Repairing without Retraining: Avoiding Disparate Impact with Counterfactual Distributions

Hao Wang ¹ Berk Ustun ¹ Flavio P. Calmon ¹

Abstract

When the performance of a machine learning model varies over groups defined by sensitive attributes (e.g., gender or ethnicity), the performance disparity can be expressed in terms of the probability distributions of the input and output variables over each group. In this paper, we exploit this fact to reduce the disparate impact of a fixed classification model over a population of interest. Given a black-box classifier, we aim to eliminate the performance gap by perturbing the distribution of input variables for the disadvantaged group. We refer to the perturbed distribution as a counterfactual distribution, and characterize its properties for common fairness criteria. We introduce a descent algorithm to learn a counterfactual distribution from data. We then discuss how the estimated distribution can be used to build a data preprocessor that can reduce disparate impact without training a new model. We validate our approach through experiments on real-world datasets, showing that it can repair different forms of disparity without a significant drop in accuracy.

1. Introduction

A machine learning model has disparate impact when its performance changes across groups defined by sensitive attributes such as race or gender (Barocas & Selbst, 2016). Recent work has shown that models can exhibit significant performance disparities between groups (see e.g. Angwin et al., 2016; Dastin, 2018). Such disparities have led to a plethora of research on fair machine learning, focusing on how disparate impact arises (Chen et al., 2018; Datta et al., 2016), how it can be measured (Žliobaitė, 2017; Pires et al., 2017; Simo et al., 2017; Kilburt et al., 2017; Kusner et al., 2017; Galhotra et al., 2017), and how it can be mitigated (Feldman et al., 2015; Corbett-Davies et al., 2017; Zafar et al., 2017; Calmon et al., 2017; Menon & Williamson, 2018; Canetti et al., 2019). Despite these developments, disparate impact remains difficult to avoid in a large class of real-world applications where:

- Models are procured from a third-party vendor who has the data or technical expertise required for model development (Guszcza et al., 2018).
- Models are deployed on a population where the data distribution differs from the distribution of the training data (i.e., due to dataset shift, Sugiyama et al., 2017).

In such settings, disparate impact is challenging to address, let alone understand. Users typically have black-box access to the model (e.g., via a prediction API), may not have access to the training data (e.g., due to privacy concerns or intellectual property rights), and may not be able to draw conclusions from the training data (e.g., due to distributional shifts in deployment).

In this paper, we aim to mitigate disparate impact in such settings. Our object of interest is a hypothetical distribution of input variables that minimizes disparate impact in the model’s deployment population. We refer to this distribution as a counterfactual distribution. As we will show, an information-theoretic analysis of counterfactual distributions has much to offer. Given a fixed classifier, disparate impact can be expressed in terms the distributions of input and output variables between groups (cf. Figure 1). In turn, a counterfactual distribution can be obtained by repeatedly perturbing the distribution of input variables until a specific measure of disparity is minimized over the deployment population. The counterfactual distribution can then be used to repair the model so that it no longer exhibits disparate impact in deployment.

Contributions The main contributions of this paper are:

1. We introduce a theoretical framework to mitigate performance disparities for a black-box classifiers in deployment.
2. We develop machinery to learn counterfactual distributions from data. Our tools recover a counterfactual...
Avoiding Disparate Impact with Counterfactual Distributions

Figure 1: Illustration of probability distributions affecting the disparate impact of a fixed classification model \( h \). Here, \( P_0 \) and \( P_1 \) denote the distributions of input variables for groups where \( S = 0 \) and \( S = 1 \), respectively. Disparate impact is a function of the distributions of predicted outcomes \( (P_{Y|S=0}, P_{Y|S=1}) \) and true outcomes \( (P_{Y|S=0}, P_{Y|S=1}) \). A counterfactual distribution \( Q_{X} \) is a perturbation of \( P_0 \) that minimizes a specific measure of disparity. If disparate impact persists under \( Q_{X} \), there may be irreconcilable differences between the groups (i.e., \( P_{Y|X,S=0} \neq P_{Y|X,S=1} \), see Prop. 1). The counterfactual distribution may not be unique, as illustrated by the shaded ellipse.

1. A rural clinic purchases a classification model to detect bone fractures in x-rays and discovers that patients with a certain physical trait have high FPR.
2. A bank enters a new market and discovers its credit score underperforms on customers over 60 years of age.

