Communications for Statistical Applications and Methods
2022, Vol. 29, No. 2, 251–261

https://doi.org/10.29220/CSAM.2022.29.2.251
Print ISSN 2287-7843 / Online ISSN 2383-4757

Learning fair prediction models with an imputed sensitive variable: Empirical studies

Yongdai Kim[1,a], Hwichang Jeong[a]

[a]Department of Statistics, Seoul National University, Korea

Abstract

As AI has a wide range of influence on human social life, issues of transparency and ethics of AI are emerging. In particular, it is widely known that due to the existence of historical bias in data against ethics or regulatory frameworks for fairness, trained AI models based on such biased data could also impose bias or unfairness against a certain sensitive group (e.g., non-white, women). Demographic disparities due to AI, which refer to socially unacceptable bias that an AI model favors certain groups (e.g., white, men) over other groups (e.g., black, women), have been observed frequently in many applications of AI and many studies have been done recently to develop AI algorithms which remove or alleviate such demographic disparities in trained AI models.

In this paper, we consider a problem of using the information in the sensitive variable for fair prediction when using the sensitive variable as a part of input variables is prohibitive by laws or regulations to avoid unfairness. As a way of reflecting the information in the sensitive variable to prediction, we consider a two-stage procedure. First, the sensitive variable is fully included in the learning phase to have a prediction model depending on the sensitive variable, and then an imputed sensitive variable is used in the prediction phase. The aim of this paper is to evaluate this procedure by analyzing several benchmark datasets. We illustrate that using an imputed sensitive variable is helpful to improve prediction accuracies without hampering the degree of fairness much.

Keywords: AI, bias, fair prediction, imputed sensitive variable

1. Introduction

Recently, artificial intelligence (AI) is being used as decision-making tools in various domains such as credit scoring, criminal risk assessment, education of college admissions (Angwin *et al.*, 2016). As AI has a wide range of influences on human social life, issues of transparency and ethics of AI are emerging. However, it is widely known that due to the existence of historical bias in data against ethics or regulatory frameworks for fairness, trained AI models based on such biased data could also impose bias or unfairness against a certain sensitive group (e.g., non-white, women) (Kleinberg *et al.*, 2018; Mehrabi *et al.*, 2021). Therefore, designing an AI algorithm which is accurate and fair simultaneously has become a crucial research topic.

Demographic disparities due to AI, which refer to socially unacceptable bias that an AI model favors certain groups (e.g., white, men) over other groups (e.g., black, women), have been observed frequently in many applications of AI such as COMPAS recidivism risk assessment (Angwin *et al.*, 2016), Amazon's prime free same-day delivery (Ingold and Soper, 2016), credit score evaluation (Dua and Graff, 2019) to name just a few. Many studies have been done recently to develop AI algorithms which remove or alleviate such demographic disparities in trained AI models so that they will treat

[1] Corresponding author: Department of Statistics, Seoul National University, 1 Gwanak-ro, Gwanak-gu, Seoul, Korea.
E-mail: ydkim0903@gmail.com

Published 31 March 2022 / journal homepage: http://csam.or.kr

sensitive groups as equally as possible. In general, these methods try to search AI models which are not only accurate but also similar between sensitive groups in a certain sense. For an example of similarity, it is required that accuracies of an AI model for each sensitive group are similar (Zafar *et al.*, 2019).

In this paper, we consider a problem of using the sensitive variable for fair prediction. In most real applications, the sensitive variable itself has important information for prediction and using the sensitive variable as a part of input variables is usually helpful to improve prediction accuracies. Moreover, some fairness AI algorithms inevitably produce prediction models which depend on the sensitive variable as well as input variables. An example is the algorithm of Jiang *et al.* (2020) for the strong demographic parity, which transfers the score function of each sensitive group such that the distributions of the scores of each sensitive group are all equal. For this algorithm, the transformation of the score function should be differ for each sensitive group and hence the sensitive variable should be known in the prediction phase. In many cases, however, using the sensitive variable as a part of input variables is prohibitive by laws or regulations to avoid unfairness. In such cases, fairness AI algorithms yielding prediction models depending on the sensitive variable cannot be used.

A simple solution to reflect the information of the sensitive variable into prediction when using the sensitive variable explicitly in prediction is prohibitive to use an imputed sensitive variable in the prediction phase. That is, the sensitive variable is fully included in the learning phase to have a prediction model depending on the sensitive variable and then an imputed sensitive variable is used in the prediction phase. The aim of this paper is to evaluate this procedure by analyzing several benchmark datasets. We illustrate that using an imputed sensitive variable is helpful to improve prediction accuracies without hampering the degree of fairness much. That is, prediction models with an imputed sensitive variable are superior compared to prediction models not using the sensitive variable at all.

