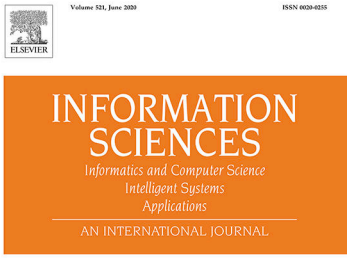
Journal Pre-proofs



Fairness Improvement for Black-box Classifiers with Gaussian Process

Dang Nguyen, Sunil Gupta, Santu Rana, Alistair Shilton, Svetha Venkatesh

PII: S0020-0255(21)00686-1
DOI: https://doi.org/10.1016/j.ins.2021.06.095
Reference: INS 16652

To appear in: *Information Sciences*

Available online at www.sciencedirect.com
ScienceDirect

Received Date: 18 December 2020
Accepted Date: 27 June 2021

Please cite this article as: D. Nguyen, S. Gupta, S. Rana, A. Shilton, S. Venkatesh, Fairness Improvement for Black-box Classifiers with Gaussian Process, *Information Sciences* (2021), doi: https://doi.org/10.1016/j.ins.2021.06.095

Fairness Improvement for Black-box Classifiers with Gaussian Process

Dang Nguyen*, Sunil Gupta, Santu Rana, Alistair Shilton, Svetha Venkatesh

Applied Artificial Intelligence Institute (A^2I^2), Deakin University, Geelong, Australia
{d.nguyen, sunil.gupta, santu.rana, alistair.shilton, svetha.venkatesh}@deakin.edu.au

Abstract

In many real-world applications, it is the fairness, not the accuracy, of a machine learning (ML) classifier that is the crucial factor. Post-processing approaches are widely considered as successful tools to improve the fairness of *black-box* ML classifiers. These aim to learn a *relabeling function* to modify initial predicted labels provided by a pre-trained "unfair" classifier, resulting in fair classification on a given test set. However, many post-processing methods require a training set with true labels to learn the relabeling function. To the best of our knowledge, there have been only two methods that learn the relabeling function without requiring the true labels of training samples. However, both of these methods require access to the predictions of the pre-trained classifier when performing on the test set, even after they learned the optimal relabeling function, and neither offers theoretical guarantees on the trade-off between accuracy loss and fairness improvement. In this paper, we propose a novel post-processing method based on Gaussian process (GP). We first train a GP with *unlabeled* samples, and use its posterior mean function to approximate the predictions of the pre-trained classifier. We then adjust the mean function (i.e. the relabeling function) to achieve two goals: (1) maximize the fairness and (2) minimize the difference between the relabeling function and the pre-trained classifier. By doing this, our method can improve fairness while maintaining high accuracy. We provide a theoretical analysis to derive an upper bound on accuracy loss for our method. We demonstrate our method on four real-world datasets, comparing with state-of-the-art baselines, to demonstrate its ability to achieve both fairness and accuracy.

Keywords: Fair classifier; Post-processing; Gaussian process; Classification; Discrimination-aware data mining; Fairness in machine learning; Black-box pre-trained classifier.

1. Introduction

Recently, machine learning (ML) models (classifiers) have had impressive successes in many real-world domains including loan approvals [1], screening of resumes [2], and decision making in the criminal justice systems [3]. However, like humans, ML models are prone to biases that make their decisions "unfair" to vulnerable sections of our communities (i.e. they make *negative* decisions toward a *unfavored* groups of people identified by a *sensitive* feature). Well-known examples include algorithms being unfairly harsh towards African-American people in granting parole [3], and a Curriculum Vitae screening algorithm unfairly rejecting resumes of capable women applicants [4].

*Corresponding author

Many solutions have been proposed in recent years to address this problem. These can be categorized into three groups: *pre-processing*, *in-processing*, and *post-processing*. The goal of pre-processing approaches is to remove the underlying discrimination from the training data so that any ML model applied to the data will be fair [5, 6, 7]. In-processing approaches modify traditional learning algorithms to remove discrimination during the training phase [8, 9, 10], resulting in a fair classifier regardless of biases in the input. Post-processing approaches take the output of any pre-trained classifier and modify it to be fair on a given test set [11, 12, 13, 14, 15]. Compared to pre-processing and in-processing approaches, post-processing approaches have two significant advantages. First, they do not require access to the original training data used to construct the pre-trained model. Second, since they only treat the pre-trained model as a *black-box* function without requiring access to the internals of learning algorithm, they are applicable to *any* ML model (this is in contrast to in-processing approaches, where the method is often applicable only to a specific ML model). An example of post-processing is illustrated in Figure 1.

