Decoupled classifiers for fair and efficient machine learning

Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Max Leiserson

July 21, 2017

Abstract

When it is ethical and legal to use a sensitive attribute (such as gender or race) in machine learning systems, the question remains how to do so. We show that the naive application of machine learning algorithms using sensitive attributes leads to an inherent tradeoff in accuracy between groups. We provide a simple and efficient *decoupling* technique, that can be added on top of any black-box machine learning algorithm, to learn different classifiers for different groups. The method can apply to a range of fairness criteria. In particular, we require the application designer to specify as joint loss function that makes explicit the trade-off between fairness and accuracy. Our reduction is shown to efficiently find the global optimum loss as long as the objective has a certain natural *monotonicity* property. Monotonicity may be of independent interest in the study of fairness in algorithms.

Please see `http://arxiv.org/abs/1707.06613` for a full and up-to-date version of this manuscript.

1 Introduction

As algorithms are increasingly used to make decisions of social consequence, the social values encoded in these decision-making procedures are the subject of increasing study, with fairness being a chief concern (Pedreshi et al., 2008; Zliobaite et al., 2011; Kamishima et al., 2011; Dwork et al., 2011; Friedler et al., 2016; Angwin et al., 2016; Chouldechova, 2017; Joseph et al., 2016; Hardt et al., 2016; Kusner et al., 2017; Berk, 2009). *Classification and regression algorithms* are one particular locus of fairness concerns. Classifiers map individuals to outcomes: applicants to accept/reject/waitlist; adults to credit scores; web users to advertisements; felons to estimated recidivism risk. Informally, the concern is whether individuals are treated "fairly," however this is defined. Still speaking informally, there are many sources of unfairness, prominent among these being training the classifier on historically biased data and a paucity of data for under-represented groups leading to poor performance on these groups, which in turn can lead to higher risk for those, such as lenders, making decisions based on classification outcomes.

Should ML systems use sensitive attributes, such as gender or race if available? The legal and ethical factors behind such a decision vary by time, country, jurisdiction, and culture, and downstream application. Still speaking informally, it is known that "ignoring" these attributes does not ensure

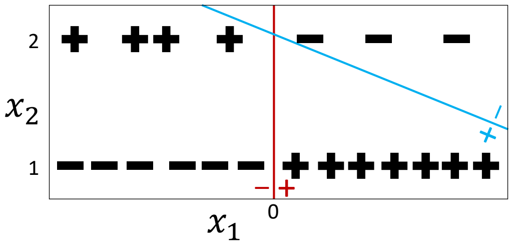

Figure 1: Disregarding group membership (feature x_2), the most accurate linear classifier (red) perfectly classifies the majority class but perfectly *misclassifies* the minority group.

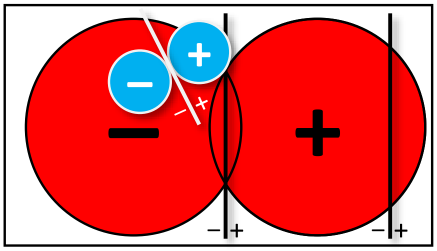

Figure 2: Decoupling helps both majority (red) and minority (blue) groups each maximize accuracy from different linear classifiers (white line and left black line). If, say, equal numbers of positives are required from both groups, the white line and right black line would maximize average accuracy.

fairness, both because they may be closely correlated with other features in the data and because they provide context for understanding the rest of the data, permitting a classifier to incorporate information about cultural differences between groups (Dwork et al., 2011). Using sensitive attributes may increase accuracy for all groups and may avoid biases where a classifier favors members of a minority group that meet criteria optimized for a majority group.

In this paper, we consider *how* to use a sensitive attribute such as gender or race to maximize fairness and accuracy, assuming that it is legal and ethical. If a data scientist wanted to fit, say, a simple linear classifier, they may use the raw data, upweight/oversample data from minority groups, or employ advanced approaches to fitting linear classifiers that aim to be accurate and fair. No matter what they do and what fairness criteria they use, assuming no linear classifier is perfect, they may be faced with an inherent tradeoff between accuracy on one group and accuracy on another. As an extreme illustrative example, consider the two group setting illustrated in Figure 1, where feature x_1 perfectly predicts the binary outcome $y \in \{-1, 1\}$. For people in group 1 (where $x_2 = 1$), the majority group, $y = \text{sgn}(x_1)$, i.e., $y = 1$ when $x_1 > 0$ and -1 otherwise. However, for the minority group where $x_2 = 2$, exactly the opposite holds: $y = -\text{sgn}(x_1)$. Now, if one performed classification without the sensitive attribute x_2, the most accurate classifier predicts $y = \text{sgn}(x_1)$, so the majority group would be perfectly classified and the minority group would be classified as inaccurately as possible. However, even using the group membership attribute x_2, it is impossible to simultaneously achieve better than 50% (random) accuracy on both groups. This is due to limitations of a linear classifier $\text{sgn}(w_1 x_1 + w_2 x_2 + b)$, since the same w_1 is used across groups.

