D}_{\text{train}}|$
  $\vartheta_a\leftarrow\mathcal{I}_a^{\mathcal{F}}$ ▷ Set of all indexes with sign $\mathcal{F}$ and group a
  $\vartheta_b\leftarrow\mathcal{I}_b^{\mathcal{F}}$ ▷ Set of all indexes with sign $\mathcal{F}$ and group b
  $K\leftarrow$ KERNEL($\mathbf{X},\mathbf{X}$)
  $\mathbf{P}\leftarrow$ OUTER(y,y) $*\,K$ ▷ Setting up optimization problem
  $\mathbf{q}\leftarrow-1*\mathbf{y}$
  $\mathbf{G}\leftarrow\left[\mathbb{D}^N,\mathbf{I}\right]$
  $\mathbf{h}\leftarrow[0,-C]$ ▷ C is the control parameter in soft SVM
  $\mathbf{A}\leftarrow[\mathbf{y},\mathbf{y}*|\hat{\mathbf{u}}_a-\hat{\mathbf{u}}_b|]$ ▷ Fairness constraint
  $\mathbf{b}\leftarrow[0,f_{\text{tol}}]$
  $\vec{\alpha}\leftarrow$ CONEQP($\mathbf{P},\mathbf{q},\mathbf{G},\mathbf{h},\mathbf{A},\mathbf{b}$)
  return GET_ALG($\vec{\alpha}$)
end procedure

Experiments

Implementation of FAIRLEARN and data are available in (TBA 2021). The goal of our experiments is threefold. First, we want to establish that FAIRLEARN achieves fairness given any fairness criteria without suffering a significant degradation in accuracy. Second, we wish to study the effect of the user supplied f_{tol} values on different data. Third, we want to demonstrate the generality of our approach by showing that FAIRLEARN works in a model and data agnostic manner. For all our experiments we use the `OPTIMIZE` procedure developed for SVM in the previous section.

Dataset	# Data points	# Features	Sensitive Attribute
Adult	32561	12	Gender
COMPAS	6171	10	Race
German	1000	59	Gender

Table 1: Statistics of real world datasets used in experiments.

Expert AI Systems Considered

In this work we consider three different types of expert AI systems along with a representative dataset for each type.

- **Income Prediction**: An automated data driven system may be deployed by a company to obtain a projection of hiring salary for a worker given their background. A pitfall in this particular system may be that it may learn to weigh the gender of a person as the most important factor leading to potentially sexist behavior. We choose the Adult dataset (Dua and Graff 2017) as a representative example of such data. The goal is to predict if a given person will earn more than $50,000 per year.
- **Criminal Recidivism Prediction**: Prisons often keep track of their inmates data and various describing attributes. An automated expert system trained on biased data may exhibit racist behavior and predict a member belonging to a particular community as more likely to commit a crime after being released. We choose the COMPAS recidivism dataset (Larson 2016) as a representative example.
- **Loan Granting**: Banks store the credit history along with different attributes of their customer in their database. A new request for loan is checked with this data and a final decision made. An AI system deployed to make these decisions can become biased either towards a particular community or gender or income-group. We choose the German credit dataset (Dua and Graff 2017) as a representative example of data used to train such an AI system.

A description of these datasets is provided in Table 1. For all datasets,we perform a 10-fold cross validation to determine the optimal hyperparamters C and γ for non-linear SVM. For all datasets except the Adult dataset, we repeat this procedure over five different runs and report the average accuracy with their standard deviation. We use the provided train/test split for the Adult dataset and report the accuracy over the test set. *Unlike previous work in this area, we evaluate our method on different values of f_{tol}.* To model a realistic scenario, one critical assumption we make in all experiments is that the sensitive parameter s is always inside the training data.

Fairness Metrics

For a supervised learning paradigm, we define three distinct variables. D is the predicted variable by a supervised learning algorithm, $\mathcal{Y}$ is the ground truth label and S is the sensitive feature of the data point. Fairness criteria for the supervised paradigm can be divided into two broad categories depending upon the choice of the conditioning variable.