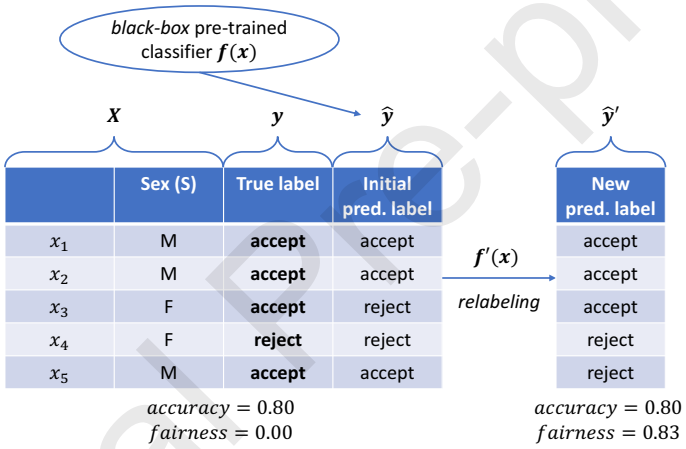


Figure 1: Illustration of post-processing approach for fairness improvement. A *black-box* pre-trained classifier $f(x)$ (e.g. a home loan approval system) achieves a high prediction performance ($accuracy = 0.80$) on a test set $D = [\boldsymbol{X}, y]$ (e.g. a list of applicants), but suffers from a high discrimination between "male" and "female" applicants ($fairness = 0.00$). Specifically, $f(x)$ rejects all applications by women and accepts all applications by men. A post-processing approach finds a *relabeling function* $f'(x)$ to change the initial predicted labels of a small set of samples such that $f'(x)$ maintains good classification performance ($accuracy = 0.80$) but improves fairness ($fairness = 0.83$). Here, the *demographic parity* measure [16] is used: $fairness = 1 - |P(\hat{y} = accept \mid Sex = M) - P(\hat{y} = accept \mid Sex = F)|$ for $f(x)$, which is maximized when the ratio of predictive positive outcomes (i.e. accept) between the two groups ("male" and "female") is balanced.

An interesting observation from Figure 1 is that we can change the initial predicted label of *any* sample in the test set to improve the fairness score since the metric is computed based on predicted labels, not true labels. Consequently, one simple approach is to randomly choose samples and change their initial predicted labels such that the number of samples receiving predicted positive outcomes ("accept") in "male" and "female" groups is balanced. Although this naive approach can improve the fairness, it can potentially reduce the accuracy. Several post-processing approaches have been proposed that take the trade-off between accuracy and fairness into account. Most require a training set

with true labels to generate the *relabeling function* [12, 13, 14], rendering them inapplicable to many real-world applications where the labels either are not given or are difficult to obtain. To the best of our knowledge, there have been only two post-processing methods [11, 15] that can learn the relabeling function based on a training set without requiring true labels. Since these two methods do not use the true labels when learning the relabeling function, they are considered to be *unsupervised learning* methods. *Reject Option based Classification* (ROC) [17, 11] attempts to find an optimal *critical region* where all samples whose initial predicted scores lie in this region are relabeled. Since the region relabeled by ROC is small, the number of relabeled samples will also be small, which ensures that the impact on accuracy caused by the relabeling is minimized. *Individual Group Debiasing* (IGD) [15] improves the fairness based on *biased scores*. It does this by finding *unfavored* samples (e.g. "female" applicants) whose initial predicted scores are different if their sensitive feature is changed (e.g. "female" $\rightarrow$ "male"), and then relabeling them. Although both ROC and IGD can improve fairness while maintaining accuracy, they are *instance-based* methods i.e. they choose samples to relabel based on the initial predicted scores of the black-box pre-trained model. Consequently, they need to query the pre-trained model when they are applied to a test set even after the optimal relabeling function has been learned. Another disadvantage of these methods is that they need to access the sensitive feature of the test set when performing relabeling. In many real-world applications, especially health-care related, sensitive data (e.g. gender of patients) may be anonymized due to privacy policies, rendering these sensitive features invisible. Consequently, neither ROC nor IGD can be applied in such cases. Moreover, neither ROC nor IGD offers a theoretical guarantee on the trade-off between accuracy loss and fairness improvement.

Our approach. To overcome the weaknesses of existing approaches, we propose a novel *Gaussian process* (GP) based method for post-processing the predictions of a pre-trained ML classifier to make them fair on the test set. Instead of choosing a set of samples to relabel, our goal is to model the whole pre-trained classifier and modify it to be fair. In particular, we treat the prediction of the pre-trained classifier as a *black-box* function and learn a relabeling function to achieve two goals: (1) minimize the distance (difference) between the pre-trained black-box classifier and the relabeling function to ensure the accuracy will not change too much and (2) maximize the fairness (i.e. minimize the discrimination) w.r.t the sensitive feature. We call our method *Fair Classifier with Gaussian Process* (**FCGP**). Since **FCGP** formalizes the problem as an optimization problem, it can learn a fairer relabeling function, also ensuring that the classification performance does not drop significantly. More importantly, after learning an optimal relabeling function, our method does not need to access either the predictions of pre-trained classifier or the sensitive feature when applied to a *hold-out* test set. This reduces expense as accessing the pre-trained model is often very expensive, if, for example, the pre-trained model is developed and deployed on the cloud by a third-party company who charges per access.

Our contributions. To summarize, we make the following contributions:

1. We propose **FCGP**, a Gaussian process-based method, for post-processing the predictions of a *black-box* pre-trained classifier to improve its fairness. Unlike existing methods, our method does not require access to either

the predictions of the pre-trained classifier or the sensitive feature after the optimal relabeling function has been learned.

