Learning Fair and Interpretable Representations via Linear Orthogonalization

Yuzi He, Keith Burghardt, Kristina Lerman
Information Sciences Institute
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292

Abstract

To reduce human error and prejudice, many high-stakes decisions have been turned over to machine algorithms. However, recent research suggests that this *does not* remove discrimination, and can perpetuate harmful stereotypes. While algorithms have been developed to improve fairness, they typically face at least one of three shortcomings: they are not interpretable, they lose significant accuracy compared to unbiased equivalents, or they are not transferable across models. To address these issues, we propose a geometric method that removes correlations between data and any number of protected variables. Further, we can control the strength of debiasing through an adjustable parameter to address the trade-off between model accuracy and fairness. The resulting features are interpretable and can be used with many popular models, such as linear regression, random forest and multilayer perceptrons. The resulting predictions are found to be more accurate and fair than several comparable fair AI algorithms across a variety of benchmark datasets. Our work shows that debiasing data is a simple and effective solution toward improving fairness.

Introduction

Machine learning (ML) models sift through mountains of data to make decisions on matters big and small: e.g., who should be shown a product recommendation, hired for a job, or given a home loan. Machine inference can systematize decision processes to take into account orders of magnitude more information, produce accurate decisions, and avoid the common pitfalls of human judgment, such as prejudice and blind spots. Moreover, unlike people, machines will never make poor decisions when tired (Danziger, Levav, and Avnaim-Pesso 2011), pressed for time or distracted by other matters (Shah, Mullainathan, and Shafir 2012; Mani et al. 2013).

Recent research suggests, however, that discrimination remains pervasive, even in ML models (Angwin et al. 2016; Chouldechova 2017; Dressel and Farid 2018; O'neil 2016). For example, a model used to evaluate criminal defendants for recidivism assigned higher risk scores to African Americans than to Caucasians (Angwin et al. 2016). As a result, reformed African American defendants, who would never commit another crime, were deemed by the model to present a higher risk to society—as much as twice as high (Angwin et al. 2016; Dressel and Farid 2018)—as reformed white defendants, with potentially grave consequences on how they were treated by the justice system.

Emerging field of AI fairness has produced approaches for mitigating harmful model biases (Dwork et al. 2012; Chouldechova 2017; Chouldechova and Roth 2018), such as penalizing unfair inferences for particular models (Dwork et al. 2012; Berk et al. 2017), or creating representations that do not strongly depend on protected features (Jaiswal et al. 2018; Moyer et al. 2018; Locatello et al. 2019). These methods, however, lack least one of three critical attributes: interpretability, accuracy, or generalizability. *Interpretability* is necessary for understanding social factors and individual features contributing to discrimination and bias, as well as improving transparency and accountability of AI systems. In contrast to black box models, fair models need to be able to explain their decisions. In terms of performance, although models must sacrifice *accuracy* to fairness (Pierson et al. 2017), the trade-off need not be as dramatic as what current methods achieve. The issue of *generalizability* stems from the specialization of current methods to specific ML models. These methods cannot easily generalize to other models. For example, while (Zafar et al. 2017; Kamiran, Calders, and Pechenizkiy 2010; Berk et al. 2017) all create different methods for fair ML, each method is specialized to regressions, support vector machines (SVM), or random forests. Similarly, while previous methods create fair latent features for neural networks (NN) (Jaiswal et al. 2018; Moyer et al. 2018), the methods cannot be easily applied to improve fairness in non-NN models. These fair AI algorithms were not meant to be generalizable, because there do not seem to be adequate meta-algorithms that debias a whole host of ML models. One might naively expect that we can just create a single fair model and apply it to all datasets. The problem is that model performance varies greatly on different datasets. While NNs are critical for, e.g., image recognition (Ciregan, Meier, and Schmidhuber 2012), other methods perform better for small data (Olson, Wyner, and Berk 2018), especially when the number of dimensions is high and the sample size low (Liu, Wei, and amd Qiang Yang 2017). There is no one-size-fits-all model and there is no one-size-fits-all debiasing method. Is there an easier way to create fairer predictions other than specialized methods for specialized ML models? Chen et al. offer some clues to ad-

dressing this fundamental issue in fair AI (Chen, Johansson, and Sontag 2018), namely that by addressing data biases, we can potentially improve fair AI across the spectrum of models, and achieve fairness without sacrificing greatly prediction quality.

Following the ideas of Chen et al., we therefore develop a geometric method for *debiasing features*. Depending on the hyperparameter we choose, these features are mathematically guarentees to be uncorrelated with specified sensitive, or *protected*, features. This method is exceedingly fast and the debiased features are highly correlated with the original features (average Pearson correlations are between 0.993–0.994 across the three datasets studied in this paper). These features are as interpretable as the original features when applied to any model. When applied to linear regression, for example, the coefficients are the same or similar to the coefficients of the original features when controlling for protected variables (see Methods). These debiased features serve as a fair representation of data that can be used with a number of ML models, such as linear regression, random forest, SVMs, and NNs. Due to the small size of the benchmark data, we do not use our features to train NNs in this paper, because NNs could easily overfit the data. While previous methods have created fair representations (Olfat and Aswani 2018; Samadi et al. 2018; Jaiswal et al. 2018; Moyer et al. 2018), these methods create representations that are either not interpretable, like PCA components, or the relationship between these fair representations and the original features have not been established. We evaluate the proposed approach on a number of now-benchmark datasets. We show that models using these debiased features are more accurate for almost any level of fairness we desire.