The paper is organized as follows. Various fairness algorithms are reviewed in Section 2. The proposed procedure to include the information of the sensitive variables by using an imputed sensitive variable into the prediction phase is explained in Section 3. Results of numerical studies are presented in Section 4 and concluding remarks follow in Section 5.

2. Review of fair AI algorithms

We let $\mathcal{D} = \{(\mathbf{x}_i, z_i, y_i)\}_{i=1}^n$ be a set of training data of size n consisting of triplets of input vector $\mathbf{x}_i$, sensitive variable z_i and class label y_i, which are assumed to be independent copies of a random vector $(\mathbf{X}, Z, Y)$ defined on $\mathcal{X} \times \mathcal{Z} \times \mathcal{Y}$, where $\mathcal{X} \subset \mathbb{R}^p$. We consider a binary classification problem, which means $\mathcal{Y} = \{-1, 1\}$, and for notational simplicity, we let $\mathcal{Z} = \{0, 1\}$, where $Z = 0$ refers to the unprivileged group and $Z = 1$ refers to the privileged group. Whenever the probability is mentioned, we mean it by either the probability of $(\mathbf{X}, Z, Y)$ or its empirical counterpart unless there is any confusion.

In this paper, we focus on between-group fairness (BGF) which requires that certain statistics of predictive values in each sensitive group should be similar. Even if we do not consider other concepts of fairness such as individual fairness (Dwork *et al.*, 2012) and counter-factural fairness (Kusner *et al.*, 2017) , our proposed method can be applied to such fairness concepts after minor modifications.

We consider AI algorithms which yield a real-valued function $f : \mathcal{X} \to \mathbb{R}$ so called a score function which assigns positive labeled instances higher scores than negative labeled instances. An example of the score function is the conditional class probability $\Pr(Y = 1|\mathbf{x} = \mathbf{x})$. In most human-related decision makings, real-valued score functions are popularly used (e.g. scores for credit scoring).

Let $\mathcal{F}$ be a given set of score functions, in which we search an optimal score function in a certain

Table 1: Some group performance functions

Fairness criteria	$\mathcal{E}$	$\mathcal{E}'$
Disparate impact (Barocas and Selbst, 2016)	$\mathbb{1}\{C_f(X) = 1\}$	$\emptyset$
Equal opportunity (Hardt *et al.*, 2016)	$\mathbb{1}\{C_f(X) = 1\}$	$\{Y = 1\}$
Disparate mistreatment w.r.t. Error rate (Zafar *et al.*, 2019)	$\mathbb{1}\{C_f(X) \neq Y\}$	$\emptyset$
Mean score parity (Coston *et al.*, 2019)	$f(X)$	$\emptyset$

sense (e.g. minimizing the cross-entropy for classification problems). Examples of $\mathcal{F}$ are linear functions, reproducing kernel Hilbert space and deep neural networks to name a few. For a given $f \in \mathcal{F}$, the corresponding classifier C_f is defined as $C_f(\mathbf{x}) = \mathbb{1}(f(\mathbf{x}) > 0)$.

2.1. Definition of between-group fairness

For a given score function f and a sensitive group $Z = z$, we consider the group performance function of f given as

$$q_z(f) := \mathbb{E}(\mathcal{E}|\mathcal{E}', Z = z), \tag{2.1}$$

for events $\mathcal{E}$ and $\mathcal{E}'$ that might depend on $f(\mathbf{X})$ and Y.

The group performance function q_z in (2.1), which is considered by Celis *et al.* (2019), includes various performance functions used in fairness AI. We summarize representative group performance functions having the form of (2.1) in Table 1.

For given group performance functions $q_z(\cdot)$, $z \in \{0, 1\}$, we say that f satisfies the BGF constraint with respect to q_z if $q_0(f) = q_1(f)$. A relaxed version of the BGF constraint so called the ϵ-BGF constraint, is frequently considered, which requires $|q_0(f) - q_1(f)| < \epsilon$ for a given $\epsilon > 0$. Typically, AI algorithms search an optimal function f among those satisfying the ϵ-BGF constraint with respect to given group performance functions $q_z(\cdot), z \in \{0, 1\}$.

2.2. Learning algorithms for fair artificial intelligence (AI)

Several learning algorithms have been proposed to find an accurate model f satisfying a given BGF constraint, which are categorized into the following three groups. In this subsection, we review some methods for each group.