2. We provide a theoretical analysis to prove that **FCGP** may reduce the accuracy but this reduction is bounded. As far as we know, our method is the first *unsupervised* post-processing algorithm that provides the upper bound for the trade-off between fairness improvement and accuracy loss.
3. We demonstrate **FCGP** on four standard datasets (each of them has two sensitive features) to show it improves fairness and maintains accuracy close to that of the pre-trained classifier. Our experimental results also show that our method is better than or comparable to state-of-the-art baselines.

The remainder of the paper is organized as follows. In Section 2, related works on fairness measures and fair algorithms are comprehensively reviewed. Our main contributions are presented in Section 3, in which an unsupervised post-processing algorithm for fairness improvement using Gaussian process is described. Section 4 provides the theoretical analyses of our proposed method, including proof that our method may reduce the accuracy of pre-trained classifier, but this reduction is bounded. Experimental results are discussed in Sections 5 while conclusions and future works are represented in Section 6.

2. Related Work

2.1. Notions of fairness

There are two main notions of fairness in decision making: *group fairness* and *individual fairness*. Group fairness uses a *sensitive* feature (e.g. Sex, Race, or Religion) to partition a population into a *favored* group (e.g. "male" applicants) and an *unfavored* group (e.g. "female" applicants), and then aims to ensure that some statistical measure be equal across two groups. Many different statistical measures have been proposed, including *disparate impact* [18], *calibration* [13], or *equalized odds* [12]. Among them, *demographic parity* [16] is one of the most widely-used. Individual fairness aims to ensure that similar individuals are treated similarly (i.e. they should receive similar classification outcomes) [19]. When checking for the individual fairness, one major challenge is to define a notion of the distance between two individuals to measure their similarity. Consequently, group fairness is more commonly used when developing a fair machine learning classifier.

2.2. Bias mitigation algorithms

Works on fair classification can be categorized into three groups: pre-processing, in-processing, and post-processing [20, 21].

Pre-processing. These approaches primarily involve massaging the training data to remove bias. Some examples include [5, 6, 7, 22, 23]. Many of these methods change the true labels and features of training data, which may have legal implications since the ML model is trained on a "fake" dataset [24].

In-processing. These approaches typically modify a specific ML algorithm to create a fair classifier. Most enforce fairness by introducing constraints in the optimization problem [8, 10, 25, 26] or adding penalties to the objective function [27, 28, 29, 30]. Some frame the problem as a two-player game [31, 32].

Post-processing. These approaches focus on relabeling the predictions of a pre-trained classifier to make them fair on a given test set [11, 12, 13, 17, 14, 15, 33]. Compared to pre-processing and in-processing approaches, post-processing approaches have two major advantages. First, they only treat the pre-trained model as a *black-box* function without requiring access to its internal learning algorithm (e.g. model parameters or derivatives), making them applicable to *any* ML model. Second, they do not require access to the original training data of the pre-trained model for learning a fair classifier.

To train the bias mitigation algorithm (i.e. learn the optimal relabeling function), most post-processing approaches require a training set with true labels. For example, [12] assumed that a *labeled* training set for learning the relabeling function was available, and changed the initial decision boundary of the pre-trained model to achieve two fairness measures: *equalized odds* and *equal opportunity*. This framework was later extended with probabilistic classifiers in [13]. The equal opportunity measure was also used in a semi-supervised method that first estimated the class condition probability using labeled training data, and then estimated the unknown decision threshold using unlabeled training data [33]. Two active learning approaches [34] were developed for fair classification, which required a training set with true labels to achieve both group fairness and intra-group fairness. Similarly, [14] also required access to a labeled training set to learn an auditing algorithm to identify which subgroups were biased, then post-processed to ensure a fair classification across all subgroups. Although these methods can improve the fairness of pre-trained classifier, their success relies on the availability of true labels for all training examples, a condition often not met in many real-world applications. As far as we know, only two methods ROC [17, 11] and IGD [15] do not need the ground truth labels of training samples when learning the optimal relabeling function. However, these two methods still require access to the predictions of the pre-trained classifier and the sensitive feature when operating on an *unseen* test set. More importantly, neither ROC nor IGD has theoretical guarantees on the trade-off between accuracy loss and fairness improvement.

Several methods have been proposed for *fair regression* [35, 36, 37]. However, *fair regression* is not our focus in this paper since we propose a novel method for *fair classification*. It is important to note that *fair classification* is totally different from *fair regression.* Since the data in regression tasks do not have labels (they only have real-valued scores), the fairness measures in classification tasks cannot be applied to regression tasks.

3. Framework

3.1. Problem definition

Let $f(x)$ be a classifier and $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$ be a dataset. Each sample $x_i \in \mathcal{D}$ has a *sensitive* feature S_i (e.g. Sex or Race), where $S_i = 1$ is the *favored* group (e.g. Sex="male") and $S_i = 0$ is the *unfavored* group (e.g.

Sex="female"). Each $y_i \in \{0, 1\}$ is a binary *true label*, where $y_i = 1$ is the *positive* outcome (e.g. Loan="accept") and $y_i = 0$ is the *negative* outcome (e.g. Loan="reject"). Given a sample $x_i \in \mathcal{D}$, $f(x_i)$ provides a probability (called *predicted score*) that x_i belongs to label 1 (i.e. $f(x_i) = P(y_i = 1 \mid x_i)$ and $f(x_i) \in [0, 1]$). We denote the *predicted label* of x_i as $\hat{y}_i \in \{0, 1\}$, where $\hat{y}_i$ is the rounding of $f(x_i)$ (i.e. $\hat{y}_i = 1$ if $f(x_i) \geq 0.5$, otherwise $\hat{y}_i = 0$).