In the rest of the paper, we first review recent advances in fair AI to highlight the novelty of our method. Next, we describe in the Methods section our methodology to improve data fairness, and the definitions of fairness we use in the paper. In Results, we describe how our method improves fairness in both synthetic data and empirical benchmark data. We compare to several competing methods and demonstrate the advantages of our method. Finally, we summarize our results and discuss future work in the Conclusion section.

Related Work

Social scientists use linear regression for data analysis due to its simplicity and interpretability. Interpretability comes from regression coefficients, which specify how the outcome, or response, changes when features change by one unit. However, regression creates unfair outcomes, even when protected features are excluded from the model, because other features may be correlated with them.

To make regression models fair, researchers introduced a loss function to penalize regression for unfair outcomes (Berk et al. 2017). Similarly, (Zafar et al. 2015) created fair logistic regression by introducing fairness constraints that limit the covariance between protected features and the outcome. An alternate method achieved fairness by constraining false positive or false negative rates (Zafar et al. 2016). There are some issues in these works. First, protected features are not included in the logistic model with fairness constraints. While this improves privacy, it forces the parameters of logistic models to take certain combinations which will minimize the correlation with the protected features. This can reduce the accuracy when the constraints are strict. The issue for the second method is mainly numeric. The algorithm requires an optimization of a convex loss function on a non-convex parameter space. While these models are generally interpretable, the approaches do not transfer to other models. Their accuracy also often suffers in comparison to neural methods.

Researchers have explored a variety of methods for learning fair representations of data (Jaiswal et al. 2018; Moyer et al. 2018; Louizos et al. 2015; Xie et al. 2017; Zemel et al. 2013; Samadi et al. 2018; Olfat and Aswani 2018). Some of those works use NNs to embed raw features in a lower-dimensional space, such that the embedding will contain the information about the outcome variable, but at the same time, contain little information about the protected feature. Fair logistic models or fair scoring, on the other hand, can be regarded as a one dimensional embedding of data, which makes sure that the predictions, $\hat{y}$, are independent of the protected features. They are mainly used with NNs, which while being highly accurate, often lack interpretability. Two methods were instead developed to improve fairness of PCA features (Samadi et al. 2018; Olfat and Aswani 2018), but, while they can be applied to non-NN ML models, they lack interpretability compared to the original features.

Johndrow and Lum (2017) proposed an algorithm which removes sensitive information about protected groups based on inverse transform sampling. The algorithm transforms individual features such that the transformed features satisfy the marginal distribution. Although this method can guarantee that predictions are fair in a probabilistic sense, it has a critical disadvantage — as the number of protected features n_p increases, the number of protected groups increases as $O(2^{n_p})$. This means that in order to properly estimate conditional and marginal distribution of features, one needs exponentially increasing population size. Our method overcomes these difficulties by using linear algebra as the basis for learning unbiased representations. This allows our algorithm to only take $O(n_p^2)$ time to debias data. Moreover, our method is a white box: it is interpretable and can be fully scrutinized, unlike a black box method.

Methods

We describe a geometric method for constructing fair interpretable representations. These representations can be used with a variety of ML methods to create fairer and accurate models of data.

Fair Interpretable Representations

We consider tabular data with n entries and m features. The features are vectors in the n-dimensional space, denoted as $\boldsymbol{x}_i$ where $i = 1, 2, \cdots, m$, and one of the columns corresponds to the outcome, or target variable $\boldsymbol{y}$. Among the features, there are also n_p protected features, $\boldsymbol{p}_i, i = 1, \ldots, n_p$. As a pre-processing step, all features are centered around the mean: $\langle x_i \rangle = 0$.

We describe a procedure to debias the data so as to create linearly fair features. We aim to construct a representation $\boldsymbol{r}_j$ of a feature $\boldsymbol{x}_j$, that is uncorrelated with n_p protected columns $\boldsymbol{p}_i, i = 1, \dots n_p$, but highly correlated to feature $\boldsymbol{x}_j$. We recall that Pearson correlation between the representation r_j and any feature x_k is defined as

$$\text{Corr}(r_j, x_k) = (\text{E}[r_j \cdot x_k] - \text{E}[r_j]\text{E}[x_k])/(\sigma_{r_j}\sigma_{x_k}),$$

where E[.] is the expectation, and $\sigma_{r_j} = \sqrt{\text{E}[r_j^2] - \text{E}[r_j]^2}$ and $\sigma_{x_k} = \sqrt{\text{E}[x_k^2] - \text{E}[x_k]^2}$. Because all the features are centered (and we also assume that r_j is centered), $\text{E}[r_j] = \text{E}[x_k] = 0$, we have

$$\sigma_{r_j} = \sqrt{\text{E}[r_j^2]} = \|r_j\|/\sqrt{n},$$

$$\sigma_{x_k} = \sqrt{\text{E}[x_k^2]} = \|x_k\|/\sqrt{n}$$

and

$$\text{E}[r_j \cdot x_k] = \boldsymbol{r}_j \cdot \boldsymbol{x}_k/n.$$

Therefore

$$\text{Corr}(r_j, p_i) = \boldsymbol{r}_j \cdot \boldsymbol{p}_i/(\|r_j\| \cdot \|p_i\|)$$

and

$$\text{Corr}(r_j, x_j) = \boldsymbol{r}_j \cdot \boldsymbol{x}_j/(\|r_j\| \cdot \|x_j\|).$$

Zero correlations between r_j and n_p protected columns requires that r_j lives in the solution space of $\boldsymbol{r}_j \cdot \boldsymbol{p}_i = 0, i = 1 \dots n_p$. Maximizing correlations between r_j and x_j under this constraint is equivalent to projecting $\boldsymbol{x}_j$ into the solution space of $\boldsymbol{r}_j \cdot \boldsymbol{p}_i = 0, i = 1 \dots n_p$.