Pre-processing methods: Pre-processing methods remove bias in training data or find a fair representation with respect to a sensitive variable before the learning phase and learn AI models based on de-biased data or fair representation (Calmon *et al.*, 2017; Creager *et al.*, 2019; Dixon *et al.*, 2018; Feldman *et al.*, 2015; Kamiran and Calders, 2012; Quadrianto *et al.*, 2019; Webster *et al.*, 2018; Xu *et al.*, 2018; Zemel *et al.*, 2013). Kamiran and Calders (2012) suggested pre-processing methods to eliminate bias in training data by use of label changing, reweighing and sampling. Based on the idea that transformed data should not be able to predict the sensitive variable, Feldman *et al.* (2015) proposed a transformation of input variables for eliminating the disparate impact. To find a fair representation, Calmon *et al.* (2017) and Zemel *et al.*(2013) proposed a data transformation mapping for preserving accuracy and alleviating discrimination simultaneously. Pre-processing methods for fair learning on text data were studied by Dixon *et al.* (2018) and Webster *et al.* (2018).

In-processing methods: In-processing methods generally train an AI model by minimizing a given cost function (e.g. the cross-entropy, the sum of squared residuals, the empirical AUC etc.) subject to a ϵ-BGF constraint. Most group performance functions $q_z(\cdot)$ are not differentiable, and thus various surrogated group performance functions and corresponding ϵ-BGF constraints have been

proposed (Bechavod and Ligett, 2017; Celis *et al.*, 2019; Cho *et al.*, 2020; Donini *et al.*, 2018; Goh *et al.*, 2016; Kamishima *et al.*, 2012; Menon and Williamson, 2018; Narasimhan, 2018; Vogel *et al.*, 2020; Zafar *et al.*, 2017, 2019). Kamishima *et al.* (2012) used a fairness regularizer which is an approximation of the mutual information between the sensitive variable and the target variable. Zafar *et al.* (2017, 2019) proposed covariance-type fairness constraints as tractable proxies targeting the disparate impact and the equality of the false positive or negative rate, and Donini *et al.* (2018) used a linear surrogated group performance function for the equalized odds. On the other hand, Menon and Williamson (2018) and Celis *et al.* (2019) derived an optimal classifier for a constrained fair classification as a form of an instance-dependent threshold. Also, for fair score functions, Vogel *et al.* (2020) proposed fairness constraints based on ROC curves of each sensitive group.

Post-processing methods: Post-processing methods first learn an AI model without any BGF constraint and then transform the decision boundary or score function of the trained AI model for each sensitive group to satisfy given BGF criteria (Chzhen *et al.*, 2019; Corbett-Davies *et al.*, 2017; Fish *et al.*, 2016; Hardt *et al.*, 2016; Jiang *et al.*, 2020; Kamiran, *et al.*, 2012; Pleiss *et al.*, 2017; Wei *et al.*, 2021). Chzhen *et al.* (2019) and Hardt *et al.* (2016) suggested finding sensitive group dependent thresholds to get a fair classifier with respect to equal opportunity. Jiang *et al.* (2020) and Wei *et al.* (2021) developed an algorithm to transform the original score function to achieve a BGF constraint.

3. Fair prediction models with an imputed sensitive variable

We consider two situations according to whether the sensitive variable can be used in the prediction. The first situation (Situation 1) is that the sensitive variable z is allowed to be used in the prediction phase and thus we assume that z is one of the entries of the input vector. That is, there exists $j \in [p]$ such that $x_j = z$. In contrast, the second situation (Situation 0) is that the sensitive variable cannot be used in the prediction phase and thus the sensitive variable does not belong to the entries of the input vector. As we mentioned in Introduction, there are many cases where the sensitive variable is not allowed to be a part of the input vector by regulations or laws to ensure fairness although using z in prediction would help improving prediction accuracy.

In this paper, we propose a method to use the information in the sensitive variable under Situation 0. The idea of the proposed method is simple and intuitive. At the learning phase, a prediction model is learned with the input vector including the sensitive variable. Then, at the prediction phase, we impute the sensitive variable based on the other input variables and make a prediction with the imputed sensitive variable.

To be more specific, let $\mathbf{x}$ be the input vector not including z and let $\mathbf{x}_z = (\mathbf{x}^\top, z)^\top$ is the input vector including z. In the learning phase, we learn a prediction model $f : \mathcal{X} \times \mathcal{Z} \to \mathbb{R}$ with the training data $(\mathbf{x}_{z,1}, y_1), \ldots, (\mathbf{x}_{z,n}, y_n)$. In addition, we learn a prediction model $g : \mathcal{X} \to \mathcal{Z}$ which predicts z based on $\mathbf{x}$. Then, in the prediction phase, for a given input vector $\mathbf{x}$, we make a prediction by $f(\mathbf{x}_{\hat{z}})$, where $\hat{z} = g(\mathbf{x})$. The proposed procedure is summarized in Algorithm 1.