Definition 1. (***Accuracy***). We define *accuracy* as $P(\hat{y} = y)$, which means the percentage of samples in $\mathcal{D}$ predicted correctly by $f(x)$.

Definition 2. (***Fairness***). We define *fairness* as $1 - |P(\hat{y} = 1 \mid S = 1) - P(\hat{y} = 1 \mid S = 0)|$, which means the samples in both favored and unfavored groups should have equal probability of being assigned to a positive outcome. Here, our definition of fairness is also known as *demographic parity* [16].

Problem statement. Given a *black-box* pre-trained classifier $f(x)$ and a small *unlabeled* training set $\mathcal{D}_f = \{x_i\}_{i=1}^N$, we assume $f(x)$ achieves *high accuracy* but *low fairness* on $\mathcal{D}_f$. This assumption is reasonable since $f(x)$ is assumed to have been pre-trained on a much larger training data compared to $\mathcal{D}_f$, with a focus on high accuracy, not fairness. Our goal is to learn a *relabeling function* $f'(x)$ that modifies predicted labels of $f(x)$ such that $f'(x)$ maintains high accuracy (close to that of $f(x)$) but with improved fairness on $\mathcal{D}_f$.

Like other unsupervised learning methods [11, 15], our method uses the predicted score $f(x_i)$ of pre-trained classifier for each sample $x_i \in \mathcal{D}_f$ to generate an auxiliary training set $\mathcal{D}_f = \{x_i, f(x_i)\}_{i=1}^N$, which is used to learn the relabeling function. Then, both the pre-trained classifier $f(x)$ and the relabeling function $f'(x)$ are evaluated on a *hold-out* test set in terms of accuracy and fairness. The process of learning and evaluating the relabeling function is illustrated in Figure 2.

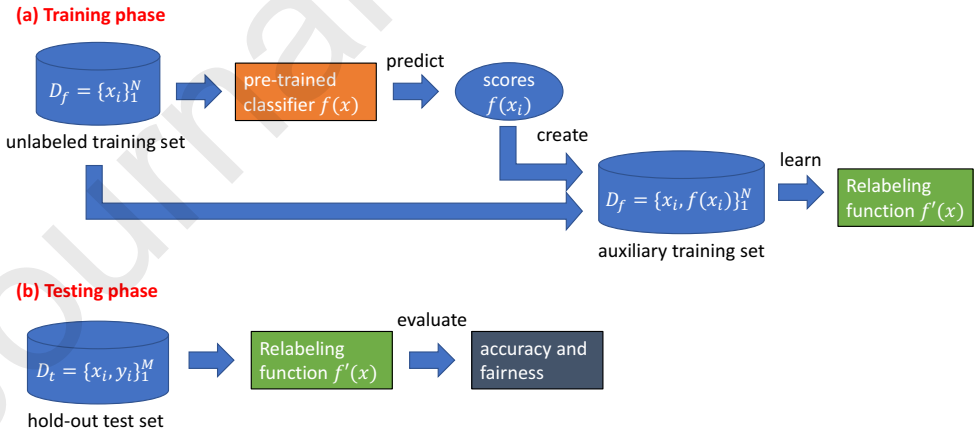


Figure 2: Training and evaluating process for the relabeling function. In the training phase (a), given an unlabeled training set $\mathcal{D}_f = \{x_i\}_1^N$, the black-box pre-trained classifier is used to predict the score $f(x_i)$ for each sample $x_i \in \mathcal{D}_f$. Then, each sample x_i along with its predicted score is used to generate an auxiliary training set to learn the relabeling function $f'(x)$. In the testing phase (b), the learned relabeling function $f'(x)$ is evaluated on a hold-out test set $\mathcal{D}_t$ to compute both accuracy and fairness.

3.2. Proposed method FCGP

To improve the fairness score of $f(x)$, we can simply select random samples in the unfavored group (i.e. $S = 0$) and change their predicted labels from negative outcome (i.e $\hat{y} = 0$) to positive outcome (i.e. $\hat{y} = 1$), which results in the balance between $P(\hat{y} = 1 \mid S = 1)$ and $P(\hat{y} = 1 \mid S = 0)$. However, we expect this simple approach will also reduce the accuracy significantly as there is no mechanism to maintain the accuracy of $f(x)$. Alternative solutions are to carefully select a small set of samples to relabel e.g. ROC [11] chooses *uncertain* samples whose initial predicted scores are close to 0.5 while IGD [15] chooses *biased* samples whose initial predicted scores are different if their sensitive values are changed. However, as discussed in Section 1, both ROC and IGD are *instance-based* methods, making them reliant on the pre-trained classifier even after learning an optimal relabeling function.