The first category is a set of metrics that are conditional on the observed variable $\mathcal{Y}$. This is equivalent to saying $D \perp \mathcal{S}|\mathcal{Y}$, where $D \perp \mathcal{S}$ implies that D is independent of $\mathcal{S}$. Some criteria that fall in this category are Equal Odds (where $D = -1 \perp \mathcal{S}|\mathcal{Y} = 1$ and $D = 1 \perp \mathcal{S}|\mathcal{Y} = 1$) (Hardt, Price, and Srebro 2016), and Equal Opportunity (where $D = 1 \perp \mathcal{S}|Y = 1$) (Donini et al. 2020; Zafar et al. 2017b). A defining characteristic of these metrics is that they can be interpreted as a score function $\mathcal{L} : \mathcal{X} \times \mathcal{S} \times \mathcal{Y} \to \mathbb{R}$. For instance, the Equal Opportunity metric can be expressed as

$$\mathcal{L}(\mathcal{X}, \mathcal{Y}, \mathcal{S}) = \mathbb{E}\left[|l_a^+(f(\mathbf{x}_i), y_i) - l_b^+(f(\mathbf{x}_i), y_i)|\right] \quad (8)$$

and a corresponding score function for Equality of True Negative Rates can be written as

$$\mathcal{L}(\mathcal{X}, \mathcal{Y}, \mathcal{S}) = \mathbb{E}\left[|l_a^-(f(\mathbf{x}_i), y_i) - l_b^-(f(\mathbf{x}_i), y_i)|\right] \quad (9)$$

In Equation 8 and 9, $l_g^{\pm}$ is the per sample loss for any learning algorithm f on either positively or negatively labeled points belonging to a sensitive group g. In this paper we consider the Equal Opportunity as a representative example of fairness from this category. Specifically, we evaluate the Difference of Equal Opportunity(DEO) metric (Donini et al. 2020)

$$\Big| P\left[D = 1|\mathcal{S} = a, \mathcal{Y} = 1\right] - P\left[D = 1|\mathcal{S} = b, \mathcal{Y} = 1\right]\Big| \leq \epsilon. \quad (10)$$

The second category is a set of metrics that are conditional on the *predicted* outcome D. This is equivalent to saying $\mathcal{Y} \perp \mathcal{S}|D$. Some examples of metrics belonging to this class are Equality of Negative Predictive Value $Y = -1 \perp \mathcal{S}|D = -1$ and Equality of Positive Predictive Value $\mathcal{Y} = 1 \perp \mathcal{S}|D = 1$. For a more thorough treatment of these metrics and the ones discussed above we refer an interested reader to (Mitchell et al. 2021). These metrics quantify the *predictive quality* of a trained algorithm. This is important since a deployed system rarely has access to true outcomes. If an algorithm has a good predictive quality then we can be confident that the decisions made by the algorithm more or less reflect the actual observed outcomes. In this paper we choose the Equality of Negative Predictive Value (NPV) defined as

$$\Big| P\left[\mathcal{Y} = -1|\mathcal{S} = a, D = -1\right] - P\left[\mathcal{Y} = -1|\mathcal{S} = b, D = -1\right]\Big| \leq \epsilon. \quad (11)$$

For both metrics in Equation 10 and Equation 11 a lower value signifies a higher amount of fairness in a model.

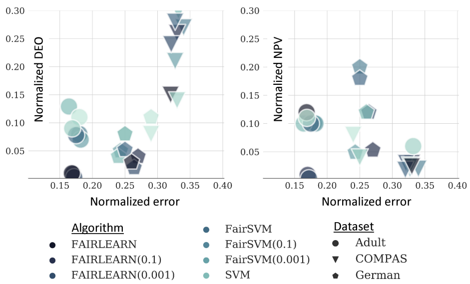

Figure 1: Comparison of FAIRLEARN classifier with other methods on different metrics. Left: DEO, Right: NPV

Experimental Evaluation

We now present our results on the three real world datasets discussed earlier. Based on the constrained optimization objective in Equation 7, we divide our experiments across two categories - a fair SVM trained with our dual formulation on the full feature set and a fair SVM trained with our proposed FAIRLEARN procedure. We call the former SVM as *FairSVM* and the latter as *FAIRLEARN*. We measure the performance of both algorithms with three baseline methods - an unfair SVM trained to optimize accuracy, the FERM method by Donini *et al.* (Donini et al. 2020) and the post processing step proposed by Hardt *et al.* (Hardt, Price, and Srebro 2016). For the baseline results of Hardt *et al.* we report the results obtained in (Donini et al. 2020). We reproduce the results for the FERM method baseline by using the code provided by the authors. We report our results for both linear and non-linear cases. For the non-linear case we use the RBF kernel $K(\mathbf{X}, \mathbf{X'}) = e^{-\gamma||\mathbf{X}-\mathbf{X'}||^2}$.