To calculate $\boldsymbol{r}_j$, we can first create an orthonormal basis of vectors $\boldsymbol{p}_i$, which we can label as $\bar{\boldsymbol{p}}_i$. We then construct a projector $P_f = \sum_{i=1}^{n_p} \bar{\boldsymbol{p}}_i \bar{\boldsymbol{p}}_i^T$. The representation $\boldsymbol{r}$ is given as

$$\boldsymbol{r}_j = \boldsymbol{x}_j - P_f \boldsymbol{x}_j = (I - P_f)\boldsymbol{x}_j. \tag{1}$$

Using the GramSchmidt process, the orthonormal basis can be constructed in $O(n \times n_p^2)$ time and for every fair representation of features, the projection takes $O(n \times n_p)$ time. Given n_f features, the total time of the algorithm is $O(n \times n_f \times n_p^2)$ Therefore our method scales linearly with respect to the size of the data and the number of features. In practice, this is exceedingly fast. For example, this algorithm takes only 90 milliseconds to run on the Adult dataset described below, which has 20K rows and over 100 features.

While the previous discussion was on how to create linearly fair features, one can make linearly fair outcome variables, $\boldsymbol{r}_y$ through the same process. In prediction tasks, however, we do not have access to the outcome data. While our method does not guarantee that every model's estimate of the outcome variable, $\hat{y}$ is fair, we find that it can significantly improve the fairness compared to competing methods. Moreover, in the special case of linear regression, it can be shown that the resulting estimate, $\hat{y}$, is uncorrelated with the protected variables.

Inevitably, the accuracy of a model using such linearly fair features will drop compared to using the original features, because the solution is more constrained. To address this issue, we introduce a parameter $\lambda \in [0, 1]$, which indicates the fairness level. We define the parameterized latent variable as

$$\boldsymbol{r}'_j(\lambda) = \boldsymbol{r}_j + \lambda \cdot (\boldsymbol{x}_j - \boldsymbol{r}_j). \tag{2}$$

Here, $\lambda = 0$ corresponds to $\boldsymbol{r}'_j(\lambda) = \boldsymbol{r}_j$, which is strictly orthogonal to the protected features $\boldsymbol{p}_i$; while $\lambda = 1$ gives $\boldsymbol{r}'(\lambda) = \boldsymbol{x}_j$.

The protected features can be both real valued and cardinal. The fair representation method can also handle categorical protected features by introducing dummy variables. Specifically, if a variable X has k categories $x_1, x_2, ..., x_k$, we can can convert them to $k-1$ binary variables where the i^{th} variable is 1 if the variable is category x_i, and otherwise 0. If all variables are 0, then the category is x_k. As a simple example, if a feature X has 3 categories, x_1, x_2, and x_3, then the dummy variables would be $\tilde{x}_1$ and $\tilde{x}_2$. If $\tilde{x}_1 = 1$, the category is x_1, if $\tilde{x}_2 = 1$, then the category is x_2, and otherwise is x_3. The condition of fairness in this case is interpreted as same mean value of the latent variables in different categorical groups.

Fair Models

Using the procedure described above, we can construct a fair representation of every feature, and use the fair features to model the outcome variable. Consider a linear regression model that includes all features: n_p protected features $p_i, i = 1, \cdots, n_p$ and $n_f = m - n_p$ non-protected features features $x_i, i = 1, \dots, n_f$.

$$\hat{y} = \beta_0 + \sum_{i=1}^{n_f} \beta_i x_i + \sum_{i=1}^{n_p} \gamma_i p_i. \tag{3}$$

After transforming the features to fair features x'_i, the fair regression model reduces to:

$$\hat{y}' = \beta'_0 + \sum_{i=1}^{n_f} \beta'_i x'_i. \tag{4}$$

We can prove that $\beta_i = \beta'_i, i = 1, \dots, n_f$, but the predicted value $\hat{y}'$ is uncorrelated with protected features $p_i, i = 1, \dots, n_p$. In general linear regression, such as logistic regression, this proof does not hold, but we numerically find that coefficients are similar.

We should take a step back at this point. The fair latent features are close approximations of the original features, therefore we expect that, and in certain cases can prove, that the regression coefficients of the fair features should be approximately the coefficients of the original features. The fair features can, by this definition, be considered almost as interpretable as the original features.

In addition to regression, fair representations could be used with other ML models, such as AdaBoost (Freund and Schapire 1997), NuSVM (Chang and Lin 2011), random forest (Breiman 2001), and multilayer perceptrons (Rosenblatt 1961). The hyperparameters used in our models are shown in the Appendix.

Measuring Fairness

While there exists no consensus for measuring fairness, researchers have proposed a variety of metrics, some focusing on representations and some on the predicted outcomes (Verma and Rubin 2018; Hutchinson and Mitchell 2019). We will therefore compare our method to competing methods using the following metrics: Pearson correlation, mutual information, discrimination, calibration, balance of classes, and accuracy of the inferred protected features. Due to space limitations, we leave mutual information out of our analysis in this paper, and do not compare calibration and balance of classes to model accuracy. Results in all cases are similar.