In this paper we define and explore *decoupled* classification systems, in which a separate classifier is trained on each group. Training a classifier involves minimizing a loss function that penalizes errors; examples include mean squared loss and absolute loss. In decoupled classification systems one first obtains, for each group separately, a collection of classifiers differing in the numbers of *positive* classifications returned for the members of the given group. Let this set of outputs for group k be denoted C_k, $k = 1, \ldots, K$. The output of the decoupled training step is an element of $C_1 \times \ldots \times C_K$, that is, a single classifier for each group. The output is chosen to minimize a *joint loss function* that can penalize differences in classification statistics between groups. Thus the loss function can capture *group fairness* properties relating the treatment of different groups, *e.g.*, the false positive (respectively, false negative) rates are the same across groups; the demographics of the group of individuals receiving positive (negative) classification are the same as the demographics of the underlying population; the positive predictive value is the same across groups.[1] By pinning down a specific objective, the modeler is forced to make explicit the tradeoff between accuracy and fairness, since often both cannot simultaneously be achieved.

The following observation provides a property essential for efficient decoupling. A *profile* is a vector specifying, for each group, a number of positively classified examples from the training set. For a given profile $(p_1, \ldots, p_K)$, the most accurate classifier also simultaneously minimizes the false positives and false negatives. *It is the choice of profile that is determined by the joint loss criterion.* We show

[1] In contrast *individual fairness* Dwork et al. (2011) requires that *similar people are treated similarly*, which requires a task-specific, culturally-aware, similarity metric.

that, as long as the joint loss function satisfies a weak form of *monotonicity*, one can use off-the-shelf classifiers to find a decoupled solution that minimizes joint loss. The monotonicity requirement is that the joint loss is non-decreasing in error rates, for any fixed profile. This sheds some light on the thought-provoking impossibility results of Chouldechova (2017) and Joseph et al. (2016) on the impossibility of simultaneously achieving three specific notions of group fairness.

The contributions of this work are: (a) showing how, when using sensitive attributes, the straightforward application of many machine learning algorithms will face inherent tradeoffs between accuracy across different groups, (b) introducing an efficient decoupling procedure that outputs separate classifiers for each class using transfer learning, and (c) modeling fair and accurate learning as a problem of minimizing a joint loss function.

1.1 Related Work

Group fairness has a variety of definitions, including conditions of *statistical parity*, *class balance* and *calibration*. The statistical parity condition requires that the assigned label of an individual is independent of sensitive attributes. Statistical parity can be approximated by either modifying the data set or by designing classifiers subject to fairness regularizers that penalize violations of statistical parity (see Feldman et al. (2015) and references therein). The class-balanced condition (called *error-rate balance* by Chouldechova (2017) or *equalized odds* by Hardt et al. (2016)), similar to statistical parity, requires that the assigned label is independent of sensitive attributes, but only *conditional on the true classification of the individual.* For binary classification tasks, a class-balanced classifier results in equal false positive and false negative rates across groups. One can also modify a given classifier to be class-balanced while minimizing loss by adding label noise (Hardt et al., 2016). The well-calibrated condition requires that, conditional on their label, an equal fraction of individuals from each group have the same true classification. A well-calibrated classifier labels individuals from different groups with equal accuracy. The class-balanced solution (Hardt et al., 2016) also fails to be well-calibrated. Chouldechova (2017) and Joseph et al. (2016) concurrently showed that, except in cases of perfect predictions or equal base rates of true classifications across groups, there is no class-balanced and well-calibrated classifier.