Table 2 shows the results of our experiments. The results show a consistent improvement in the DEO objective for both FairSVM and FAIRLEARN classifiers across all datasets. Additionally, we note that the FAIRLEARN classifier with the non-linear kernel does not suffer a significant decrease in accuracy and improves upon the DEO metric, achieving the best performance amongst all methods on the German and Adult datasets in the non-linear case. This result suggests that for critical features discovered by our algorithm are useful and using them does not significantly lower the classification performance of the algorithm and leads to a much more fairer version of the same. In the case of linear kernel as well, our method performs much better than baselines on both Adult and German datasets in terms of fairness performance. We observe a significant drop in accuracy on the GERMAN dataset for the linear case and we feature this drop to the linear kernel not being the best choice for critical features selected by the algorithm. On the Compas dataset, our results are close to the best performing FERM method on the DEO metric. For the NPV metric, the results suggest that a better performance may be reached if we use the FairSVM method since it consistently achieves a lower score on the different datasets.

The results also show an interesting relationship between the two metrics. We note that for datasets where an unfair algorithm had a low DEO, the fairer version of the algorithms had a much higher NPV rate. However, when the unfair algorithm had a higher DEO, the our proposed fair algorithms achieved a much lower NPV rate along with the DEO *without being explicitly optimized for the former metric*. These observations suggest that there may be an implicit relation between the metrics that are conditioned on the observed variable and the ones that are conditioned on the predicted variable. In a broader sense, our results showcase the trade-offs that need to be made depending on the real world situation.

It is helpful to understand the relationship between the error rate of an algorithm and the fairness metrics since it helps to establish a quantitative understanding of achievable fairness at a desired error rate. Figure 1 shows the normalized value of fairness metrics along with the error rate at which this value was achieved. For the Adult dataset, we observe that we can get a fair classifier with a minimal loss in accuracy. For the German credit dataset and the COMPAS dataset the decision to use a classifier is more nuanced since different classifiers (including ours with different f_{tol}) can attain different fairness values at different error rates. For instance, if DEO is the fairness metric of choice, then for German dataset our method provides a lower value with a slightly higher error rate than by the classifier trained according to (Donini et al. 2020).To the best of our knowledge, this is the first time any work has compared the effect of two different metrics at different fairness thresholds across different datasets.

Effect of Tolerance Parameter

We introduced the fairness tolerance parameter f_{tol} as a way to allow a user to control the amount of fairness in a model. For the SVM dual optimization objective, this parameter translates to the allowed "influence" of the sensitive groups over the final decisions. Our goals in this experiment were to see if the change in f_{tol} changes the fairness of the model and to quantify the effect of retaining only the critical feature set against the baseline of an unfair SVM trained with exactly the same parameter grid. We compare the two classifiers that we discussed earlier. For both classifiers we used the RBF kernel and performed a 10-fold cross validation procedure. We averaged the results over five independent runs on the COMPAS and the German Dataset.

Table 3 shows the results of our experiments. We can see that at all thresholds both algorithms significantly improve in the DEO metric for both Adult and Compas datasets. The case of the German dataset is interesting since the DEO metric does not uniformly decrease from the baseline for the classifier trained with full features. One reason for this observation can be the low DEO of the baseline algorithm on the German dataset. In the other two, this metric is higher in the baseline which implies a higher percentage of misclassified samples across sensitive groups. The classifier trained with our proposed procedure either matches or reduces the DEO. We also evaluate the classifiers on the NPV metric. The classifier trained with full features shows a consistent reduction from baseline across the thresholds. However, classifier trained with the FAIRLEARN procedure shows a