The tools in this paper will be able to scrutinize and repair performance disparities in all three settings — regardless of whether the model directly uses the sensitive attribute. More importantly, they will be able to achieve parity-based notions of fairness in a way that: (i) only affects one group; (ii) benefits the group it affects (on average); and (iii) incentivizes individuals to reveal their sensitive attributes at prediction time. The latter two points (i.e., do-no-harm and opt-in) are important elements of ethical treatment disparity (see e.g., Lipton et al., 2018; Ustun et al., 2019a).

Related Work We develop a theoretical framework that is used to design methods to determine counterfactual distributions in practice. We then use counterfactual distributions to design optimal transport-based pre-processing methods for ensuring fairness. In this regard, the closest work to ours are those of Feldman et al. (2015); Johndrow & Lum (2017); Del Barrio et al. (2019), which propose methods to control specific disparate impact metrics via optimal transport. These methods differ from ours in that they (i) focus on reducing measures of disparity related to predicted outcomes; (ii) map the input variable distributions across all sensitive groups to a common distribution. More broadly, our approach differs from other model-agnostic approaches to mitigate disparate impact (e.g., pre-processing methods such as Kamiran & Calders, 2012; Calmon et al., 2017) in that it does not require access to the training data, and does not require training a new model.

The term “counterfactual distribution” is often used to describe different kinds of hypothetical effects. In the statistics and economics literature (see e.g., Balke & Pearl, 1994; DiNardo et al., 1995; Rubin, 2005; Fortin et al., 2011; Chernozhukov et al., 2013; Johansson et al., 2016; Peters et al., 2017; Fisher & Kennedy, 2018), a counterfactual distribution refers to a hypothetical distribution of an outcome variable given a specific distribution of input variables (e.g., the distribution of wages (outcome variable) for young workers if young workers had the same qualifications as older workers). The counterfactual distribution in this work describes a different kind of effect — i.e., a distribution of input variables to minimize disparate impact — and, consequently, must be derived using a different set of tools.

Use Cases Our approach provides a way to repair classifiers in domains where treatment disparity is legal and ethical\(^1\). Illustrative use cases include:

- A hospital purchases a readmissions prediction model that satisfies equal opportunity for different genders in the training data but violates it in deployment;
- A bank enters a new market and discovers its credit score underperforms on customers over 60 years of age.

\(^1\)Treatment disparity is not illegal nor unethical in all applications of machine learning where fairness is important. In medicine, for example, it is acceptable to use sensitive attributes in prediction. In lending, the Equal Credit Opportunity Act permits the use of age in credit scoring models.

Additional Resources We provide a software implementation of our tools at http://github.com/ustunb/ctfdist. This paper extends work that was first presented at the NeurIPS WESGAI Workshop (Wang et al., 2018b).
2. Framework

In this section, we formally define counterfactual distributions and discuss their properties.

Preliminaries  We consider a standard classification task where the goal is to predict a binary outcome variable \( Y \in \{0, 1\} \) using a vector of input variables \( X = (X_1, \ldots, X_d) \in \mathcal{X} \) drawn from the probability distribution \( P_X \). We are given a black-box classifier \( h: \mathcal{X} \rightarrow [0, 1] \). We assume that \( h(x) \in \{0, 1\} \) if the classifier outputs a predicted outcome (e.g., SVM) and \( h(x) \in [0, 1] \) if it outputs a predicted probability (e.g., logistic regression).

We evaluate differences in the performance of the classifier with respect to a sensitive attribute \( S \in \{0, 1\} \) with distribution \( P_S \). We refer to the subset of individuals with \( S = 0 \) and \( S = 1 \) as the target and baseline groups, respectively. We denote their distributions of input variables as \( P_0 = P_{X|S=0} \), and \( P_1 = P_{X|S=1} \). Likewise, we let \( P_{Y|S=0}(1) = \mathbb{E}[h(X)|S = 0] \) and \( P_{Y|S=1}(1) = \mathbb{E}[h(X)|S = 1] \). For the sake of clarity, we assume that \( S \) is not an input variable to the model. Note, however, that our tools and results can be extended to settings where \( S \) is an input variable for the classifier.

Disparity Metrics  We measure the performance disparity between groups in terms of a *disparity metric*. Formally, a disparity metric is a mapping \( M: \mathcal{P} \rightarrow \mathbb{R} \) where \( \mathcal{P} \) is the set of probability distributions over \( \mathcal{X} \). We provide examples of \( M(P_0) \) for common fairness criteria in Table 1. Note that we write disparity metrics as \( M(P_0) \) since they can be expressed as a function \( P_0 \) once the classifier and the distributions \( P_{Y|X,S}, P_1, \) and \( P_S \) are fixed.\(^2\)

Counterfactual Distributions  A counterfactual distribution is a hypothetical probability distribution of input variables for the target group that minimizes a specific disparity metric.