Dwork et al. (2011) propose a “fair affirmative action” methodology that carefully relaxes between-group individual fairness constraints in order to achieve group fairness. Zemel et al. (2013) introduce a representational approach that attempts to “forget” group membership while maintaining enough information to classify similar individuals similarly; this approach also permits generalization to unseen data points. To our knowledge, the earliest work on trying to learn fair classifiers from historically biased data is by Pedreshi et al. (2008); see also (Zliobaite et al., 2011) and (Kamishima et al., 2011).

Additionally, a number of recent works explore causal approaches to defining and detecting bias (Nabi and Shpitser, 2017; Kusner et al., 2017; Bareinboim and Pearl, 2016; Kilbertus et al., 2017).

2 Preliminaries

Let $\mathcal{X} = \mathcal{X}_1 \cup \mathcal{X}_2 \cup \ldots \cup \mathcal{X}_K$ be the set of possible *examples* partitioned by group. The set of possible *labels* is $\mathcal{Y}$ and the set of possible *classifications* is $\mathcal{Z}$. A *classifier* is a function $c : \mathcal{X} \to \mathcal{Z}$. We assume that there is a fixed family $\mathcal{C}$ of classifiers.

We suppose that there is a joint distribution $\mathcal{D}$ over labeled examples $x, y \in \mathcal{X} \times \mathcal{Y}$ and we have access to n training examples $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$ drawn independently from $\mathcal{D}$. We denote by $g(x)$ the group number to which x belongs and $g_i = g(x_i)$, so $x_i \in \mathcal{X}_{g_i}$.

Finally, as is common, we consider the loss $\ell_{\mathcal{D}}(c) = \mathrm{E}_{x,y\sim\mathcal{D}}[\ell(y, c(x))]$ for an application-specific loss function $\ell : \mathcal{Y} \times \mathcal{Z} \to \mathbb{R}$ where $\ell(y, z)$ accounts for the cost of classifying as z an example whose true label is y. The group-k loss for $\mathcal{D}, c$ is defined to be $\ell_{\mathcal{D}k}(c) = \mathrm{E}_{\mathcal{D}}[\ell(y, c(x))|x \in \mathcal{X}_k]$ or 0 if $\mathcal{D}$ assigns 0 probability to $\mathcal{X}_k$. The standard approach in ML is to minimize $\ell_{\mathcal{D}}(c)$ over $c \in \mathcal{C}$. Common loss functions include the L_1 loss $\ell(y, z) = |y - z|$ and L_2 loss $\ell(y, z) = (y - z)^2$. In Section 4, we provide a methodology for incorporating a range of fairness notions into loss.

3 Decoupling and the cost of coupling

For a vector of K classifiers, $\vec{c} = (c_1, c_2, \ldots, c_K)$, the decoupled classifier $\gamma_{\vec{c}} : \mathcal{X} \to \mathcal{Z}$ is defined to be $\gamma_{\vec{c}} = c_{g(x)}(x)$. The set of decoupled classifiers is denoted $\gamma(\mathcal{C}) = \{\gamma_{\vec{c}} \mid \vec{c} \in \mathcal{C}^K\}$. Some classifiers, such as decision trees of unbounded size over $\mathcal{X} = \{0,1\}^d$, are already decoupled, i.e., $\gamma(\mathcal{C}) = \mathcal{C}$. As we shall see, however, in high dimensions common families of classifiers in use are coupled to avoid the curse of dimensionality.

The cost of coupling of a family $\mathcal{C}$ of classifiers (with respect to ℓ) is defined to be the worst-case maximum of the difference between the loss of the most accurate coupled and decoupled classifiers over distributions $\mathcal{D}$.

$$\text{cost-of-coupling}(\mathcal{C}, \ell) = \max_{\mathcal{D} \in \Delta(\mathcal{X} \times \mathcal{Y})} \left[\min_{c \in \mathcal{C}} \ell_{\mathcal{D}}(c) - \min_{\gamma_{\vec{c}} \in \gamma(\mathcal{C})} \ell_{\mathcal{D}}(\gamma_{\vec{c}}) \right].$$

Here $\Delta(S)$ denotes the set of probability distributions over set S. We now show that the cost of coupling related to fairness across groups. All proofs are deferred to the full version.