There are at least two advantages of the proposed method compared to the standard method for Situation 0 that learns a prediction model $f^0 : \mathcal{X} \to \mathbb{R}$ in the learning phase. First of all, using the information of z in the prediction phase would be usually helpful for improving prediction accuracy. This advantage would keep being materialized even when an imputed z is used, which will be confirmed by numerical studies on Section 4.

The second advantage, which is the main motivation of our proposed method, lies in that there are several useful fair AI algorithms which yield prediction models which are functions on $\mathcal{X} \times \mathcal{Z}$. An example is the Wasserstein fair algorithm (Jiang *et al.*, 2020) whose details are given in Section 4.1. It

would not be easy to modify this algorithm for not using z in the prediction phase. In contrast, we can use such algorithms without much modification by use of an imputed sensitive variable. By analyzing several real datasets, we show that using an imputed sensitive variable instead of the sensitive variable itself does not degrade the performance of prediction models much in view of prediction accuracy as well as fairness.

Algorithm 1 Learning fair prediction models with an imputed sensitive variable

[1] Learning phase: Learn a prediction model $f(\mathbf{x}_z)$ based on the training data $\mathcal{D}$
[2] Imputation phase: Learn a prediction model $g : \mathcal{X} \to \{0, 1\}$ which predicts the sensitive variable z by $\mathbf{x}$.
[3] Prediction phase: For a given new input $\mathbf{x}$, we predict y based on $f(\mathbf{x}_{\hat{z}})$, where $\hat{z} = g(\mathbf{x})$.

4. Numerical studies

In this section, we investigate the performance of the proposed procedure and compare it with the predictions models without using an imputed sensitive variable (either not using the sensitive variable under S0 or using the sensitive variable fully in the prediction phase under S1).

4.1. Considered fair AI algorithms

For fairness AI algorithms, we consider the three algorithms : (1) fair classifier with the disparity impact (DI) constraint (Zafar *et al.*, 2017), (2) fair classifier with the prejudice index (PI) constraint (Kamishima *et al.*, 2012) and (3) Wasserstein fair classifier of (Jiang *et al.*, 2020). The first two algorithms are in-processing methods and the third one is a post-processing method.

Fair classifier with the DI constraint: The DI constraint requires that a prediction model f satisfies

$$|\mathbb{E}_n(\mathbb{1}\{f(\mathbf{X}) > 0\}|Z = 0) - \mathbb{E}_n(\mathbb{1}\{f(\mathbf{X}) > 0\}|Z = 1)| \leq \epsilon, \tag{4.1}$$

where $\mathbb{E}_n$ is the expectation with respect to the empirical distribution. By use of the Lagrangian multiplier, we could learn a prediction model by minimizing the penalized empirical risk given as

$$C_n(f) + \lambda|\mathbb{E}_n(\mathbb{1}\{f(\mathbf{X}) > 0\}|Z = 0) - \mathbb{E}_n(\mathbb{1}\{f(X) > 0\}|Z = 1)|$$

over $\mathcal{F}$, where $C_n(\cdot)$ is a given empirical risk. In this paper, we consider the cross-entropy for the empirical risk. The DI constraint (4.1) is hard to be optimized since the indicator function $\mathbb{1}(\cdot)$ is neither continuous nor convex. A typical remedy is to replace the indicator function by a convex surrogate function. One of the popularly used convex surrogate functions is the hinge function defined as $\phi_{\text{hinge}}(w) = (1 - w)_+$ (Donini *et al.*, 2018). Using the hinge surrogate loss, we learn the prediction model by minimizing the surrogate penalized empirical risk

$$C_n(f) + \lambda\Big|\mathbb{E}_n(\phi_{\text{hinge}}(f(\mathbf{X}))|Z = 0) - \mathbb{E}_n(\phi_{\text{hinge}}(f(\mathbf{X}))|Z = 1)\Big|,$$

over $\mathcal{F}$.