Our strategy to solve the problem as discussed in Section 3.1 differs from ROC and IGD. Instead of selecting a subset of training samples to relabel, we learn a relabeling function that approximates the prediction of the pre-trained classifier on the whole space, and adjust this relabeling function to be fair on the training set. Thus, after learning an optimal relabeling function we can use it as a fair classifier on a new test set without further accesses to the pre-trained classifier (i.e. the relabeling function is used *independently* of the pre-trained classifier). We formalize our proposal as an optimization problem as follows:

$$\underset{f'(x)}{\text{argmin}} \underbrace{\frac{1}{N}\sum_{i=1}^{N}|f'(x_i) - f(x_i)|}_{\text{Term-1}} + \underbrace{|P(\hat{y}' = 1 \mid S = 1) - P(\hat{y}' = 1 \mid S = 0)|}_{\text{Term-2}} \tag{1}$$

where $f'(x)$ is the relabeling function. Similar to $f(x)$, given a sample $x_i \in \mathcal{D}_f$, $f'(x_i)$ provides the *predicted score* of x_i and $\hat{y}'_i \in \{0, 1\}$ is the *predicted label* of x_i (i.e. $\hat{y}'_i$ is the rounding of $f'(x_i)$).

Our objective function in Equation (1) has two terms. In **Term-1** (*diff*), we minimize the *difference* between the relabeling function $f'(x)$ and the pre-trained classifier $f(x)$ so that $f'(x)$ can maintain the high accuracy. In **Term-2** (*disc*), we minimize the *discrimination* (i.e. maximize the fairness) score of $f'(x)$ w.r.t the sensitive attribute S.

To find the optimal $f'(x)$, we propose a novel method based on a Gaussian process (GP) [38, 39, 40]. Our method, called *Fair Classifier with Gaussian Process* (**FCGP**), has two main steps: (1) modeling $f(x)$ using a GP and (2) finding the optimal relabeling function $f'(x)$.

3.2.1. Modeling the black-box pre-trained classifier $f(x)$ using GP

Since $f(x)$ is a *black-box* function, we do not know its closed form and can only observe its predicted score $f(x_i) \in [0, 1]$ for each sample $x_i \in \mathcal{D}_f$. To find a relabeling function $f'(x)$ as close as $f(x)$, we first need to model $f(x)$.

To model $f(x)$, we choose to use a GP model, which is one of the most popular methods for modeling black-box functions [38, 39]. We assume that $f(x)$ is a continuous function drawn from a GP i.e. $f(x) \sim \text{GP}(\mu(x), k(x, x'))$,

where $\mu(x)$ is a prior *mean function* (this can be safely assumed to be a zero function) and $k(x, x')$ is a *covariance function* that models the covariance between any two function values $f(x)$ and $f(x')$. A common covariance function is the *squared exponential* kernel [40], $k(x, x') = \sigma_k^2 \exp(-\frac{1}{2l^2}||x - x'||_2^2)$, where σ_k^2 is a parameter dictating the uncertainty in $f(x)$, and l is a length-scale parameter indicating the smoothness of $f(x)$.

To update the belief about $f(x)$, we use the training set $\mathcal{D}_f = \{x_i, f(x_i)\}_{i=1}^N$ (i.e. observations of $f(x)$) to compute the posterior distribution of $f(x)$. The *predictive distribution* for $f(x)$ conditioned on $\mathcal{D}_f$ is also a Gaussian distribution i.e. $f(x) \mid \mathcal{D}_f \sim \mathcal{N}(\mu_f(x), \sigma_f^2(x))$. Following [38, 39], we compute the posterior *mean function* $\mu_f(x)$ and the *variance function* $\sigma_f^2(x)$ as follows:

$$\mu_f(x) = \boldsymbol{k}^\mathsf{T} K^{-1} f_{1:N} \tag{2}$$

$$\sigma_f^2(x) = k(x, x) - \boldsymbol{k}^\mathsf{T} K^{-1} \boldsymbol{k} \tag{3}$$

where $f_{1:N} = [f(x_1), f(x_2), ..., f(x_N)]$ is a vector of predicted scores of the pre-trained classifier, K is a covariance matrix of size $N \times N$ with (i, j)-th element defined as $k(x_i, x_j)$ ($x_i, x_j \in \mathcal{D}_f$), and $\boldsymbol{k} = [k(x_i, x)]_{\forall x_i \in \mathcal{D}_f}$ is a vector containing the corvariance between a new point x and all observed points x_i in the training set $\mathcal{D}_f$.

Since the mean function $\mu_f(x)$ in Equation (2) can approximate the predicted score of $f(x)$ at *any* point x, we use it as the relabeling function $f'(x)$[2].

3.2.2. Finding the optimal relabeling function $f'(x)$

After obtaining a GP mean function $\mu_f(x)$ to approximate the pre-trained classifier $f(x)$ in Section 3.2.1, the next question is how we can adjust $\mu_f(x)$ to obtain a fair classification on $\mathcal{D}_f$. From Figure 3, we see that, given a set of samples in the training set $\mathcal{D}_f = \{x_i, f(x_i)\}_{i=1}^N$, the predicted scores of $f(x)$ and $\mu_f(x)$ are exactly the same at each sample. This is because the mean function $\mu_f(x)$ computed in Equation (2) treats the predicted scores of $f(x)$ as *noise-free* observations. Although this behavior achieves the goal of Term-1 in Equation (1) (i.e. minimizing the difference between the relabeling function and the pre-trained classifier down to 0), it cannot achieve the goal of Term-2. In other words, we cannot use $\mu_f(x)$ to change the predicted labels given by $f(x)$. This leads to the problem that $\mu_f(x)$ can maintain the accuracy of $f(x)$ but cannot improve the fairness score.