Method	GERMAN			COMPAS			ADULT		
	Acc.	DEO	NPV	Acc.	DEO	NPV	Acc.	DEO	NPV
SVM	0.76 ± 0.06	0.04 ± 0.02	0.08 ± 0.04	0.67 ± 0.02	0.30 ± 0.08	0.06 ± 0.03	0.83	0.129	0.10
FairSVM	0.75 ± 0.01	0.05 ± 0.02	$\mathbf{0.04 \pm 0.04}$	0.67 ± 0.02	0.21 ± 0.06	0.026 ± 0.01	0.824	0.071	**0.10**
FAIRLEARN	0.736 ± 0.04	$\mathbf{0.024 \pm 0.01}$	0.12 ± 0.07	0.68 ± 0.006	0.15 ± 0.023	$\mathbf{0.018 \pm 0.01}$	0.831	**0.003**	0.127
FERM	0.75 ± 0.014	0.083 ± 0.045	0.18 ± 0.11	0.67 ± 0.01	$\mathbf{0.14 \pm 0.026}$	0.024 ± 0.021	0.831	0.09	0.11
Hardt	0.71 ± 0.03	0.11 ± 0.18	-	0.71 ± 0.01	0.08 ± 0.01	-	0.82	0.11	-
Lin. SVM	0.767 ± 0.02	0.08 ± 0.06	0.11 ± 0.05	0.65 ± 0.01	0.244 ± 0.57	0.03 ± 0.02	0.80	0.013	0.17
Lin. FairSVM	0.74 ± 0.02	$\mathbf{0.04 \pm 0.02}$	0.12 ± 0.06	0.65 ± 0.01	0.15 ± 0.03	0.03 ± 0.02	0.803	**0.003**	**0.16**
Lin. FAIRLEARN	0.701 ± 0.03	0.05 ± 0.1	$\mathbf{0.10 \pm 0.09}$	0.65 ± 0.01	0.153 ± 0.05	0.02 ± 0.01	0.78	0.03	0.17
Lin. FERM	0.74 ± 0.02	0.072 ± 0.04	0.17 ± 0.063	0.65 ± 0.01	$\mathbf{0.09 \pm 0.038}$	0.038 ± 0.02	0.80	0.01	0.165
Lin. Hardt	0.61 ± 0.15	0.15 ± 0.13	-	0.67 ± 0.03	0.21 ± 0.09	-	0.80	0.10	-

Table 2: Results for all datasets on the accuracy, DEO and NPV metrics. In dataset where an explicit test set is not provided, the results are reported as mean $\pm$ standard deviation of five independent runs. Best results are shown in bold.

DATASET	f_{tol}	Fair SVM			FAIRLEARN		
		Acc	DEO	NPV	Acc	DEO	NPV
ADULT	0	0.82	0.07	0.101	0.83	0.003	0.127
	0.1	0.82	0.08	0.101	0.83	0.009	0.126
	0.001	0.82	0.07	0.10	0.83	0.003	0.127
Baseline	-	0.835	0.129	0.1117	0.835	0.129	0.117
COMPAS	0	0.67 ± 0.01	0.28 ± 0.04	0.023 ± 0.01	0.68 ± 0.01	0.15 ± 0.02	0.023 ± 0.01
	0.1	0.66 ± 0.01	0.27 ± 0.05	0.02 ± 0.01	0.67 ± 0.01	0.26 ± 0.09	0.03 ± 0.02
	0.001	0.673 ± 0.01	0.21 ± 0.07	0.02 ± 0.01	0.679 ± 0.01	0.24 ± 0.07	0.03 ± 0.02
Baseline	-	0.668 ± 0.02	0.30 ± 0.08	0.06 ± 0.03	0.668 ± 0.02	0.30 ± 0.08	0.06 ± 0.03
GERMAN	0	0.76 ± 0.03	0.05 ± 0.04	0.12 ± 0.06	0.73 ± 0.02	0.03 ± 0.02	0.27 ± 0.15
	0.1	0.75 ± 0.02	0.08 ± 0.07	0.20 ± 0.078	0.73 ± 0.04	0.04 ± 0.072	0.053 ± 0.05
	0.001	0.757 ± 0.01	0.05 ± 0.02	0.049 ± 0.04	0.736 ± 0.04	0.02 ± 0.02	0.12 ± 0.07
Baseline	-	0.767 ± 0.068	0.04 ± 0.02	0.08 ± 0.04	0.767 ± 0.068	0.04 ± 0.02	0.08 ± 0.04

Table 3: Effect of f_{tol} on fairness on algorithm trained with full feature and with critical set for the non-linear case. Baseline in a biased SVM trained on the same dataset.

consistent decrease only in the case of German and Compas datasets. For the Adult dataset, the NPV increases across thresholds. We hypothesize that the results are a combination of two factors. First, the NPV and DEO appear to be inversely related and the DEO decreases sharply. Second, if the sensitive parameter happens to be a part of the critical feature set, then the algorithm may aggressively optimize for the provided fairness constraint. We shall examine the second hypothesis in some detail in the next subsection.