Fairness of Outcomes One can argue that outcomes are fair if they do not depend on the protected features. If this is the case, a malicious adversary won't be able to guess the protected features from the model's predictions. One way to quantify the dependence is through *Pearson correlation* between (real valued or cardinal) predictions and protected features. For models making binary predictions, fairness can be measured using the *mutual information* between predictions and the protected features, given that protected features are discrete. We find mutual information and Pearson correlations create similar findings, therefore we focus on Pearson correlations in this paper. Previous work (Zemel et al. 2013) has also defined a *discrimination metric* for binary predictions as below. Consider a protected variable p_1, a binary prediction $\hat{y}$ of an outcome y. The metric measures the bias of a binary prediction $\hat{y}$ with respect to a single binary protected feature p_1 using the difference of positive rates between the two groups.

$$y\text{Discrim} = \left| \frac{\sum_{n:p_1[n]=0} \hat{y}[n]}{\sum_{n:p_1[n]=0} 1} - \frac{\sum_{n:p_1[n]=1} \hat{y}[n]}{\sum_{n:p_1[n]=1} 1} \right| \quad (5)$$

For real valued predictions ($\hat{y} \in [0, 1]$), Kleinberg, Mullainathan, and Raghavan (2016) suggested a more nuanced way to measure fairness:

- **Calibration within groups**: Individuals assigned predicted probability $\hat{y} \in [r_0 - \delta, r_0 + \delta]$, ($\delta > 0$ and $\delta \ll 1$) should have an approximate positive rate of r. This should hold for both protected groups ($p_1 = 0$ and $p_1 = 1$).
- **Balance for the negative class**: The mean $\hat{y}$ of group $p_1 = 0, y = 0$ and group $p_1 = 1, y = 0$ should be the same.
- **Balance for the positive class**: The mean $\hat{y}$ of group $p_1 = 0, y = 1$ and group $p_1 = 1, y = 1$ should be the same.

In some cases, calibration error is difficult to calculate, as it depends on how predictions are binned. In these cases, we can measure calibration error using log-likelihood of the labels given the real valued predictions as a proxy. By definition, logistic regression maximizes the (log-)likelihood function, assuming the observations are sampled from independent Bernoulli distributions where $P(y[n]|X[n]) = \hat{y}_i[n]$. Better log-likelihood implies that the individuals assigned probabilities $\hat{y} \in [r_0 - \delta, r_0 + \delta]$ are more likely to have a positive rate r, which is better calibrated according to Kleinberg, Mullainathan, and Raghavan.

Fairness of Representations Several past studies examined the fairness of representations, arguing that models using fair representations will also make fair predictions. Learned representations are considered fair if they do not reveal any information about the protected features (Jaiswal et al. 2018; Moyer et al. 2018; Louizos et al. 2015; Xie et al. 2017; Verma and Rubin 2018). The studies trained a discriminator to predict protected features from the learned representations—using accuracy as a measure of fairness.

Following this approach, we treat the predicted probabilities as a 1-dimensional representation of data and use the *accuracy of the inferred protected features* as a measure of fairness. However, this method is not effective in situations where the protected classes are unbalanced. Let us assume the fair representation is R and the protected feature is p_1. For simplicity, we only consider the case of a single binary protected feature. The discriminator infers the protected feature in a Bayesian way, namely,

$$P(p_1 = c|R) = \frac{P(R|p_1 = c)P(p_1 = c)}{P(R)}, c = 0|1 \quad (6)$$

In the case where there is a large difference between $P(p_1 = 0)$ and $P(p_1 = 1)$, even if there is useful information in the distribution $P(R|p_1 = c)$, the discriminator will not perform significantly better than the baseline model, the majority class classifier.

Results

In this section, we demonstrate how our method can achieve fair classification using synthetic data, and then compare our prediction accuracy and fairness to other fair AI algorithms using benchmark datasets.

Synthetic Data

We create synthetic biased data using the procedure described in (Zafar et al. 2016). We generate data with one binary protected variable s, one binary outcome y, and two continuous features, x_1 and x_2, which are bivariate Gaussian distributed within each value of s. In the Fig. 1, we use color to represent protected feature values (red, blue) and outcome using symbol ($\times$, $\circ$). The first observation is that there is an imbalance in the joint distribution of the protected features and the outcome variable. For blue color markers, there are more blue $\circ$s than blue $\times$s. We expect that a logistic classifier trained on this data will show similar unbalanced behavior. To demonstrate our method, we choose two different fairness levels, $\lambda = \{0.0, 1.0\}$. We first transform the two features into their corresponding fair representations and then we train logistic classifiers using these fair representations. In Fig. 1, we plot the data using the fair representations and we show the classification boundary using a green dashed line. We can observe that for $\lambda = 0$, the blue markers and red markers are mixed (less discrimination and bias), but for $\lambda = 1.0$ (equivalent to raw data), the blue and red markers tend to separate from each other. We can estimate this imbalance by comparing the ratio of blue in individuals predicted as $\circ$ and the ratio of blue in individuals predicted as $\times$. The larger the difference, the more the imbalance. Quantitatively, for $\lambda = 0.0$, there are 62.7% blue

in o-predictions and 52.9% in x-predictions. For $\lambda = 1.0$, those ratios are 76.2% and 36.5%. The accuracy of outcome predictions are 0.811 and 0.870 for the fair and original features, respectively, thus demonstrating that, while increasing fairness does indeed sacrifice in accuracy, the loss can be relatively small. Overall, the results suggest that biased data creates biased models, but our method can make fairer models.

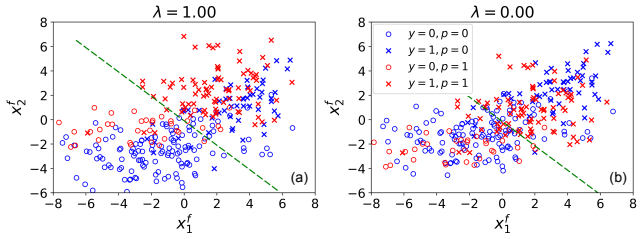

Figure 1: Fair synthetic data. (a) raw data ($\lambda = 1.0$), (b) plot for fairness level $\lambda = 0.0$. The two features in the data are x_1^f and x_2^f, and the two classes we want to protect are in red and blue. The two outcome classes are represented as two symbols: $\times$ and $\circ$.