[2]For the remaining of paper, we use the mean function $\mu_f(x)$ and the relabeling function $f'(x)$ interchangeably.

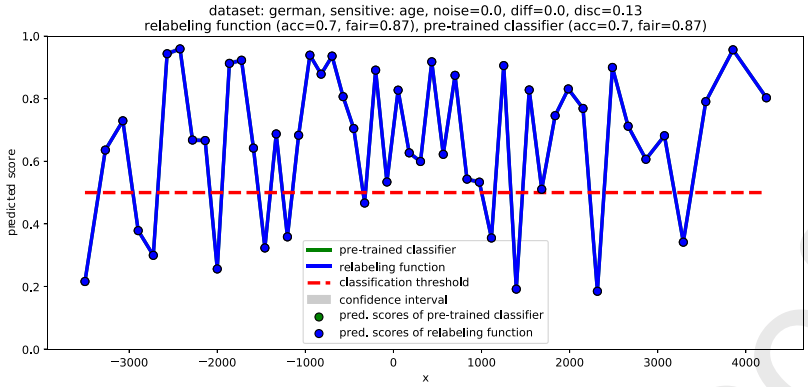


Figure 3: Modeling the black-box pre-trained classifier $f(x)$ (green line) via a GP mean function $\mu_f(x)$ (blue line). Since the evaluation of $f(x)$ is *noise-free*, the mean function $\mu_f(x)$ has the same predicted scores as $f(x)$ at all samples in the training set $\mathcal{D}_f$. Although the difference between two functions is minimized ($diff = \frac{1}{N}\sum_{i=1}^{N} |\mu_f(x_i) - f(x_i)| = 0$) i.e. $\mu_f(x)$ maintains the same accuracy ($acc = 0.7$) as $f(x)$, the fairness score ($fair = 0.87$) of $\mu_f(x)$ is not improved.

To solve this problem, we assume that each predicted score given by $f(x_i)$ is perturbed by noise noise $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. Thus, when fitting a GP to the set of noisy samples $\mathcal{D}_f = \{x_i, f(x_i) + \epsilon_i\}_{i=1}^{N}$, we have an in-built error level for each predicted score of $\mu_f(x)$. In other words, the mean function $\mu_f(x)$ predicts a different score from $f(x)$ at each sample in the training set $\mathcal{D}_f$, depending on the noise variance σ^2. This allows us to optimize σ^2 to generate $\mu_f(x)$ such that it achieves a better fairness score while maintaining good accuracy.

To compute the mean function $\mu_f(x)$ with the noise variance σ^2, we replace the covariance matrix K by the following matrix:

$$K = \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_N) \\ \dots & \dots & \dots \\ k(x_N, x_1) & \dots & k(x_N, x_N) \end{bmatrix} + \sigma^2 I,$$

where I is an identity matrix with the same dimension as K.

With the new covariance matrix K and noise variance σ^2, the posterior *mean function* $\mu_f(x)$ and the *variance function* $\sigma_f^2(x)$ are computed as:

$$\mu_f(x) = \boldsymbol{k}^{\mathsf{T}}[K + \sigma^2 I]^{-1} f_{1:N} \tag{4}$$

$$\sigma_f^2(x) = k(x, x) - \boldsymbol{k}^{\mathsf{T}}[K + \sigma^2 I]^{-1}\boldsymbol{k} \tag{5}$$

Our idea of leveraging a mean function $\mu_f(x)$ with noise variance σ^2 to improve the fairness score is illustrated in Figure 4.

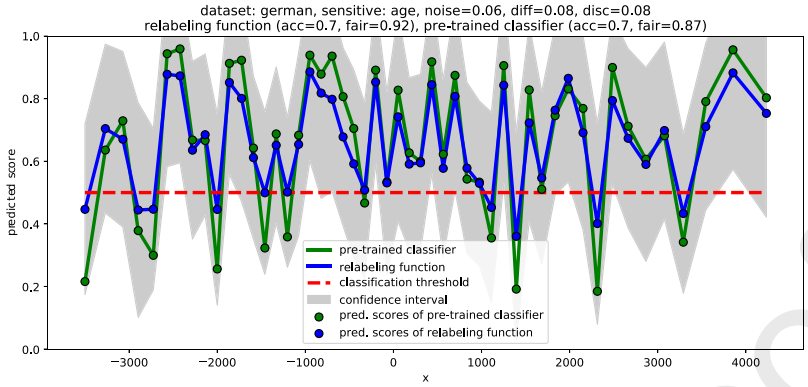


Figure 4: With an assumption that each predicted score of the pre-trained classifier $f(x)$ (green line) has a noise $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ ($\sigma^2 = 0.06$), the mean function $\mu_f(x)$ (blue line) provides different predicted scores (blue dots) from those of $f(x)$ (green dots) for samples in the training set $\mathcal{D}_f$. Some samples are relabeled when the predicted scores of $\mu_f(x)$ are equal or greater than the classification threshold $\alpha = 0.5$ (red dot line). This results in the difference between two functions being increased ($diff = 0.08$) but the fairness score is also improved from 0.87 to 0.92.