**Definition 1.** A counterfactual distribution \( Q_X \) is a distribution of input variables for the target group such that:

\[
Q_X \in \arg\min_{Q_X \in \mathcal{P}} |M(Q_X')|, 
\tag{1}
\]

where \( M(\cdot) \) is a given disparity metric and \( \mathcal{P} \) is the set of probability distributions over \( \mathcal{X} \).

There exist several ways to resolve the performance disparity of a fixed classifier by perturbing the distributions of input variables of sensitive groups. For example, one could simultaneously perturb the input distributions for all groups to a “midpoint” distribution (see e.g., the distributions considered by Feldman et al., 2015; Johndrow & Lum, 2017; Del Barrio et al., 2019, to achieve statistical parity).

While our tools could recover such distributions, we will purposely consider a counterfactual distribution that alters the input variables for a sensitive group that attains the less favorable performance (i.e., the target group \( S = 0 \)). This choice reflects our desire to resolve the performance disparity by having the target group perform better, rather than having the baseline group perform worse. As we discuss later, this choice reduces the data requirements to estimate the counterfactual distribution and the individuals who are affected by the repair (i.e., this approach only produces a preprocessor that affects individuals where \( S = 0 \)).

At this point, an observant reader may wonder why a counterfactual distribution for the target group is not simply the distribution of input variables over the baseline group (i.e., \( Q_X \equiv P_1 \)). In fact, the distribution of input variables for the baseline group \( P_1 \) is not necessarily a counterfactual distribution when \( P_{Y|X,S=0} \neq P_{Y|X,S=1} \). We illustrate this point with the following example.

**Example 1.** Consider a classification task where the input variables \( X = (X_1, X_2) \in \{0, 1\}^2 \) are drawn from distributions such that \( P_{X|S=s} = P_{X_1|S=s} \cdot P_{X_2|S=s} \) for \( s \in \{0, 1\} \) where:

\[
\Pr(X_1 = 1|S = 0) = 0.9, \quad \Pr(X_2 = 1|S = 0) = 0.2, \\
\Pr(X_1 = 1|S = 1) = 0.1, \quad \Pr(X_2 = 1|S = 1) = 0.5.
\]

Assume that the true outcome variables \( Y \) are drawn from the conditional distributions:

\[
P_{Y|X,S=0}(1|x) = \logistic(2x_1 - 2x_2), \\
P_{Y|X,S=1}(1|x) = \logistic(2x_1 + 4x_2 - 3).
\tag{2}
\]

In this case, the Bayes optimal classifier for \( S = 1 \) is \( h(x) = 1[x_2 = 1] \). Using the difference in FPR as the disparity metric, \( h \) achieves \( M(P_0) = 25.1\% \). In this case, setting \( P_0 \leftarrow P_1 \) would achieve a disparity of \( M(P_1) = 43.6\% \). In contrast, we can achieve a disparity metric of \( M(Q_X) = 0.0\% \) for a counterfactual distribution such that:

\[
Q_X(0, 0) = 0.50, \quad Q_X(0, 1) = 0.09, \\
Q_X(1, 0) = 0.41, \quad Q_X(1, 1) = 0.00.
\]

Example 1 shows that counterfactual distributions may be non-trivial when the conditional distributions of \( Y \) given \( X \) differ across groups (i.e., \( P_{Y|X,S=0} \neq P_{Y|X,S=1} \)). In particular, the condition \( P_{Y|X,S=0} \neq P_{Y|X,S=1} \) will always hold whenever counterfactual distributions do not completely eliminate the disparity between groups. We formalize this statement in the next proposition.
We start by formally defining the local perturbation of an input distribution. The perturbed distribution \( \hat{P}_0 \) over the target group \( S = 0 \) is given by
\[
\hat{P}_0(x) = P_0(x)(1 + \epsilon f(x)), \quad \forall x \in \mathcal{X} \tag{3}
\]
where \( f : \mathcal{X} \to \mathbb{R} \) is a perturbation function from the class of all functions with zero mean and unit variance w.r.t. \( P_0 \), and \( \epsilon > 0 \) is a positive scaling constant chosen so that \( \hat{P}_0 \) is a valid probability distribution.