Real-World Data

We compare our fair logistic model to state-of-the-art methods on three real-world datasets, which have become benchmarks for fair AI.

German dataset has 61 features about 1,000 individuals, with a binary outcome variable denoting whether an individual has a good credit score or not. The protected feature is gender. (https://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data))

COMPAS dataset contains data about 6,172 defendants. The binary outcome variable denotes whether the defendant will recidivate (commit a crime) within two years. The protected feature is race (whether the race is African American or not), and there are nine features in total. (https://github.com/propublica/compas-analysis)

Adult dataset contains data about 45,222 individuals. The outcome variable is binary, denoting whether an individual has more than $50,000. The protected feature is age, and there are 104 features in total. (https://archive.ics.uci.edu/ml/datasets/Adult)

Debiased features had mean correlations of 0.993 (90% quantiles: 0.954–0.999), 0.994 (90% quantiles: 0.980–0.999), and 0.994 (90% quantiles: 0.948–1.000), for the German, COMPAS, and Adult data, respectively. We reserved 20% of the data in the Adult and COMPAS datasets for testing and used the remaining data to perform 5-fold cross validation. This ensured no leakage of information from the training set to the testing set. The German dataset is much smaller than the rest, so it was randomly divided into five folds of training, validation and testing sets. Each set had 50%, 20% and 30% of all the data. We measured the performance metrics on the test data.

We varied the fairness parameter λ between 0 and 1 and applied the debiased features to logistic regression, AdaBoost, NuSVM, random forest, and multilayer perceptrons. In practice, one could use a host of commercial ML models and pick the most accurate one given their fairness tolerance.

Comparison Against State-of-the-Art

We compared our method to several previous fair AI algorithms. For the models proposed by (Zafar et al. 2015; 2016), we vary the fairness constraints from perfect fairness to unconstrained. For the "Unified Adversarial Invariance" (UAI) model proposed by (Jaiswal et al. 2018), we vary the δ term in the loss function from 0 (no fairness) to very large value, e.g., 9.0×10^{19} for COMPAS dataset, (large δ value corresponds to perfect fairness). The predictions of the UAI model for the German and Adult dataset are provided by the authors. We are interested in (1) how different models trade off between accuracy and fairness and (2) how different metrics of fairness compare to each other.

Fairness Versus Accuracy We first investigate the tradeoffs between prediction accuracy ($Acc\ Y$) and fairness, which we measure three different ways: (1) Pearson correlation between the protected feature and model predictions, (2) discrimination between the binary protected feature and the binarized predictions (predicted probabilities above 1/2 are given a value of 1, and are otherwise 0) and (3) the accuracy of predicting protected features from the predictions ($Acc\ P$). To robustly predict the protected features from the model predictions, we used both a NN with three hidden layers, which is used by former works (Jaiswal et al. 2018; Moyer et al. 2018; Louizos et al. 2015; Xie et al. 2017; Zemel et al. 2013) and a random forest model. We report the better accuracy of those two models. Figure 2,3 and 4 shows the resulting comparisons.

The figures show that models using the proposed fair features achieve significantly higher accuracy—for the same degree of fairness—compared to competing methods. In Fig. 4, we find $Acc\ P$ shows little difference from the baseline majority class classifier for the German and Adult datasets. The reason is explained in Eq.(6). On the other hand, $Acc\ P$ of COMPAS dataset shows a clear trend because the majority baseline is around 0.51, which is consistent with the Eq.(6). For the Adult dataset, the fair logistic regression cannot achieve perfect fairness but the situation is improved by AdaBoost. We discover, in other words, that there is no single ML model that achieves greater accuracy for a given value of fairness, but our method allows us to choose suitable models to achieve greater accuracy.

Fairness of Representations We further compared our method with previous works on fair representations. As mentioned before, some previous works use NNs to encode the features into a high dimensional embedding space and then use separately trained discriminators to infer the protected feature and the outcome variable. The accuracy of inferring protected feature and outcome are reported. Ideally, the accuracy for the outcome should be high and the accuracy of inferring the protected features should be close to the majority class baseline. We set the fairness level to $\lambda = 0$, namely perfect fairness when comparing to previous works. We show $Acc\ P$ and $Acc\ Y$ for various methods in Table 1