As shown in Figure 4, with different noise levels set by σ^2 we obtain different relabeling functions $f'(x)$. Our next step is to optimize σ^2 to obtain the optimal $f'(x)$ that minimizes the objective function in Equation (1) (i.e. minimizing both difference and discrimination). To optimize σ^2, we use grid search as in [11, 15]. Alternative solutions can be a local optimizer with multi-starts or Thompson sampling in Bayesian optimization [41].

Our **FCGP** is summarized in Algorithm 1. **FCGP** receives a training set (without true labels) $\mathcal{D}_f$ as input. It uses a grid search to find the optimal noise variance σ^2_*, and then computes the optimal relabeling function $\mu^*_f(x)$. Since **FCGP** returns the optimal relabeling function $\mu^*_f(x)$ as output, $\mu^*_f(x)$ is used as a *fair classifier* to relabel samples in the training set $\mathcal{D}_f$ or new samples in an *unseen* test set.

3.3. A variant of FCGP focusing on fairness improvement

As shown in Equation (1), our objective function tries to balance accuracy (Term-1) and fairness (Term-2) i.e. it finds a relabeling function $f'(x)$ that improves the fairness score but remains close to the pre-trained classifier $f(x)$. This objective function is intuitive and works well in most cases where it can improve the fairness. However, there is one case where the relabeling function $f'(x)$ may not improve the fairness score. If the samples in the training set are dense i.e. they are nearby each other, the objective function tends to find $f'(x)$ that is very similar to $f(x)$, which keeps both accuracy and fairness unchanged. This is because whenever $f'(x)$ changes the predicted score of an sample x_i, it also changes the predicted scores of x_i's neighbor, leading to the difference between two functions increasing significantly.

Since our goal is to improve the fairness of pre-trained classifier $f(x)$, we propose a new objective function as

Input: $\mathcal{D}_f = \{x_i, f(x_i)\}_{i=1}^N$: training set without true labels, T: # of iterations

begin

define a grid of noise variances $[\sigma_1^2, \sigma_2^2, ..., \sigma_T^2]$;

fit a GP (i.e. $P(f(x) \mid \mathcal{D}_f)$) using $\mathcal{D}_f$;

for $t = 1, 2, ..., T$ **do**

compute $diff = \frac{1}{N}\sum_{i=1}^N |\mu_f(x_i) - f(x_i)|$, with $\mu_f(x_i) = \boldsymbol{k}^\mathsf{T}[K + \sigma_t^2 I]^{-1} f_{1:N}$ Eq. (4);

compute $\hat{y}' = \mathbb{I}_{\forall x \in \mathcal{D}_f}(\mu_f(x) \geq 0.5)$;

compute $disc = |P(\hat{y}' = 1 \mid S = 1) - P(\hat{y}' = 1 \mid S = 0)|$;

compute $score_{\sigma_t^2} = diff + disc$ Eq. (1);

end

choose optimal noise variance $\sigma_*^2 = \arg\min_{\sigma_t^2} score_{\sigma_t^2}$;

compute optimal relabeling function $\mu_f^*(x) = \boldsymbol{k}^\mathsf{T}[K + \sigma_*^2 I]^{-1} f_{1:N}, \forall x \in \mathcal{D}_f$;

end

Algorithm 1: The proposed **FCGP** algorithm.

follows:

$$\underset{f'(x)}{\text{argmin}} \underbrace{\frac{1}{N}\sum_{i=1}^N |\hat{y}'_{x_i} - \hat{y}_{x_i}| + \mathbb{I}(\hat{y}' = \hat{y})}_{\text{Term-1}} + \underbrace{|P(\hat{y}' = 1 \mid S = 1) - P(\hat{y}' = 1 \mid S = 0)|}_{\text{Term-2}} \tag{6}$$

where $f'(x)$ is a relabeling function. Given a sample x_i, $f'(x_i)$ returns a predicted score for x_i and $\hat{y}'_{x_i}$ is a predicted label of x_i (i.e. $\hat{y}'_{x_i} = \mathbb{I}(f'(x_i) \geq 0.5)$).

In Term-1, instead of using the predicted scores, we compute the distance between two functions $f'(x)$ and $f(x)$ using their predicted labels. By doing this, we solve the above problem since we do not care about the difference between $f'(x)$ and $f(x)$ at samples whose labels are unchanged. As long as $f'(x)$ relabels a sample x_i, the difference between $f'(x)$ and $f(x)$ at x_i's neighbor is no longer important. We also add a penalty term $\mathbb{I}(\hat{y}' = \hat{y})$ to Term-1 to ignore relabeling functions that are identical to $f(x)$ (i.e. they do not improve the fairness). When $f'(x) \equiv f(x)$, $\frac{1}{N}\sum_{i=1}^N |\hat{y}'_{x_i} - \hat{y}_{x_i}| = 0$ whereas $\mathbb{I}(\hat{y}' = \hat{y}) = 1$, maximizing Term-1, which is in conflict with our objective to minimize Term-1.