Here, \( f(x) \) represents a direction in the probability simplex while \( \epsilon \) represents the magnitude of perturbation (see e.g., Borade & Zheng, 2008; Anantharam et al., 2013; Huang et al., 2014, for other applications of local perturbations of measures in information theory).

As we will see shortly, the direction of steepest descent for disparate impact can be measured using an influence function (see e.g., Huber, 2011; Koh & Liang, 2017, for other uses in machine learning).

Definition 3. For a disparity metric \( M(\cdot) \), the influence function \( \psi : \mathcal{X} \to \mathbb{R} \) is given by
\[
\psi(x) \triangleq \lim_{\epsilon \to 0} \frac{M((1 - \epsilon)P_0 + \epsilon \delta_x) - M(P_0)}{\epsilon} \tag{4}
\]
where \( \delta_x(z) = \mathbb{I}[z = x] \) is the delta function at \( x \).

Intuitively, given a sufficiently large dataset from the deployment population, the influence function approximates the change in a disparity metric when a sample \( x \in \mathcal{X} \) from the target group is removed (or added) to the dataset.

In Proposition 2, we show that perturbing the distribution \( P_0 \) along the direction defined by \( -\psi(x) \) produces the largest local decrease of the disparity metric. That is, \( -\psi(x) \) reflects the direction of steepest descent in disparate impact.

Proposition 2. Given a disparity metric \( M(\cdot) \), we have that
\[
\arg \min_{f(x)} \lim_{\epsilon \to 0} \frac{M(\hat{P}_0) - M(P_0)}{\epsilon} = \frac{-\psi(x)}{\sqrt{\mathbb{E}[(\psi(x))^2 | S = 0]}} \neq 0 \tag{5}
\]
for any influence function \( \psi : \mathcal{X} \to \mathbb{R} \) such that \( \mathbb{E}[(\psi(x))^2 | S = 0] \neq 0 \).

Proposition 3 shows that when disparity is measured using a linear combination of metrics, the influence function for the compound metric can be expressed as a linear combination of the influence functions for its components.
We now present closed-form expressions for the influence we can then compute influence functions using closed-form

\[ \psi(x) = \sum_{i=1}^{K} \lambda_i \psi_i(x). \]  

Proposition 3 allows us to consider a larger class of disparity measures than those shown in Table 1. For instance, one can recover a counterfactual distribution to achieve equalized odds (Hardt et al., 2016) by using a convex combination of influence functions for FPR and FNR.

### 3.2. Computing Influence Functions

We now present closed-form expressions for the influence functions of disparity metrics shown in Table 1. The expressions will be cast in terms of three classifiers:

- \( h(x) \): the black-box classifier that we aim to repair;
- \( \hat{s}(x) \): a classifier that uses the same input variables as \( h \), but aims to predict the probability that an individual belongs to the baseline group, \( P_{S|X}(1|x) \);
- \( \hat{y}_0(x) \): a classifier that uses the same input variables as \( h \), but aims to predict the true outcome for individuals in the target group, \( P_{Y|X,S=0}(1|x) \).

Given \( h(x) \), we would train \( \hat{s}(x) \) and \( \hat{y}_0(x) \) using an auditing dataset \( D_{\text{audit}} = \{(x_i, y_i, s_i)\}_{i=1}^n \) drawn from the deployment population. With these three models in hand, we can then compute influence functions using closed-form expressions shown in the following proposition.

**Proposition 4.** The influence functions for the disparity metrics in Table 1 can be expressed as

\[
\begin{align*}
\psi_{SP}^D(x) &= -h(x) + \hat{\mu}_0, \\
\psi_{FDR}^D(x) &= h(x)(1 - \hat{y}_0(x)) - \nu_{0,1} h(x), \\
\psi_{FNR}^D(x) &= \frac{(1 - h(x))\hat{y}_0(x) - \gamma_{0,1}\hat{y}_0(x)}{\mu_0}, \\
\psi_{FPR}^D(x) &= h(x)(1 - \hat{y}_0(x)) - \gamma_{1,0}(1 - \hat{y}_0(x)), \\
\psi_{DA}^D(x) &= \log\left(\frac{\hat{\mu}_0(1 - \hat{\mu}_1)}{(1 - \hat{\mu}_0)\hat{\mu}_1}\right) h(x) + \lambda \log\left(1 - \frac{\hat{s}(x)}{\hat{s}(x)}\right) \\
&\quad - \hat{\mu}_0 \log\left(\frac{\hat{\mu}_0(1 - \hat{\mu}_1)}{(1 - \hat{\mu}_0)\hat{\mu}_1}\right) - \lambda \mathbb{E}\left[\log\left(1 - \frac{\hat{s}(X)}{\hat{s}(X)}\right) | S = 0\right],
\end{align*}
\]