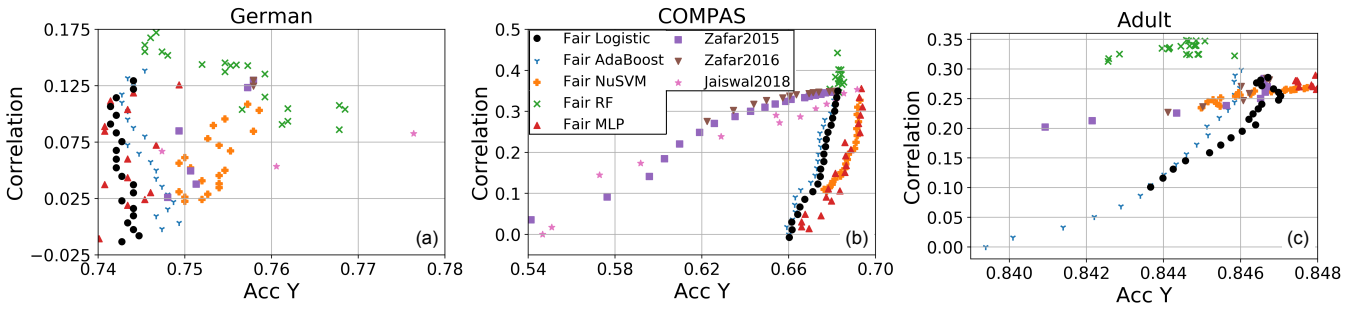

Figure 2: Fairness versus accuracy. Plots show Pearson correlation versus accuracy of predictions (*Acc Y*) for the German, COMPAS and Adult datasets. For each plot, *Zafar2015* stands for "Fair Constraints" (Zafar et al. 2015), *Zafar2016* stands for "Fairness Beyond Disparate Treatment & Disparate Impact" (Zafar et al. 2016) and *Jaiswal2018* stands for "Unified Adversarial Invariance" method (Jaiswal et al. 2018). *Fair NuSVM*, *Fair RF*, *Fair AdaBoost*, and *Fair MLP* results are produced using the fair representations constructed by our proposed method with NuSVM (Chang and Lin 2011), random forest (Breiman 2001), AdaBoost (Freund and Schapire 1997), and multilayer perceptrons (Rosenblatt 1961) models, respectively. The results of UAI are not shown for the Adult dataset, since its best accuracy (0.83) lies outside of the boundary of the plot. (Same for Figure 3 and 4.)

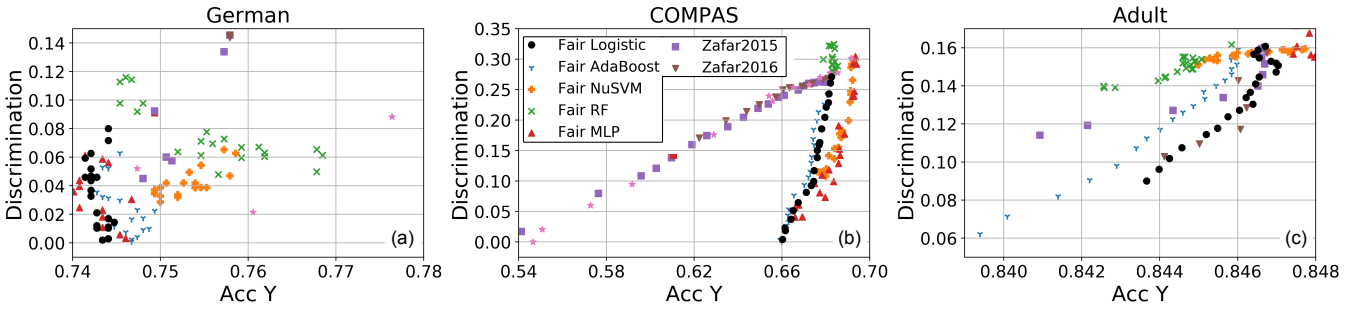

Figure 3: Discrimination versus accuracy plots for the three datasets.

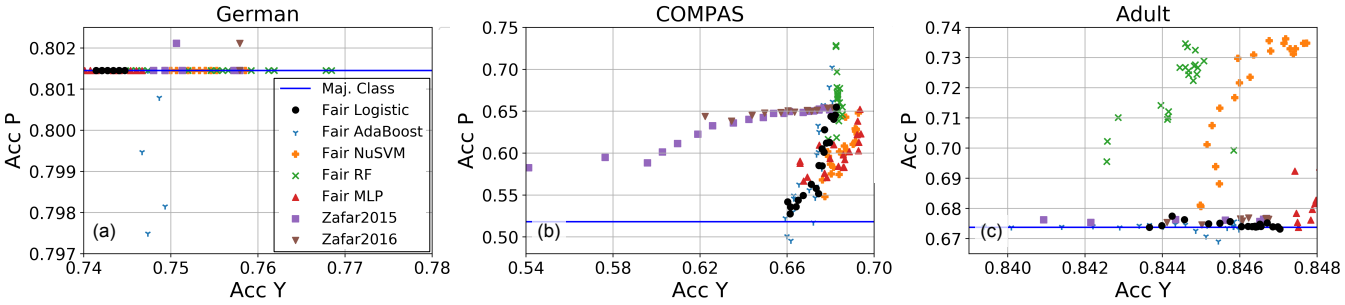

Figure 4: Accuracy of inferring the protected variable from the model's predictions ($Acc\ P$) versus the accuracy of predicting the outcome ($Acc\ Y$) for the three datasets.

and Fig. 4. Our method applied to a logistic model has similar fairness to the best existing methods but is very fast, easy to understand, and creates more interpretable features.

Balance Versus Calibration Finally, we use another measure of fairness that captures the degree to which each model makes mistakes. Figure 5 shows delta score (i.e., balance) versus negative log-likelihood (i.e., calibration error). Fairer predictions are located in the lower left hand corner of each figure, meaning that there are fewer differences in outcomes for the different classes. We only compare the logistic model with fair features to the models proposed by Zafar et al. (Zafar et al. 2015; 2016), because these models maximize the log-likelihood function (minimize calibration error) when selecting parameters. For all datasets, our method generally achieves greater fairness.