Fair classifier with the PI constraint: PI measures statistical dependence between the class label and sensitive variable. We think that a given classifier is more fair if its PI is smaller. For data

$(x_i, z_i)_{i=1}^n = \mathcal{D}$, we define PI as the prejudice remover regularizer given as

$$PI(f) = \sum_{(x_i,z_i)\in\mathcal{D}} \sum_{y\in\{0,1\}} \mathcal{M}(y|x_i, z_i) \ln \frac{\widehat{\Pr}(y|z_i)}{\widehat{\Pr}(y)},$$

where

$$\begin{aligned}
\mathcal{M}(y|x_i, z_i) &= y\sigma(f(x_i)) + (1-y)(1-\sigma(f(x_i))), \\
\widehat{\Pr}(y|z) &= \frac{\sum_{(x_i,z_i)\in\mathcal{D} \text{ s.t. } z_i=z} \mathcal{M}(y|x_i, z)}{|\{(x_i, z_i) \in \mathcal{D} \text{ s.t. } z_i = z\}|}, \\
\widehat{\Pr}(y) &= \frac{\sum_{(x_i,z_i)\in\mathcal{D}} \mathcal{M}(y|x_i, z)}{|\mathcal{D}|}.
\end{aligned}$$

σ is sigmoid function. We learn a fair classifier by minimizing $C_n(f) + \lambda_n PI(f)$ over $\mathcal{F}$.

Wasserstein fair classifier: This method is a post-processing method. Let $S_z(\mathbf{x})$ be the estimated conditional class probability $\Pr(Y = 1|\mathbf{X} = \mathbf{x}, Z = z)$. Let $P_{n,z}$ be the empirical distribution of $\{S_z(\mathbf{x}_i) : z_i = z\}$. Then, the barycenter distribution $\bar{P}$ is defined as

$$\bar{P} = \arg\min_{P\in\mathcal{P}(\Omega)} \sum_{z\in\{0,1\}} \frac{n_z}{n} \mathcal{W}_1(P_{n,z}, P),$$

where $\mathcal{W}_1$ is the L_1 Wasserstein distance and $\mathcal{P}(\Omega)$ is set of distributions on $\Omega = [0, 1]$. Once the barycenter $\bar{P}$ is obtained, the fair classifier is constructed by use of the quantile matching between $P_{n,z}$ and $\bar{P}$. For detailed procedures, we refer (Jiang *et al.*, 2020).

4.2. Analyzed data sets

To evaluate the proposed procedure, we analyze three benchmark real world datasets : Adult income dataset, Bank marketing dataset and Law school dataset.

- Adult dataset is composed of 45,222 individuals' features (e.g. age, education, race) with a binary label which indicates whether individual's income is larger (positive class) than 50K USD. We take the variable 'sex' as a sensitive variable.
- Bank dataset is composed of 41,188 clients' features (e.g. job, education) with a binary label which indicates whether has a client subscribed a term deposit. We take the variable 'age' as a sensitive variable which is 1 when client's age is between 25 and 60 years.
- Law dataset is composed of 26,551 law school applicants' features (e.g. lsat score, family income) with a binary label which indicates whether applicant is accepted to law school. We take the variable 'race' as a sensitive variable which is 1 when applicant is white.

4.3. Performance of the proposed method

We use three classification algorithms for g : logistic regression, boosting, deep neural network. Also we use logistic regression and deep neural network for classifier f. The structure of DNN for g is two hidden layers with the number of nodes for each hidden layers is equal to the number of input nodes. The structure of DNN for f is two hidden layers with the numbers of nodes for each hidden layers

Table 2: No fairness constraint : This table shows the performance of classfiers with no fairness constraint. We fit classifiers f through logistic regression(f: Logistic) and DNN(f: DNN). We calculate test data accuracy, DI and PI in various situations when sensitive variables are not used(S0), sensitive variables are used(S1), and imputed sensitive variables through Boosting, Logistic and DNN are used

Data	Value	f : Logistic					f : DNN				
		S0	S1	Our method			S0	S1	Our method		
				Boost	Logistic	DNN			Boost	Logistic	DNN
Adults	ACC	0.852	0.852	0.851	0.852	0.852	0.851	0.852	0.851	0.852	0.851
	DI	0.172	0.177	0.186	0.187	0.194	0.172	0.178	0.187	0.188	0.195
	PI	0.021	0.023	0.026	0.026	0.026	0.021	0.024	0.041	0.026	0.029
Bank	ACC	0.911	0.911	0.911	0.911	0.910	0.911	0.911	0.911	0.911	0.911
	DI	0.174	0.229	0.343	0.395	0.336	0.173	0.210	0.316	0.366	0.307
	PI	0.007	0.009	0.011	0.013	0.011	0.007	0.008	0.010	0.012	0.011
Law	ACC	0.823	0.823	0.823	0.822	0.823	0.823	0.823	0.823	0.823	0.823
	DI	0.119	0.148	0.277	0.277	0.260	0.119	0.145	0.272	0.273	0.251
	PI	0.009	0.012	0.022	0.018	0.019	0.009	0.012	0.022	0.018	0.019

are 100 and 50. When optimization we use SGD algorithm : weight decay $5e^4$, learning rate is 0.005 and the number of epoch is 30,000 and reduce learning rate 0.1 times at epoch [10000, 20000, 25000] for g. For f, weight decay is $5e^4$, learning rate is 0.1 and the number of epoch is 50000 and reduce learning rate 0.1 times at epoch [30000, 40000, 45000].