where \( \mu_s, \hat{\mu}_s, \gamma_{a,b}, \) and \( \nu_{a,b} \) are constants such that

\[
\begin{align*}
\mu_s &\triangleq \Pr(Y = 1 | S = s), \\
\hat{\mu}_s &\triangleq \Pr(\hat{Y} = 1 | S = s), \\
\gamma_{a,b} &\triangleq \Pr(\hat{Y} = a | Y = b, S = 0), \\
\nu_{a,b} &\triangleq \Pr(Y = a | \hat{Y} = b, S = 0).
\end{align*}
\]

### 3.3. Learning Counterfactual Distributions from Data

So far we have shown that influence functions can be used to evaluate the direction of steepest descent of a disparity metric (Proposition 2), and that the value of an influence function can be estimated using data from the deployment population (Proposition 4).

Considering these results, one would expect that disparity could be minimized by repeatedly (i) perturbing the distribution in the direction of steepest descent (5), and (ii) estimating the influence function at the new, perturbed distribution. Repeating these steps, we would recover an approximate solution to (1) – i.e., an approximate counterfactual distribution.

In Algorithm 1, we formalize this intuition by presenting a descent procedure to recover a counterfactual distribution for a given disparity metric \( M(\cdot) \). The procedure is analogous to stochastic gradient descent in the space of distributions over \( \mathcal{X} \), where the resampling at each iteration corresponds to a gradient step.

**Algorithm 1 Distributional Descent.**

**Input:**

\[
\begin{align*}
&h : \mathcal{X} \rightarrow [0, 1] & \triangleright \text{classification model} \\
&D = \{(x_i, y_i, s_i)\}_{i=1}^n & \triangleright \text{data from deployment population} \\
&M(\cdot) & \triangleright \text{disparity metric} \\
&\epsilon > 0 & \triangleright \text{step size}
\end{align*}
\]

**Initialize**

\[
\begin{align*}
&I_0 \leftarrow \{i = 1, \ldots, n \mid s_i = 0\} & \triangleright \text{samples where } s_i = 0 \\
&D_0 \leftarrow \{(x_i, y_i)\} \text{ for } i \in I_0 & \triangleright \text{samples where } s_i \neq 0 \\
&w_0 \leftarrow \{w_i\}_{i \in I_0} \text{ where } w_i = 1.0 & \triangleright \text{initial weights} \\
&M \leftarrow M(D_0 \cup D_1) & \triangleright \text{evaluate disparity metric}
\end{align*}
\]

**repeat**

\[
\begin{align*}
&M^{\text{old}} \leftarrow M \\
&\psi_i \leftarrow \psi(x_i) \text{ for } i \in I_0 & \triangleright \text{compute } \psi(x_i) \text{ for points in } D_0 \\
&w_i \leftarrow (1 - \epsilon \psi_i) \cdot w_i \text{ for } i \in I_0 & \triangleright \text{step size} \\
&D_0 \leftarrow \text{RESAMPLE}(D_0, w_0) & \triangleright \text{resample} \\
&M \leftarrow M(D_0 \cup D_1) & \triangleright \text{evaluate disparity metric}
\end{align*}
\]

until \( M \geq M^{\text{old}} \)

**return:** \( w_0, D_0 \) \triangleright samples from counterfactual distribution

**procedure** \text{RESAMPLE}(D, w)

**return:** \( |D| \) points sampled from \( D \) using the weights \( w \)

**end procedure**

Given a classifier \( h \) and a dataset \( \{(x_i, y_i, s_i)\}_{i=1}^n \) from the distribution \( P_{X,Y,S} \), the procedure outputs a dataset drawn from a counterfactual distribution.

The procedure pairs each point with a sampling weight \( w_i \), which is initialized as \( w_i = 1.0 \). At each iteration, it first computes the value of the influence function \( \psi_i(x) \). Next, it updates the values of each sampling weight for each point in the target group as \( (1 - \epsilon \psi_i(x_i)) \cdot w_i \), where \( \epsilon \) is a
user-specified step size parameter. The updated sampling weights represent the direction in which the distribution for the target group should be perturbed to reduce $M(\cdot)$ on the set of resampled points. These steps are repeated until $M(\cdot)$ ceases to decrease. Once the procedure stops, it outputs: (i) dataset drawn from the perturbed distribution; (ii) a set of sampling weights for each point from the target group, which can be used to draw samples from the counterfactual distribution.