Critical Features of Different Datasets

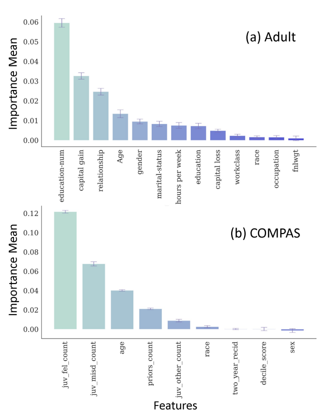

Figure 2: Critical features discovered by the FAIRLEARN.

One of the goals of the FAIRLEARN problem was to provide an interpretable way to estimate the features that contribute most to the overall outcome. To the best of our knowledge, we believe that our algorithm marks the first step in the direction of interpretable fairness in machine learning.We have previously argued that all features are not critically important for the final prediction made by the model. By extension, it also follows that some features which were thought to be the sensitive parameter did not exert a significant influence on the final outcome. Figure 2 shows the critical features discovered by the FAIRLEARN algorithm for the Adult and the COMPAS dataset. The bars show the mean of five independent runs along with their standard deviations of the feature permutation algorithm. In the case of Adult dataset, we can see that the sensitive feature "gender" is not as significant as other metrics like education and age. Interestingly, another potential sensitive feature "race" has a very low importance. The critical features discovered for the COMPAS dataset are surprising. We find that the model places the *least* importance on the "Race" feature. On the other hand, the gender feature contributes more towards the overall outcome. Critical features of German dataset are not shown since we binarized the features into 59 categorical features. The most important features were found to be - Property, Status, Credit History, Employment and Savings. The sensitive feature "Sex" had an average feature importance of only 0.001.

Discussion

We discuss the main results of our work and possible future directions that emerge from it.

Configurable Fairness. The results of the previous section show that having the right setting can lead to significantly better fairness results. A key feature of our algorithm is to integrate flexible constraints into the optimization procedure. From our experiments, we recommend a range of

$f_{tol} \in \left[10^{-2}, 10^{-3}\right]$ as a "good" initial setting for optimization. In this paper we have shown an instance of optimization procedure using SVM, but in practice any classifier can be used once the appropriate objective has been defined.

In our experiments we have shown a way to formulate a fairness metric into an optimization constraint and integrated it into a quadratic program to find decision boundaries using SVM. However, our algorithm is not limited by the choice of the machine learning algorithm. Once a fairness constraint has been translated into an optimization metric, it can be integrated with non-convex algorithms like neural networks using the method proposed in (Yang et al. 2020). We leave this direction for future work.

Interpretability and Fairness. The analysis of critical features in the previous section showed a surprising result - the attribute that was assumed to be the source of unfairness in the model, actually did not contribute anything to the overall prediction of the classifier. These incorrect assumptions about the sensitive attribute can be detrimental to the performance of the algorithm and can have unintended consequences in the real world. To the best of our knowledge, our method is the first to provide a principled approach for determining the sensitive parameter in the given dataset. We note that model-free methods of feature importance (Fisher, Rudin, and Dominici 2019) may yield better results.

Automatic Threshold Detection. In certain scenarios there may not be a good setting for the tolerance parameter. In this case automated methods like (Maclaurin, Duvenaud, and Adams 2015; Franceschi et al. 2017) can be used to perform an iterative optimization of the fairness hyperparameter before the actual model training is started.

One potential strategy for automatic threshold detection can be *hyperparameter optimization* (Maclaurin, Duvenaud, and Adams 2015; Franceschi et al. 2017; Bengio 2000). Given an initial estimate of f_{tol} we can iteratively adjust it's value in the direction that leads to a lower amount of disparity between sensitive groups. In the case where the objective is convex (e.g. an SVM) we can solve a max-max optimization problem where we aim to maximize the dual of an SVM subject to a constraint that maximizes the distance of the sensitive attribute cluster from the decision boundary. In general, this question is an open problem and a rich area of research.

Related Work

In this section we discuss some related work in the development of fair classification algorithms.