Lemma 1. *Suppose cost-of-coupling$(\mathcal{C}, \ell) = \text{\textcent}$. Then there is a distribution $\mathcal{D}$ such that no matter which classifier $c \in \mathcal{C}$ is used, there will always be a group k and a classifier $c' \in \mathcal{C}$ whose group-k loss is at least ¢ smaller than that of c, i.e., $\ell_{\mathcal{D}k}(c') \leq \ell_{\mathcal{D}k}(c) - \text{\textcent}$.*

Hence, if the cost of coupling is positive, then the learning algorithm that selects a classifier faces an inherent tradeoff in accuracy across groups. We now show that the cost of coupling is large (a constant) for linear classifiers and decision trees.

Theorem 1. *Fix $\mathcal{X} = \{0,1\}^d$, $\mathcal{Y} = \{0,1\}$, and $K = 2$ groups (encoded by the last bit of x). Then the cost of coupling is at least 1/4 for* **linear regression** *($\mathcal{Z} = \mathbb{R}$, $\mathcal{C} = \{w \cdot x + b \mid w \in \mathbb{R}^d, b \in \mathbb{R}\}$, and $\ell(y,z) = (y-z)^2$),* **linear separators** *($\mathcal{Z} = \{0,1\}$, $\mathcal{C} = \{\mathbb{I}[w \cdot x + b \geq 0] \mid w \in \mathbb{R}^d, b \in \mathbb{R}\}$, and $\ell(y,z) = |y-z|$), and* **bounded-size decision trees** *(for $\mathcal{Z} = \{0,1\}$, $\mathcal{C}$ being the set of binary decision trees of size $\leq 2^s$ leaves, and $\ell(y,z) = |y-z|$).*

4 Joint loss and monotonicity

As discussed, the classifications output by an ML classifier are often evaluated by their empirical loss $\frac{1}{n}\sum_i \ell(y_i, z_i)$. To account for fairness, we generalize loss to joint classifications across groups. In particular, we consider an application-specific *joint loss* $\hat{L} : ([K] \times \mathcal{Y} \times \mathcal{Z})^* \to \mathbb{R}$ that assigns a cost to a set of classifications, where $[K] = \{1, 2, \ldots, K\}$ indicates the group number for each example. A joint loss might be, for parameter $\lambda \in [0,1]$:

$$\hat{L}\big(\langle g_i, y_i, z_i \rangle_{i=1}^n\big) = \frac{\lambda}{n} \sum_{i=1}^n |y_i - z_i| + \frac{1-\lambda}{n} \sum_{k=1}^K \left| \sum_{i:g_i=k} z_i - \frac{1}{K} \sum_i z_i \right|.$$

The above $\hat{L}$ trades off accuracy for differences in number of positive classifications across groups. For $\lambda = 1$, this is simply L_1 loss, while for $\lambda = 0$, the best classifications would have an equal number of positives in each group.

For the remainder of our analysis, we henceforth consider binary labels and classifications, $\mathcal{Y} = \mathcal{Z} = \{0,1\}$. Our approach is general, however. For a given $\langle x_i, y_i, z_i \rangle_{i=1}^n$, and for any group $k \leq K$ and all $(y,z) \in \{0,1\}^2$, recall that the groups are $g_i = g(x_i)$ and define:

$$\textit{counts}\text{:}\ \ n_k = \big|\{i \mid g_i = k\}\big| \in \{1, 2, \ldots, n\}$$

$$\textit{profile}\text{:}\ \ \hat{p}_k = \frac{1}{n} \sum_{i:g_i=k} z_i \in [0, n_k/n]$$

$$\textit{group losses}\text{:}\ \ \hat{\ell}_k = \frac{1}{n_k} \sum_{i:g_i=k} |z_i - y_i| \in [0,1]$$

Note that the normalization is such that the standard 0-1 loss is $\sum_k \frac{n_k}{n}\hat{\ell}_k$ and the fraction of positives within any class is $\frac{n}{n_k}\hat{p}_k$.