The procedure determines if the classifier still has disparate impact at the end of each iteration by computing the value of $M(\cdot)$ on the set of resampled points. These steps are repeated until $M(\cdot)$ ceases to decrease. Once the procedure stops, it outputs: (i) dataset drawn from the perturbed distribution; (ii) a set of sampling weights for each point from the target group, which can be used to draw samples from the counterfactual distribution.

In Figure 2, we show the progress (and convergence) of Algorithm 1 when recovering a counterfactual distribution for a synthetic dataset described in Appendix B.1.

4. Model Repair

In this section, we describe how to use counterfactual distributions to repair classifiers that exhibit disparate impact.

**Preprocessor** Given a classifier $h(x)$, we aim to mitigate disparate impact by constructing a preprocessor $T : \mathcal{X} \rightarrow \mathcal{X}$ that alters the features of the target group. Thus, the repaired classifier $\tilde{h}(x)$ will operate as:

$$\tilde{h}(x) = \begin{cases} h(T(x)) & \text{if } s = 0, \\ h(x) & \text{otherwise.} \end{cases}$$ (7)

The preprocessor is a (potentially randomized) mapping that transforms the distribution of samples over the target population into the counterfactual distribution, i.e., given a random variable $X$ drawn from the target population distribution, the distribution of $T(X)$ will approximate counterfactual distribution.

**Optimal Transport** We produce the preprocessor by solving an optimal transport problem. To this end, we require the following inputs:

- $D_0$, which represents the original samples for the target group. We assume $D_0$ contains $n_0$ samples, of which $m$ are distinct: $\{x_1, \cdots, x_m\}$.
- $\tilde{D}_0$, which represents the samples drawn from the counterfactual distribution (i.e., the data produced via resampling in Algorithm 1). We assume that $\tilde{D}_0$ contains $\tilde{n}_0$ samples, of which $\tilde{m}$ are distinct: $\{\tilde{x}_1, \cdots, \tilde{x}_{\tilde{m}}\}$.

With these samples at hand, we formulate an optimal transport problem of the form:

$$\min_{\gamma_{ij} \in \mathbb{R}^+} \sum_{i=1}^{m} \sum_{j=1}^{\tilde{m}} C_{ij} \gamma_{ij}$$

s.t. $\sum_{j=1}^{\tilde{m}} \gamma_{ij} = p_i \quad i = 1, \cdots, m$ \quad (8a)

$\sum_{i=1}^{m} \gamma_{ij} = q_j \quad j = 1, \cdots, \tilde{m}$. \quad (8b)

Here, $C_{ij}$ represents the cost of altering the input variables from $x_i$ to $\tilde{x}_j$ given a user-specified cost function that we will discuss shortly; $p, q$ are the empirical estimates of $P_0$ and $Q_X$, respectively,

$$p_i = \frac{1}{n_0} \sum_{x \in D_0} \delta_{x_i}(x), \quad q_j = \frac{1}{\tilde{n}_0} \sum_{\tilde{x} \in \tilde{D}_0} \delta_{\tilde{x}_j}(\tilde{x});$$

The optimal transport problem in (8) is a standard linear program that aims to find a coupling of $p$ and $q, \gamma$ (see e.g., Peyrè & Cuturi, 2017; Villani, 2008). Formally, a coupling is a joint probability distribution with marginals specified by $p$ and $q$. Given the minimal-cost coupling $\gamma^*$, one can construct a (randomized) preprocessor $T(\cdot)$ which takes a sample $x_i$ and returns an altered sample $\tilde{x}_j$ with probability $\gamma^*_{ij}/p_i$.

We note that the linear programming formulation in (8) is designed for settings with discrete input distributions. In settings when the distributions $P_0$ and $Q_X$ are continuous, an analogous optimal transport problem can be formulated and solved with other approaches (see e.g., Benamou & Brenier, 2000; Angenent et al., 2003).
Avoiding Disparate Impact with Counterfactual Distributions

Choice of Cost Function The cost function $C_{ij}$ controls how samples of the target group are perturbed. By default, one could use a standard distance metric such as the $L_2$-norm (e.g., Feldman et al., 2015; Johndrow & Lum, 2017; Del Barrio et al., 2019). However, one could also consider additional criteria to fine-tune the mapping specified by $T(\cdot)$. For example, one can specify a cost function that avoids specific kinds of change by setting the value of $C_{ij}$ to a large constant so as to penalize undesirable mappings (e.g., a mapping that would alter immutable attributes such as marital status Ustun et al., 2019b).