Fairness Definitions. There have been a number of fairness definitions proposed for algorithmic fairness that address both group and individual fairness. The definitions of group fairness vary in the way the predictive measures for a classifier are considered. Almost all the definitions in group fairness are proposed for a binary classifier and a binary sensitive attribute. Two definitions of group fairness consider group parity to be tantamount to fairness. Disparate Impact (Feldman et al. 2015; Zafar et al. 2017b) principle defines a classifier to be fair if the ratio of the positive prediction rates across sensitive groups is less than a certain threshold. Demographic Parity deems a classifier fair if the positive prediction rates are equal across sensitive groups. Both these metrics are defined for ideal cases and are hard to apply in the real world since datasets are rarely class balanced. Other metrics define fairness by considering the confusion matrix of a binary classifier and requiring that some performance metrics remain the same across sensitive groups. For instance, Equalized odds metric (Hardt, Price, and Srebro 2016) requires that the true positive and false rates of a classifier are same across the sensitive group. Equal opportunity (Hardt, Price, and Srebro 2016; Donini et al. 2020) relaxes this constraint by only requiring the true positive rates remain the same. A different definition of (un)fairness is provided by Zafar *et al* (Zafar et al. 2017a) based on the misclassification rates in a binary classifier. A more thorough review of fairness metrics and their applications can be found in (Mitchell et al. 2021). One persistent issue in the algorithmic fairness community is that the fairness metrics while being internally consistent are inconsistent with other fairness metrics. Pleiss *et al.* (Pleiss et al. 2017) show that it is impossible to satisfy equalized odds at the same time except when the true positive and false negative rates are at zero. Dwork *et al.* (Dwork and Ilvento 2018) also note the incompatibility of the fairness metrics under composition. Our work does not propose a new fairness metric, but it provides a way to integrate different metrics in a model agnostic fashion. This flexibility also makes our method future proof, since new metrics can be easily integrated into our algorithm.

Fair Optimization Strategies. Fair classification algorithms can be grouped into three major categories depending on the stage in which they integrate the fairness constraint. The first category consists of algorithms that preprocess the data before running classification. Some examples of works that fall in this category are by Dwork *et al.* (Dwork and Ilvento 2018) which proposes to learn a new data representation that removes the effect of protected attribute on the final decision. Kamiran *et al.* (Kamiran and Calders 2009) also preprocess the data by changing the labels of highest ranking negative samples. This ranking is learned by fitting a Naive Bayesian model on the dataset. Friedler *et al.* (Friedler, Scheidegger, and Venkatasubramanian 2016) also define a data transformation method based on the Earth mover distance to mitigate disparate impact in algorithms. A drawback of methods in this category is that they assume that the classifier is inherently fair. This prevents any measurement of the bias introduced by the model in the optimization process. Thus, even if the data is bias free, the resulting classifier may still output biased predictions due to it's internal bias. The second category of algorithms aims to integrate the fairness constraint during the training process of the classifier. Zafar *et al.* (Zafar et al. 2017b) integrate the disparate treatment metric as a fairness constraint of a convex optimization problem. Donini *et al.* (Donini et al. 2020) present a way to integrate the equal opportunity metric as a linear constraint in the SVM objective. Kamishsima *et al.* (Kamishima, Akaho, and Sakuma 2011) integrate fairness by adding a regularizer to the training objective that measures the degree of bias introduced by the algorithm. Dwork *et al.* (Dwork et al. 2017) also propose a method which aims to train independent classifiers and then jointly fine-tune them to mitigate unfairness in the models. While this method does not integrate the fairness constraint into the training procedure, we put this method in this category since it mitigates fairness during an optimization procedure. The main advantage of these methods is that they allow for explicit modeling of fairness criteria and provide a way to control the amount of fairness in a model. Our work also falls in this category of algorithms since we expect to optimize for fairness during the training process. The third category of algorithms introduce fairness as a post-processing step. A prominent work in this direction include the method by Hardt *et al.* (Hardt, Price, and Srebro 2016) who introduce a post-processing step for the Equal Odds metric. This post processing takes the form of a linear optimization that is applied to the predictions made by a given unfair model. An advantage of methods that use a post processing step is that they are immensely flexible since they do not depend on the training data. Thus, they provide the most general methods of all the three categories and are useful in cases where the training data may not be available due to privacy concerns (e.g., medical record data). On the flip side however, there is no way to control the amount of fairness in the model which makes them useful in a limited number of applications as well as their start is always dictated by the initial solution of the original algorithm.