Conclusion

We show that our algorithm simultaneously achieves three advances over many previous fair AI algorithms. First, it is interpretable; the features we construct are minimally

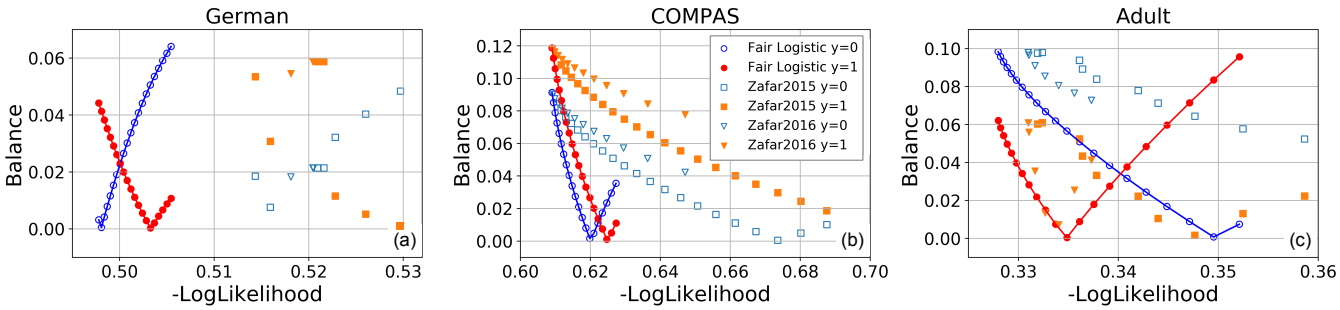

Figure 5: Balance vs. negative log-likelihood (calibration error) for the German, COMPAS and Adult datasets. In the plot, there are two sets of curves for every model, labeled $y = 0$ and $y = 1$. $y = 0$ stands for the difference of mean $\hat{y}$ (between different protected classes) given to the individuals with negative $y = 0$, and $y = 1$ stands for individuals with positive outcomes $y = 1$. (These differences are called balance of negative or positive class by (Kleinberg, Mullainathan, and Raghavan 2016).) Fairer models are those in the lower-left hand corner of each plot.

	German		*Adult*	
Method	*Acc Y*	*Acc P*	*Acc Y*	*Acc P*
Maj. Class	0.71	0.80	0.75	0.67
Li (2014) *	0.74	**0.80**	0.76	**0.67**
VFAE (2015) *	0.73	0.70	0.81	**0.67**
Xie (2017) *	0.74	**0.80**	0.84	**0.67**
Moyer (2018) *	0.74	0.60	0.79	0.69
Jaiswal (2018) *	**0.78**	**0.80**	0.84	**0.67**
Fair Logistic	0.74	**0.80**	0.84	**0.67**
Fair NuSVM	0.75	**0.80**	**0.85**	0.73
Fair AdaBoost	0.75	**0.80**	0.84	**0.67**
Fair RF	0.75	**0.80**	**0.85**	0.72
Fair MLP	0.75	**0.80**	**0.85**	**0.67**

Table 1: Accuracy of predicted outcomes (*Acc Y*) and protected features (*Acc P*) for the German and Adult datasets. The proposed fair methods (bottom four rows) use $\lambda = 0.0$. Higher *Acc Y* indicates better predictions while *Acc P* closer to the majority class baseline indicates fairer predictions. Results marked * were reported by (Jaiswal et al. 2019). Best performance is shown in bold.

affected by our fair transform. While this does not mean the models trained on these features are interpretable (they could be a black box), it does mean that any method used to interpret features could easily be used for these fairer features as well. Next, the features preserve model accuracy. Namely, models using these features were more accurate than competing methods when the value of the fairness metric was held fixed. This is in part due to the third principle: that our method can be applied to any number of commercial models; it merely acts as a pre-processing step. Different models have different strengths and weaknesses; while some are more accurate, others are fairer. We can pick and choose particular models that achieve both high fairness and accuracy, whether it is a linear model like logistic regression or a non-linear model like a multilayer perceptron, as shown in Figs. 2, 3, & 4.

Importantly, we only remove linear correlations between each feature and the protected features. While this works very well in practice, and beats state-of-the-art models, the fairness could be improved by removing non-linear correlations. Second, we can extend our method to more easily address categorical protected variables. In the present method, a categorical variable with alphabet size n becomes a set of $n - 1$ bivariate variables, It would be ideal, however, if a method reduced the mutual information between the categorical variable directly, rather than first creating $n - 1$ variables, and removed correlations.

Acknowledgements

Authors would like to thank Ayush Jaiswal for providing the code for learning adversarial models and feedback on results. Authors also thank Daniel Moyer and Greg Ver Steeg for insightful discussions about the approach. This material is based upon work supported in part by the Defense Advanced Research Projects Agency (DARPA)) under Contracts No. W911NF-18-C-0011. This research is also based upon work supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via 2017-17071900005. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

Appendix

Here are the hyperparameters used for modeling the empirical datasets. All models were trained using the *sklearn* library in Python 3.

References

Angwin, J.; Larson, J.; Mattu, S.; and Kirchner, L. 2016. Machine Bias.

Data	Model	Hyperparameters
German	Adaboost	Logit model, $C = 0.3$, L_2 penalty, 100 estimators (max)
	MLP	64-node, 3-layer network, $\alpha = 5.68$
	Logit	$C = 0.10$, L_2 penalty
	Random Forest	100 trees, max depth$= 15$
	NuSVM	$\nu = 0.51$, Radial basis function kernel
COMPAS	Adaboost	Logit model, $C = 0.6$, L_2 penalty, 100 estimators (max)
	MLP	64-node, 3-layer network, $\alpha = 0.43$
	Logit	$C = 20$, L_2 penalty
	Random Forest	100 trees, max depth$= 6$
	NuSVM	$\nu = 0.69$, Radial basis function kernel
	UAI	predictor loss weight$= 2.5$, decoder loss weight$= 0.0025$, disentangler loss weight$= 1.0$, epochs$= 200$
Adult	Adaboost	Logit model, $C = 10^4$, L_2 penalty, 100 estimators (max)
	MLP	64-node, 3-layer network, $\alpha = 0.71$
	Logit	$C = 2$, L_2 penalty
	Random Forest	100 trees, max depth$= 15$
	NuSVM	$\nu = 0.32$, Radial basis function kernel

Table 2: Hyperparamters for each dataset.