Customization via Constraints Users can also fine-tune the behavior of the preprocessor by adding custom constraints to the feasible region of optimal transport problems as in (8). For example, one can impose constraints on individual fairness to ensure that the repaired classifier will “treat similar individuals similarly” (see e.g., Dwork et al., 2012). This behavior could be induced by including constraints of the form:

$$\frac{1}{2} \sum_{j=1}^{m} \left| \gamma_{i,j} - \tilde{\gamma}_{i,j} \right| \leq d(x_i, \tilde{x}_i) \quad \text{for all } i, l \in [m].$$

Here, the LHS is the total-variation distance (Cover & Thomas, 2012) between the distributions of $T(x_i)$ and $T(\tilde{x}_i)$, and $d(\cdot, \cdot) : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$ is a distance metric that reflects the similarity between samples.

5. Experiments

In this section, we demonstrate how counterfactual distributions can be used to avoid disparate impact for classifiers on real-world datasets. We include all datasets and scripts to reproduce our results at http://github.com/ustunb/ctfdist.

Setup We aim to recover counterfactual distributions for different disparity metrics in Table 1. To this end, we consider processed versions of the adult dataset (Bache & Lichman, 2013) and the ProPublica compas dataset (Angwin et al., 2016).

For each dataset, we use:

- 30% of samples to train a classifier $h(x)$ to repair;
- 50% of samples to recover a counterfactual distribution via Algorithm 1;
- 20% of samples as a hold-out set to evaluate the performance of the repaired model.

We use $\ell_2$-logistic regression to train a classifier $h(x)$ as well as the classifiers $\tilde{g}_0(x)$ and $\tilde{s}(x)$ that we use to estimate the influence functions in Algorithm 1. We tune the parameters and estimate the performance of each classifiers using a standard 10-fold CV setup.

Our setup assumes that the data used to train and repair the model are drawn from the same distribution, which may not be the case in settings with dataset shift. Our setup also differs from real-world settings in that we use 70% of the samples in each dataset to estimate the counterfactual distribution. In practice, however, we would use all available samples since we would be given the classifier to repair.

Discussion In Table 2, we show the effectiveness of preprocessors built to mitigate different kinds of disparity for the classifiers trained on the adult and compas datasets.

We build each preprocessor as follows. We first resample data from the target population according to Algorithm 1. This outputs a dataset of samples drawn from the counterfactual distribution. Next, we use the resampled dataset to produce an empirical estimate of the counterfactual distribution $Q_X$. This distribution is then used to obtain the preprocessor by solving a version of (8) with the cost function $C_{ij} = \|x_i - \tilde{x}_j\|_2^2$.

As shown, the approach reduces disparate impact in the target group, while having a minor effect on test accuracy at various decision points across the full ROC curve. Counterfactual distributions provide a way to scrutinize this mapping in greater detail. As shown in Table 3, one can visualize the differences between the observed distribution and counterfactual distribution to understand how the input variable distributions are altered to reduce disparity.

This kind of constrastive analysis may be helpful in understanding the factors that produce performance disparities in the first place. For example, the differences between the observed distribution $P_0$ and the counterfactual distribution $Q_X$ could be used to identify prototypical samples (see e.g, Bien & Tibshirani, 2011; Kim et al., 2016), or to score features in terms of their ability to produce disparities in the deployment population (Datta et al., 2016; 2017; Adler et al., 2018).

6. Conclusion

We have introduced a new distributional paradigm to understand and mitigate disparate impact. Our framework is based on counterfactual distributions, which can be efficiently computed given a fixed model and data from a population of interest. The tools in this work apply to binary classification models. However, our approach can be extended to handle other kinds of supervised learning models.