Whenever the regularization parameter is selected (e.g. Lagrangian multipliers for DI and PI), we search it so that the corresponding estimated classifier achieves a certain level of fairness which is set in advance and compare the accuracy of the classifier on test data. We select the one with the highest train accuracy among classifiers corresponded to regularization parameters that make the DI of the train data less than 0.05(the case of PI is 0.005). We repeat 5 times to split train data and test data with ratio 7:3 and average the performances.

For the Wasserstein fair classifier which is a post-processing method, we evaluate the strong demographic parity (SDP) measure as well as the DI.The SDP for a given belief function $S(\mathbf{x})$ that is an estimate of $\Pr(Y = 1|\mathbf{X} = \mathbf{x})$ is defined as

$$\sum_{z=0}^{1} E_{\tau\sim U[0,1]}|P_n(S(\mathbf{X}) > \tau|Z = z) - P_n(S(\mathbf{X}) > \tau)|,$$

where P_n is the empirical distribution.

Results of our numerical studies are presented in Tables 2 to 5. The remarks are summarized as follows.

- Table 2 summarizes the results without fairness constraints. It is observed that using the sensitive variable in prediction is not helpful. However, as we see in Tables 3 and 4, using the sensitive variable in prediction is helpful for fair learning algorithms. It is interesting to figure out why the role of the sensitive variable in prediction is different for standard and fair learning algorithms, which we leave as a future work.

- From Tables 3 and 4, we can see that the fair classifiers with an imputed sensitive variable are superior to the fair classifiers without using the sensitive variable (S_0). Moreover, the fair classifiers with an imputed sensitive variable is not much worse than the fair classifier using the sensitive variable fully (S_1).

Table 3: In-processing DI : This table shows the performance of classfiers with DI fairness constraint. We fit classifiers f through logistic regression(f: Logistic) and DNN(f: DNN). We calculate test data accuracy and DI in various situations when sensitive variables are not used(S0), sensitive variables are used(S1), and imputed sensitive variables through Boosting, Logistic and DNN are used

Data	Value	f : Logistic					f : DNN				
		S0	S1	Our method			S0	S1	Our method		
				Boost	Logistic	DNN			Boost	Logistic	DNN
Adults	ACC	0.832	0.832	0.832	0.832	0.832	0.832	0.832	0.832	0.832	0.832
	DI	0.027	0.028	0.037	0.039	0.045	0.028	0.029	0.037	0.040	0.045
Bank	ACC	0.904	0.908	0.908	0.908	0.908	0.905	0.909	0.909	0.909	0.909
	DI	0.031	0.007	0.019	0.032	0.017	0.031	0.005	0.025	0.047	0.021
Law	ACC	0.811	0.820	0.820	0.821	0.821	0.811	0.820	0.819	0.820	0.820
	DI	0.017	0.030	0.082	0.093	0.085	0.016	0.030	0.081	0.091	0.083

Table 4: In-processing PI : This table shows the performance of classfiers with PI fairness constraint. We fit classifiers f through logistic regression(f: Logistic) and DNN(f: DNN). We calculate test data accuracy and PI in various situations when sensitive variables are not used(S0), sensitive variables are used(S1), and imputed sensitive variables through Boosting, Logistic and DNN are used

Data	Value	f : Logistic					f : DNN				
		S0	S1	Our method			S0	S1	Our method		
				Boost	Logistic	DNN			Boost	Logistic	DNN
Adults	ACC	0.829	0.832	0.833	0.832	0.833	0.829	0.832	0.833	0.833	0.834
	PI	0.003	0.000	0.001	0.001	0.002	0.003	0.000	0.001	0.001	0.002
Bank	ACC	0.907	0.909	0.909	0.909	0.909	0.908	0.911	0.911	0.911	0.911
	PI	0.002	0.001	0.001	0.002	0.007	0.002	0.004	0.006	0.007	0.006
Law	ACC	0.812	0.818	0.817	0.818	0.818	0.812	0.822	0.817	0.819	0.818
	PI	0.002	0.000	0.004	0.004	0.004	0.002	0.005	0.004	0.004	0.004

Table 5: Wasserstein post-processing : This table shows the performance of classfiers with Wasserstein post-processing. We fit classifiers f through logistic regression(f: Logistic) and DNN(f: DNN). We calculate test data accuracy, DI and SDP in various situations when sensitive variables are used(S1), and imputed sensitive variables are used (Boosting, Logistic and DNN)