In many applications there is a different cost for false positives where $(y, z) = (0, 1)$ and false negatives where $(y, z) = (1, 0)$. The fractions of false positives and negatives for each group k, defined below, can be computed based on the fraction of positive labels $\pi_k \frac{1}{n_k}\sum_{i:g_i=k} y_i$ in each group:

$$\mathrm{FP}_k = \frac{1}{n_k}\sum_{i:g_i=k} z_i(1-y_i) = \frac{\hat{\ell}_k + \hat{p}_k\frac{n}{n_k} - \pi_k}{2} \tag{1}$$

$$\mathrm{FN}_k = \frac{1}{n_k}\sum_{i:g_i=k} (1-z_i)y_i = \frac{\hat{\ell}_k + \pi_k - \hat{p}_k\frac{n}{n_k}}{2}, \tag{2}$$

While minimizing group loss $\hat{\ell}_k = \mathrm{FP}_k + \mathrm{FN}_k$ in general does not minimize false positives or false negatives on their own, the above implies that for a fixed profile $\hat{p}_k$, the most accurate classifier on group k simultaneously minimizes false positives and false negatives. The above can be derived by adding or subtracting the equations $\hat{\ell}_k = \mathrm{FP}_k + \mathrm{FN}_k$ (since every error is a false positive or a false negative) and $\frac{n}{n_k}\hat{p}_k = \mathrm{FP}_k + (\pi_k - \mathrm{FN}_k)$ (since every positive classification is either a false positive or true positive, and the fraction of true positives from group k are $\pi_k - \mathrm{FN}_k$). We also define the *false negative rate* $\mathrm{FNR}_k = \mathrm{FN}_k/\pi_k$. False positive rates can be defined similarly.

Equations (1) and (2) imply that, if one desires fewer false positives and false negatives (all other things being fixed), then greater accuracy is better. That is, for a fixed profile, the most accurate classifier simultaneously minimizes false positives and false negatives. This motivates the following monotonicity notion.

Definition 1 (Monotinicity)**.** *Joint loss $\hat{L}$ is monotonic if, for any fixed $\langle g_i, y_i\rangle_{i=1}^n \in ([K]\times\mathcal{Y})^*$, $\hat{L}$ can be written as $c(\langle\hat{\ell}_k, \hat{p}_k\rangle_{k=1}^K)$ where $c : [0,1]^{2K} \to \mathbb{R}$ is a function that is nondecreasing in each $\hat{\ell}_k$ fixing all other inputs to c.*

That is, for a fixed profile, increasing $\hat{\ell}_k$ can only increase joint loss.

The monotonicity requirement admits a range of different fairness criteria, but not all. We do not mean to imply that monotonicity is necessary for fairness, but rather to discuss the implications of minimizing a non-monotonic loss objective. For example, fix $K = 2$ and $\lambda \le 1/2$. Then the following joint loss is monotonic: $(1-\lambda)(\hat{\ell}_1 + \hat{\ell}_2) + \lambda|\hat{\ell}_1 - \hat{\ell}_2|$. This loss trades off accuracy for differences in loss rates between groups. What we see is that monotonic losses can account, to a limited extent, for differences across groups in fractions of errors, and related statements can be made for combinations of rates of false positive and false negative, inspired by “equal odds” definitions of fairness. However, when the weight λ on the fairness term exceeds $1/2$, then the loss is non-monotonic and one encounters situations where one group is punished with lower accuracy in the name of fairness. This may still be desirable in a context where equal odds is a primary requirement, and one would rather have random classifications (e.g., a lottery) than introduce any inequity.

5 Minimizing joint loss on training data

Here, we show how to use learning algorithm to find a decoupled classifier in $\gamma(\mathcal{C})$ that is optimal on the training data. Our approach to decoupling uses a learning algorithm for $\mathcal{C}$ as a black box. A *$\mathcal{C}$-learning algorithm* $A : (\mathcal{X}\times\mathcal{Y})^* \to 2^{\mathcal{C}}$ returns one or more classifiers from $\mathcal{C}$ with differing numbers of positive classifications on the training data, i.e., for any two distinct $c, c' \in A(\langle x_i, y_i\rangle_{i=1}^n)$, $\sum_i c(x_i) \ne \sum_i c'(x_i)$. In ML, it is common to simultaneously output classifiers with varying number of positive classifications, e.g., in computing ROC or precision-recall curves (Davis and Goadrich, 2006). Also note that a classifier that purely minimizes errors can be massaged into one that outputs different fractions of positive and negative examples by reweighting (or subsampling) the positive- and negative-labeled examples with different weights.