Limitations Our approach requires collecting data on sensitive attribute, which may infringe privacy (though this is difficult to avoid as discussed in Žliobaitė & Custers, 2016).
Avoiding Disparate Impact with Counterfactual Distributions

<table>
<thead>
<tr>
<th>DATASET</th>
<th>METRIC</th>
<th>TARGET GROUP</th>
<th>ORIGINAL MODEL</th>
<th>REPAIRED MODEL</th>
<th>TARGET GROUP AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>BASELINE GROUP</td>
<td>TARGET GROUP</td>
<td>DISC. GAP</td>
</tr>
<tr>
<td>adult</td>
<td>SP</td>
<td>Female</td>
<td>0.696</td>
<td>0.874</td>
<td>0.178</td>
</tr>
<tr>
<td></td>
<td>FNR</td>
<td>Female</td>
<td>0.478</td>
<td>0.639</td>
<td>0.161</td>
</tr>
<tr>
<td>adult</td>
<td>FPR</td>
<td>Male</td>
<td>0.021</td>
<td>0.119</td>
<td>0.098</td>
</tr>
<tr>
<td>compas</td>
<td>SP</td>
<td>White</td>
<td>0.514</td>
<td>0.594</td>
<td>0.079</td>
</tr>
<tr>
<td>compas</td>
<td>FNR</td>
<td>White</td>
<td>0.350</td>
<td>0.487</td>
<td>0.137</td>
</tr>
<tr>
<td>compas</td>
<td>FPR</td>
<td>Non-white</td>
<td>0.190</td>
<td>0.278</td>
<td>0.087</td>
</tr>
</tbody>
</table>

Table 2: Change in disparate impact for classification models for adult and compas when paired with a randomized preprocessor built to mitigate different kinds of disparity. Each row shows the value of a specific performance metric for the classifier over the target and baseline groups (e.g., SP, FNR, and FPR). The target group is defined as the group that attains the less favorable value of the performance metric. The preprocessor aims to reduce to difference in performance metric by randomly perturbing the input variables for individuals in the target group. We also include AUC to show the change in performance due to the randomized preprocessor. All values are computed using a hold-out sample that is not used to train the model or build the preprocessor.

<table>
<thead>
<tr>
<th></th>
<th>OBSERVED</th>
<th>COUNTERFACTUAL</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Female</td>
<td>Male</td>
</tr>
<tr>
<td>Married</td>
<td>18%</td>
<td>63%</td>
</tr>
<tr>
<td>Immigrant</td>
<td>10%</td>
<td>11%</td>
</tr>
<tr>
<td>HighestDegree_is_HS</td>
<td>32%</td>
<td>32%</td>
</tr>
<tr>
<td>HighestDegree_is_AS</td>
<td>7%</td>
<td>8%</td>
</tr>
<tr>
<td>HighestDegree_is_BS</td>
<td>15%</td>
<td>18%</td>
</tr>
<tr>
<td>HighestDegree_is_MSorPhD</td>
<td>6%</td>
<td>7%</td>
</tr>
<tr>
<td>AnyCapitalLoss</td>
<td>3%</td>
<td>5%</td>
</tr>
<tr>
<td>Age ≤ 30</td>
<td>39%</td>
<td>29%</td>
</tr>
<tr>
<td>WorkHrsPerWeek&lt;40</td>
<td>38%</td>
<td>17%</td>
</tr>
<tr>
<td>JobType_is_WhiteCollar</td>
<td>34%</td>
<td>19%</td>
</tr>
<tr>
<td>JobType_is_BlueCollar</td>
<td>5%</td>
<td>34%</td>
</tr>
<tr>
<td>JobType_is_Specialized</td>
<td>23%</td>
<td>21%</td>
</tr>
<tr>
<td>JobType_is_ArmedOrProtective</td>
<td>1%</td>
<td>2%</td>
</tr>
<tr>
<td>Industry_is_Private</td>
<td>73%</td>
<td>69%</td>
</tr>
<tr>
<td>Industry_is_Government</td>
<td>15%</td>
<td>12%</td>
</tr>
<tr>
<td>Industry_is_SelfEmployed</td>
<td>5%</td>
<td>15%</td>
</tr>
</tbody>
</table>

Table 3: Counterfactual distributions produced using Algorithm 1 for a classifier on adult. We observe that different metrics produce different counterfactual distributions. By comparing the distribution of the target group with the counterfactual distribution, we can evaluate how the repaired classifier will perturb their features to reduce disparity.

Our approach also aims to mitigate performance using a randomized preprocessor. While randomization is a common technique to reduce disparate impact in the literature (see e.g., Hardt et al., 2016; Agarwal et al., 2018), it may not be practical in applications such as loan approval since an applicant could achieve a different predicted outcome by applying multiple times. Some effects of randomization can be mitigated by heuristic strategies. Given that we have considered counterfactual distributions that improve the performance for the target group, however, our approach does have a benefit in that randomization will only apply to individuals in the target group and only be applied in a way that will improve their outcomes.
Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. CCF-18-45852.

References


Avoiding Disparate Impact with Counterfactual Distributions