Conclusion

In this paper we introduced a training algorithm that provides a way to train fair classifiers in a configurable and interpretable manner. Our method is based on the observation that all features do not contribute equally to the final prediction and exploits a learned ranking of features to train models with top most predictive features. We decouple the fairness metric and the classifier thereby allowing a user to run optimization on the classifier with any fairness metric. Moreover, we allow the user to control the amount of fairness that can be introduced in the model through a tolerance parameter. As we observe in our experiments, our method consistently outputs better fairness results on different fairness metrics. We believe that our algorithm marks a step towards a unified view of fairness in machine learning.

Our work opens up many different directions of future research. One direction is to search for a more nuanced method of feature ranking. We would like to see the effect of Bayesian modeling in feature ranking and it's efficacy on fairness. A second direction is in the direction of optimal threshold detection for fairness. A hyperparameter optimization approach that detects the best tolerance parameter would be a welcome extension to our work. A relatively underexplored area in fairness literature is the optimization of algorithms with composite sensitive attributes. We define these attributes to be a combination of one or more sensitive attributes and jointly varying over their individual value. For example, if the sensitive attribute is race and gender, then a composite attribute is race-gender that takes all values in both race and gender combined. A future work direction then is to examine the critical feature hypothesis with composite attributes and extend our work to handle these kind of sensitive attributes. We note that multi-level SVM (Sadrfaridpour, Razzaghi, and Safro 2019) is a way to improve the scaling of SVMs to large datasets. The adaptive SVM parameter training version (Sadrfaridpour, Palmer, and Safro 2020) takes the scalability one step further and could be an interesting approach to combine with the fariness parameter training. While our algorithm can be integrated directly into the framework, we leave the analysis of performance of our algorithm on this framework as future work.

References

Ahmadian, S.; Epasto, A.; Knittel, M.; Kumar, R.; Mahdian, M.; Moseley, B.; Pham, P.; Vassilvitskii, S.; and Wang, Y. 2020. Fair Hierarchical Clustering. *arXiv:2006.10221 [cs, stat]*. ArXiv: 2006.10221.

Ahmadian, S.; Epasto, A.; Kumar, R.; and Mahdian, M. 2019. Clustering without Over-Representation. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 267–275. ArXiv: 1905.12753.

ApS, M. 2019. *The MOSEK Optimization Toolbox*.

Barocas, S.; and Selbst, A. 2016. Big Data's Disparate Impact. *California Law Review*, 104: 671.

Bengio, Y. 2000. Gradient-Based Optimization of Hyperparameters. *Neural Computation*, 12: 1889–1900.

Breiman, L. 2001. Random Forests. *Machine Learning*, 45(1): 5–32.

Calders, T.; Kamiran, F.; and Pechenizkiy, M. 2009. Building Classifiers with Independency Constraints. In *2009 IEEE International Conference on Data Mining Workshops*, 13–18.

Chierichetti, F.; Kumar, R.; Lattanzi, S.; and Vassilvitskii, S. 2017. Fair Clustering Through Fairlets. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30*, 5029–5037. Curran Associates, Inc.

Dastin, J. 2018. Amazon scraps secret AI recruiting tool that showed bias against women. *Reuters*.

Datta, A.; Tschantz, M. C.; and Datta, A. 2015. Automated Experiments on Ad Privacy Settings: A Tale of Opacity, Choice, and Discrimination. arXiv:1408.6491.

Donini, M.; Oneto, L.; Ben-David, S.; Shawe-Taylor, J.; and Pontil, M. 2020. Empirical Risk Minimization under Fairness Constraints. *arXiv:1802.08626 [cs, stat]*. ArXiv: 1802.08626.

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository.

Dwork, C.; Hardt, M.; Pitassi, T.; Reingold, O.; and Zemel, R. 2012. Fairness through Awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, 214–226. New York, NY, USA: Association for Computing Machinery. ISBN 9781450311151.

Dwork, C.; and Ilvento, C. 2018. Group fairness under composition. In *Proceedings of the 2018 Conference on Fairness, Accountability, and Transparency (FAT* 2018)*.

Dwork, C.; Immorlica, N.; Kalai, A. T.; and Leiserson, M. 2017. Decoupled classifiers for fair and efficient machine learning. *arXiv:1707.06613 [cs]*. ArXiv: 1707.06613.

Feldman, M.; Friedler, S.; Moeller, J.; Scheidegger, C.; and Venkatasubramanian, S. 2015. Certifying and removing disparate impact. arXiv:1412.3756.