Data	Value	f: Logistic				f: DNN			
		S1	Boosting	Logistic	DNN	S1	Boosting	Logistic	DNN
Adults	ACC	0.721	0.832	0.831	0.833	0.720	0.838	0.837	0.839
	DI	0.008	0.010	0.003	0.013	0.002	0.013	0.005	0.017
	SDP	0.003	0.008	0.006	0.013	0.003	0.008	0.007	0.015
Bank	ACC	0.875	0.909	0.909	0.909	0.872	0.911	0.911	0.911
	DI	0.044	0.024	0.048	0.030	0.041	0.024	0.042	0.030
	SDP	0.056	0.039	0.064	0.044	0.052	0.037	0.050	0.043
Law	ACC	0.789	0.815	0.817	0.816	0.790	0.825	0.828	0.827
	DI	0.034	0.042	0.052	0.044	0.046	0.043	0.052	0.046
	SDP	0.029	0.060	0.071	0.066	0.036	0.061	0.070	0.066

- For the Wasserstein fair classifier whose results are presented in Table 5, it is somehow surprising that the fair classifier with an imputed sensitive variable is superior to that using the sensitive variable itself. A possible answer would be that an imputed sensitive variable could regularize the estimated classifier and so that could avoid overfitting.

- In general, the performances of fair prediction models with an surrogated sensitive variable do not strongly depend on the choice of an imputational algorithm. Table 6 summarizes the accuracies of

Table 6: Accuracy of g : This table shows the accuracy of various classfiers with-boosting, logistic regression and DNN

Data	Method	Acc	
		train	test
Adults	Boosting	0.879	0.848
	Logistic	0.844	0.842
	DNN	0.872	0.837
Bank	Boosting	0.982	0.967
	Logistic	0.964	0.964
	DNN	0.972	0.966
Law	Boosting	0.928	0.881
	Logistic	0.877	0.878
	DNN	0.883	0.882

the three imputation algorithms. Boosting seems to be the best but the performances of the corresponding fair prediction models are similar to the other imputation algorithms. These observations suggest that the accuracy of an imputed sensitive variable is not important for the performance of the corresponding fair prediction model unless the accuracy is too bad.

5. Concluding remarks

In this paper, we have illustrated that using an imputed sensitive variable is helpful when using the sensitive variable itself in the prediction phase is not allowed. Also, the accuracy of imputing the sensitive variable does not affect the overall performance of fair classifiers. Any reasonable supervised learning algorithms would be enough to obtain an imputed sensitive variables.

In this paper, we proposed a two-step procedure where we learn the fair classifier and the prediction model for surrogated sensitive variables separately. A better procedure would do this two jobs at the same time. That is, we are to learn the fair classifier and imputed sensitive variable simultaneously. This would be a promising direction for future works.

References

Angwin J, Larson J, Mattu S, and Kirchnerb L (2016). Machine bias, *ProPublica*, **23**, 139–159.

Barocas S and Selbst AD (2016). Big data's disparate impact, *California Law Review*, **104**, 671–732.

Bechavod Y and Ligett K (2017). *Learning Fair Classifiers: A Regularization-Inspired Approach*, arXiv preprint arXiv:1707.00044.

Calmon F, Wei D, Vinzamuri B, Ramamurthy KN, and Varshney KR (2017). Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems*, **30**, 3992—4001.

Celis LE, Huang L, Keswani V, and Vishnoi NK (2019). Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 319-–328, ACM.

Cho J, Hwang G, and Suh C (2020). A fair classifier using kernel density estimation. In *34th Conference on Neural Information Processing Systems*, **33**, 15088–15099.

Chzhen E, Denis C, Hebiri M, Oneto L, and Pontil M (2019). Leveraging labeled and unlabeled data for consistent fair binary classification. In *Advances in Neural Information Processing Systems*, **32**, 12760-–12770.

Corbett-Davies S, Pierson E, and Feller A, Goel S, and Huq A (2017). Algorithmic decision making

and the cost of fairness. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 797-–806, ACM.

Coston A, Ramamurthy K N, Wei D, Varshney KR, Speakman S, Mustahsan Z, and Chakraborty S (2019). Fair transfer learning with missing protected attributes. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 91—98.

Creager E, Madras D, Jacobsen JH, Weis MA, Swersky K, Pitassi T, and Zemel R (2019). Flexibly fair representation learning by disentanglement. In *International Conference on Machine Learning*, 1436—1445, PMLR.