Our analysis will be based on the assumption that the classifier is in some sense optimal, but importantly note that it makes sense to apply the reduction to any off-the-shelf learner. Formally,

Algorithm 1 The simple decoupling algorithm partitions data by group and runs the learner on each group. Within each group, the learner outputs one or more classifiers of differing numbers of positives.

```
procedure DECOUPLE($A, \hat{L}, \langle x_i, y_i \rangle_{i=1}^n, \mathcal{X}_1, \ldots, \mathcal{X}_K$)    ▷ Minimize training loss $\hat{L}$ using learner $A$
    for $k = 1$ to $K$ do
        $C_k \leftarrow A\big(\langle x_i, y_i \rangle_{i: x_i \in \mathcal{X}_k}\big)$    ▷ Learner outputs a set of classifiers
    return $\gamma_{\vec{c}}$ that minimizes $\min_{\vec{c} \in C_1 \times \ldots \times C_K} \hat{L}\big(\langle g_i, y_i, \gamma_{\vec{c}}(x_i) \rangle_{i=1}^n\big)$    ▷ $\gamma_{\vec{c}}(x_i) = c_{g_i}(x))$
```

we say A is *optimal* if for every achievable number of positives $P \in \left\{ \sum_i c(x_i) \;\middle|\; c \in \mathcal{C} \right\}$, it outputs exactly one classifier that classifies exactly P positives, and this classifier has minimal error among all classifiers which classify exactly P positives. Theorem 2 shows that an optimal classifier can be used to minimize any (monotonic) joint loss

Theorem 2. *For any monotonic joint loss function $\hat{L}$, any $\mathcal{C}$, and any optimal learner A for $\mathcal{C}$, the* DECOUPLE *procedure from Algorithm 1 returns a classifier in $\gamma(\mathcal{C})$ of minimal joint loss $\hat{L}$. For constant K,* DECOUPLE *runs in time linear in the time to run A and polynomial in the number of examples n and time to evaluate $\hat{L}$ and classifiers $c \in \mathcal{C}$.*

References

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks. *ProPublica, May*, 23, 2016.

Elias Bareinboim and Judea Pearl. Causal inference and the data-fusion problem. *Proceedings of the National Academy of Sciences*, 113(27):7345–7352, 2016.

Richard Berk. The role of race in forecasts of violent crime. *Race and social problems*, 1(4):231, 2009.

Alexandra Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *arXiv*, 2017.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. *ITCS*, 2011.

Michael Feldman, Sorelle A. Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.

Sorelle A Friedler, Carlos Scheidegger, and Suresh Venkatasubramanian. On the (im)possibility of fairness. *arXiv preprint arXiv:1609.07236*, 2016.

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. *NIPS*, 2016.

Matthew Joseph, Michael Kearns, Jamie H Morgenstern, and Aaron Roth. Fairness in learning: Classic and contextual bandits. In *Advances in Neural Information Processing Systems*, pages 325–333, 2016.

Toshihiro Kamishima, Shotaro Akaho, and Jun Sakuma. Fairness-aware learning through regularization approach. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops*, ICDMW '11, pages 643–650, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-0-7695-4409-0. doi: 10.1109/ICDMW.2011.83. URL http://dx.doi.org/10.1109/ICDMW.2011.83.

Niki Kilbertus, Mateo Rojas-Carulla, Giambattista Parascandolo, Moritz Hardt, Dominik Janzing, and Bernhard Schölkopf. Avoiding discrimination through causal reasoning. *arXiv preprint arXiv:1706.02744*, 2017.

M. J. Kusner, J. R. Loftus, C. Russell, and R. Silva. Counterfactual Fairness. *ArXiv e-prints*, March 2017.

Matt J Kusner, Joshua R Loftus, Chris Russell, and Ricardo Silva. Counterfactual fairness. *arXiv preprint arXiv:1703.06856*, 2017.

Razieh Nabi and Ilya Shpitser. Fair inference on outcomes. *arXiv preprint arXiv:1705.10378*, 2017.

Dino Pedreshi, Salvatore Ruggieri, and Franco Turini. Discrimination-aware data mining. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 560–568, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-193-4. doi: 10.1145/1401890.1401959. URL http://doi.acm.org/10.1145/1401890.1401959.

R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork. Learning fair representations. *Proc. of Intl. Conf. on Machine Learning*, 2013.

Indre Zliobaite, Faisal Kamiran, and Toon Calders. Handling conditional discrimination. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, ICDM '11, pages 992–1001, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-0-7695-4408-3. doi: 10.1109/ICDM.2011.72. URL http://dx.doi.org/10.1109/ICDM.2011.72.