Fisher, A.; Rudin, C.; and Dominici, F. 2019. All Models are Wrong, but Many are Useful: Learning a Variable's Importance by Studying an Entire Class of Prediction Models Simultaneously. arXiv:1801.01489.

Franceschi, L.; Donini, M.; Frasconi, P.; and Pontil, M. 2017. Forward and Reverse Gradient-Based Hyperparameter Optimization. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 1165–1173. PMLR.

Friedler, S. A.; Scheidegger, C.; and Venkatasubramanian, S. 2016. On the (im)possibility of fairness. arXiv:1609.07236.

Hardt, M.; Price, E.; and Srebro, N. 2016. Equality of Opportunity in Supervised Learning. *arXiv:1610.02413 [cs]*. ArXiv: 1610.02413.

Kamiran, F.; and Calders, T. 2009. Classifying without discriminating. In *2009 2nd International Conference on Computer, Control and Communication*, 1–6.

Kamishima, T.; Akaho, S.; and Sakuma, J. 2011. Fairness-aware learning through regularization approach. In *2011 IEEE 11th International Conference on Data Mining Workshops*, 643–650. IEEE.

Kimeldorf, G.; and Wahba, G. 1971. Some results on Tchebycheffian spline functions. *Journal of mathematical analysis and applications*, 33(1): 82–95.

Kleinberg, J.; Ludwig, J.; Mullainathan, S.; and Rambachan, A. 2018. Algorithmic Fairness. *AEA Papers and Proceedings*, 108: 22–27.

Larson, J. 2016. Correctional Offender Management Profilingfor Alternative Sanctions (COMPAS) dataset. https://github.com/propublica/compas-analysis/.

Maclaurin, D.; Duvenaud, D.; and Adams, R. 2015. Gradient-based Hyperparameter Optimization through Reversible Learning. In Bach, F.; and Blei, D., eds., *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, 2113–2122. Lille, France: PMLR.

Mitchell, S.; Potash, E.; Barocas, S.; D'Amour, A.; and Lum, K. 2021. Algorithmic Fairness: Choices, Assumptions, and Definitions. *Annual Review of Statistics and Its Application*, 8(1): 141–163.

Obermeyer, Z.; Powers, B.; Vogeli, C.; and Mullainathan, S. 2019. Dissecting racial bias in an algorithm used to manage the health of populations. *Science*, 366(6464): 447–453.

Pessach, D.; and Shmueli, E. 2020. Algorithmic Fairness. *arXiv:2001.09784 [cs, stat]*. ArXiv: 2001.09784.

Pleiss, G.; Raghavan, M.; Wu, F.; Kleinberg, J.; and Weinberger, K. Q. 2017. On Fairness and Calibration. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30*, 5680–5689. Curran Associates, Inc.

Sadrfaridpour, E.; Palmer, K.; and Safro, I. 2020. AML-SVM: Adaptive Multilevel Learning with Support Vector Machines. In *2020 IEEE International Conference on Big Data (Big Data)*, 788–797.

Sadrfaridpour, E.; Razzaghi, T.; and Safro, I. 2019. Engineering fast multilevel support vector machines. *Machine Learning*, 108(11): 1879–1917.

TBA. 2021. FAIRLEARN: code and data. https://anonymous.4open.science/r/FAIRLEARN-0FD7. The anonymous link will be replaced with GitHub link.

Yang, Y.; Yuan, Y.; Chatzimichailidis, A.; van Sloun, R. J.; Lei, L.; and Chatzinotas, S. 2020. ProxSGD: Training Structured Neural Networks under Regularization and Constraints. In *International Conference on Learning Representations*.

Zafar, M. B.; Valera, I.; Gomez Rodriguez, M.; and Gummadi, K. P. 2017a. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th international conference on world wide web*, 1171–1180.

Zafar, M. B.; Valera, I.; Rodriguez, M. G.; and Gummadi, K. P. ???? Fairness Constraints: Mechanisms for Fair Classification. 9.

Zafar, M. B.; Valera, I.; Rodriguez, M. G.; and Gummadi, K. P. 2017b. Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment. *Proceedings of the 26th International Conference on World Wide Web*, 1171–1180. ArXiv: 1610.08452.

Zliobaite, I. 2015. On the relation between accuracy and fairness in binary classification. *arXiv preprint arXiv:1505.05723*.