Dixon L, Li J, Sorensen J, Thain N, and Vasserman L (2018). Measuring and mitigating unintended bias in text classification. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 67—73.

Donini M, Oneto L, Ben-David S, Shawe-Taylor J, and Pontil M (2018). Empirical risk minimization under fairness constraints. In *Advances in Neural Information Processing Systems*, **31**, 2791-–2801.

Dua D and Graff C (2017). UCI machine learning repository.

Dwork C, Hardt M, Pitassi T, Reingold O, and Zemel R (2012). Fairness through awareness. In *Proceedings of the 3rd innovations in theoretical computer science conference*, **10**, 214—226.

Feldman M, Friedler SA, Moeller J, Scheidegger C, and Venkatasubramanian S (2015). Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 259—268, ACM.

Fish B, Kun J, and Lelkes ÁD (2016). A confidence-based approach for balancing fairness and accuracy. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, 144—152, SIAM.

Goh G, Cotter A, Gupta M, and Friedlander M (2016). Satisfying real-world goals with dataset constraints. In *Advances in Neural Information Processing Systems*, **29**, 2415-–2423.

Hardt M, Price E, and Srebro N (2016). Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems*, 3315-–3323.

Ingold D and Soper S (2016).*Amazon Doesn't Consider the Race of Its Customers. Should it*, Bloomberg, April .

Jiang R, Pacchiano A, Stepleton T, Jiang H, and Chiappa S (2020). Wasserstein fair classification. In *Uncertainty in Artificial Intelligence*, 862-–872, PMLR.

Kamiran F and Calders T (2012). Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, **33**, 1-–33.

Kamiran F, Karim A, and Zhang X (2012). Decision theory for discrimination-aware classification. In *2012 IEEE 12th International Conference on Data Mining*, 924-–929, IEEE.

Kamishima T, Akaho S, Asoh H, and Sakuma J (2012). Fairness-aware classifier with prejudice remover regularizer. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 35-–50, Springer.

Kleinberg J, Ludwig J, Mullainathan S, and Rambachan A (2018). Algorithmic fairness. In *Aea Papers and Proceedings*, **108**, 22-–27.

Kusner MJ, Loftus J, Russell C, and Silva R (2017). Counterfactual fairness, In *Advances in Neural Information Processing Systems*, **30**, 4066-–4076.

Mehrabi N, Morstatter F, Saxena N, Lerman K, and Galstyan A (2021). A survey on bias and fairness in machine learning, *ACM Computing Surveys (CSUR)*, **54**, 1–35.

Menon AK and Williamson RC (2018). The cost of fairness in binary classification, *In Conference on Fairness, Accountability and Transparency*, 107—118, PMLR.

Narasimhan H (2018). Learning with complex loss functions and constraints, In *International Conference on Artificial Intelligence and Statistics*, 1646-–1654. PMLR.

Pleiss G, Raghavan M, Wu F, Kleinberg J, and Weinberger KQ (2017). On fairness and calibration. In *Advances in Neural Information Processing Systems*, 5680-–5689.

Quadrianto N, Sharmanska V, and Thomas O (2019). Discovering fair representations in the data domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8227—8236.

Vogel R, Bellet A, and Clémençon (2020). *Learning Fair Scoring Functions: Fairness Definitions, Algorithms and Generalization Bounds for Bipartite Ranking*, arXiv preprint arXiv:2002.08159.

Webster K, Recasens M, Axelrod V, and Baldridge J (2018). Mind the gap: A balanced corpus of gendered ambiguous pronouns, *Transactions of the Association for Computational Linguistics*, **6**, 605-–617.

Wei D, Ramamurthy KN, and du Pin Calmon F (2021). Optimized Score Transformation for Fair Classification. In *Proceedings of Machine Learning Research*, **108**, 1673—1683.

Xu D, Yuan S, Zhang L, and WU X (2018). Fairgan: Fairness-aware generative adversarial networks. In *2018 IEEE International Conference on Big Data (Big Data)*, 570-–575. IEEE.

Zafar MB, Valera I, Rogriguez MG, and Gummadi KP (2017). Fairness constraints: Mechanisms for fair classification, In *Artificial Intelligence and Statistics*, 962-–970.

Zafar MB, Valera I, Rodriguez MG, and Gummadi KP (2019). Fairness constraints: A flexible approach for fair classification, *Journal of Machine Learning Research*, **20**, 1-–42.

Zemel R, Wu Y, Swersky K, Pitassi T, and Dwork C (2013). Learning fair representations. In *International Conference on Machine Learning*, 325-–333, PMLR.

Received October 13, 2021; Revised November 30, 2021; Accepted December 26, 2021