Berk, R.; Heidari, H.; Jabbari, S.; Joseph, M.; Kearns, M.; Morgenstern, J.; Neel, S.; and Roth, A. 2017. A Convex Framework for Fair Regression. 1–15.

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

Chang, C.-C., and Lin, C.-J. 2011. Libsvm: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)* 2(3):27.

Chen, I.; Johansson, F. D.; and Sontag, D. 2018. Why Is My Classifier Discriminatory?

Chouldechova, A., and Roth, A. 2018. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*.

Chouldechova, A. 2017. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big data* 5(2):153–163.

Ciregan, D.; Meier, U.; and Schmidhuber, J. 2012. Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3642–3649.

Danziger, S.; Levav, J.; and Avnaim-Pesso, L. 2011. Extraneous factors in judicial decisions. *Proceedings of the National Academy of Sciences* 108(17):6889–6892.

Dressel, J., and Farid, H. 2018. The accuracy, fairness, and limits of predicting recidivism. *Science advances* 4(1):eaao5580.

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*, 214–226. ACM.

Freund, Y., and Schapire, R. E. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences* 55(1):119–139.

Hutchinson, B., and Mitchell, M. 2019. 50 years of test (un) fairness: Lessons for machine learning. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 49–58. ACM.

Jaiswal, A.; Wu, Y.; AbdAlmageed, W.; and Natarajan, P. 2018. Unsupervised Adversarial Invariance. In Bengio, S.; Wallach, H.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc. 5092–5102.

Jaiswal, A.; Wu, Y.; AbdAlmageed, W.; and Natarajan, P. 2019. Unified Adversarial Invariance. 1–16.

Johndrow, J. E., and Lum, K. 2017. An algorithm for removing sensitive information: application to race-independent recidivism prediction. 1–25.

Kamiran, F.; Calders, T.; and Pechenizkiy, M. 2010. Discrimination Aware Decision Tree Learning. In *2010 IEEE International Conference on Data Mining*, 869–874.

Kleinberg, J.; Mullainathan, S.; and Raghavan, M. 2016. Inherent Trade-Offs in the Fair Determination of Risk Scores. 1–23.

Li, Y.; Swersky, K.; and Zemel, R. 2014. Learning unbiased features. 1(2):1–8.

Liu, B.; Wei, Y.; and amd Qiang Yang, Y. Z. 2017. Deep neural networks for high dimension, low sample size data. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI-17)*, 2287–2293.

Locatello, F.; Abbati, G.; Rainforth, T.; Bauer, S.; Schölkopf, B.; and Bachem, O. 2019. On the fairness of disentangled representations. *arXiv preprint arXiv:1905.13662*.

Louizos, C.; Swersky, K.; Li, Y.; Welling, M.; and Zemel, R. 2015. The Variational Fair Autoencoder. 1–11.

Mani, A.; Mullainathan, S.; Shafir, E.; and Zhao, J.

2013. Poverty impedes cognitive function. *science* 341(6149):976–980.

Moyer, D.; Gao, S.; Brekelmans, R.; Steeg, G. V.; and Galstyan, A. 2018. Invariant Representations without Adversarial Training. (Nips).

Olfat, M., and Aswani, A. 2018. Convex Formulations for Fair Principal Component Analysis.

Olson, M.; Wyner, A. J.; and Berk, R. 2018. Modern neural networks generalize on small data sets. In *NIPS'18 Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 3623–3632.

O'neil, C. 2016. *Weapons of math destruction: How big data increases inequality and threatens democracy*. Broadway Books.

Pierson, E.; Simoiu, C.; Overgoor, J.; Corbett-Davies, S.; Ramachandran, V.; Phillips, C.; and Goel, S. 2017. A large-scale analysis of racial disparities in police stops across the United States. *preprint arXiv:1706.05678*.

Rosenblatt, F. 1961. *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Washington DC: Spartan Books.

Samadi, S.; Tantipongpipat, U.; Morgenstern, J.; Singh, M.; and Vempala, S. 2018. The Price of Fair PCA: One Extra Dimension. (Nips).

Shah, A. K.; Mullainathan, S.; and Shafir, E. 2012. Some consequences of having too little. *Science* 338(6107):682–685.

Verma, S., and Rubin, J. 2018. Fairness definitions explained. In *2018 IEEE/ACM International Workshop on Software Fairness (FairWare)*, 1–7. IEEE.

Xie, Q.; Dai, Z.; Du, Y.; Hovy, E.; and Neubig, G. 2017. Controllable invariance through adversarial feature learning. *Advances in Neural Information Processing Systems* 2017-December(Mmd):586–597.

Zafar, M. B.; Valera, I.; Rodriguez, M. G.; and Gummadi, K. P. 2015. Fairness Constraints: Mechanisms for Fair Classification. 54.

Zafar, M. B.; Valera, I.; Rodriguez, M. G.; and Gummadi, K. P. 2016. Fairness Beyond Disparate Treatment & Disparate Impact: Learning Classification without Disparate Mistreatment.

Zafar, M. B.; Valera, I.; Rodriguez, M. G.; Gummadi, K. P.; and Weller, A. 2017. From Parity to Preference-based Notions of Fairness in Classification. In *NIPS*.

Zemel, R.; Wu, Y.; Swersky, K.; Pitassi, T.; and Dwork, C. 2013. Learning Fair Representations. In Dasgupta, S., and McAllester, D., eds., *Proceedings of the 30th International Conference on Machine Learning*, Proceedings of Machine Learning Research, 325–333. Atlanta, Georgia, USA: PMLR.