Term-1 consists of $\frac{1}{N}\sum_i |\hat{y}'_{x_i} - \hat{y}_{x_i}|$ and the penalty term $\mathbb{I}(\hat{y}' = \hat{y})$. We minimize $\frac{1}{N}\sum_i |\hat{y}'_{x_i} - \hat{y}_{x_i}|$ to make the accuracy of relabeling function close to that of the pre-trained classifier. We add the penalty term $\mathbb{I}(\hat{y}' = \hat{y})$ to ignore relabeling functions that provide the same predictions as those of the pre-trained classifier because they do not improve the fairness score. For example, if $\hat{y}' = [1, 0, 1, 1]$ and $\hat{y} = [1, 0, 1, 1]$, then there is no different prediction between $\hat{y}'$ and $\hat{y}$, and $\sum_i |\hat{y}'_{x_i} - \hat{y}_{x_i}| = 0$. In this case, $\mathbb{I}(\hat{y}' = \hat{y})$ becomes 1, making the whole Term-1

$= \frac{1}{N}\sum_i |\hat{y}'_{x_i} - \hat{y}_{x_i}| + \mathbb{I}(\hat{y}' = \hat{y}) = 1$. Since our goal is to minimize both Term-1 and Term-2, and the value range of Term-1 is $[0, 1]$, Term-1 $= 1$ (the maximum value) will conflict with our objective, so $\hat{y}' = [1, 0, 1, 1]$ will not be selected by our method.

4. Theoretical Analysis

This section provides a theoretical analysis of our method. Assuming that the *black-box* pre-trained classifier is trained with accurate labels, fairness and accuracy are two conflicting goals, and so increasing the classifier's fairness may lead to some increase in its error rate. In this section, we provide an upper bound on the additional classification error incurred due to our relabeling process that aims to maximize fairness. A sketch of our proof is outlined below.

Since our Gaussian process is a flexible non-parametric regression model that we train using the output of the black-box classifier, without noise its lowest training error rate will be the same as the generalization error rate of the black-box classifier. When treating black-box outputs as noisy observations (which is required in our case to improve the fairness), the Gaussian process has an asymptotic error rate that differs from that of the black-box classifier by at most the noise variance. We utilize a generalization error bound for Gaussian process to bound its error rate in terms of the number of samples in the training set. An interesting property of this bound is that the additional error rate of our Gaussian process asymptotically reduces to the noise variance. Details are as follows.

Let us use $\mathcal{D}_f = \{x_i, f(x_i)\}_{i=1}^N$ to denote a training set sampled from a data distribution P on $X \times F$, where $F \in [0, 1]$ are the predicted scores of X. In our case, $f(x_i)$ is the predicted score of the black-box pre-trained classifier for an instance $x_i \in \mathcal{D}_f$. Then the goal of any non-parametric least squares regression is to find a function $g : X \to \mathbb{R}$ by minimizing a risk functional of the form $\mathcal{R}_{L,P}(g) = \mathbb{E}_{X,F\sim P}\left[L(f(x), g(x))\right] = \int_{X\times F} L(f(x), g(x))dP(x, f(x))$, where L is the least square loss, i.e. $L(a, b) = (a-b)^2$. Assuming a function class such as reproducing kernel Hilbert space (RKHS), the optimal risk is given as:

$$\mathcal{R}^*_{L,P} = \inf\{\mathcal{R}_{L,P}(g) \mid g : X \to \mathbb{R}\}$$

There are many ways to solve the non-parametric least squares regression problem (see [42]). In this work, we use a Gaussian process that is a kernel-based method. Gaussian process regression can be shown to estimate a function through the following formulation:

$$g_{\mathcal{D}_f,\lambda} = \underset{g\in H}{\operatorname{argmin}} \left(\lambda||g||^2_H + \mathcal{R}_{L,\mathcal{D}_f}(g)\right) \tag{7}$$

A solution for the above optimization problem in Equation (7) has the form $g_{\mathcal{D}_f,\lambda}(x) = \boldsymbol{k}^\mathsf{T}[K + \lambda I]^{-1} f_{1:N}$, where $\mathbf{k} = [k(x, x_1), \ldots, k(x, x_N)]$ and $f_{1:N} = [f(x_1), \ldots, f(x_N)]$. In our case, since we fit a Gaussian process with noise variance σ^2, we have $\lambda = \frac{\sigma^2}{N}$. We adapt a generalization bound from [43] to our problem and write as:

$$\mathcal{R}_{L,P}(g_{\mathcal{D}_f,\lambda}) - \mathcal{R}^*_{L,P} \leq 9A(\frac{\sigma^2}{N}) + c\frac{a^p\omega}{\sigma^{2p}N^{1-p}}, \tag{8}$$

where $A(r) = cr^{\beta}$ with $\beta \in (0,1)$ and c being a constant only depending on p and $C \geq 1$ such that $||g||_{\infty} \leq C||g||_H^p \cdot ||g||_{L_2(P_X)}^{1-p}$; and $a \geq 16, \omega > 0, p \in (0,1)$ are constants.

Next, we define the risk of black-box pre-trained classifier $f(x)$ as $\mathcal{R}_{L,P'}(f) = \mathbb{E}_{X,Y\sim P'}\left[L(y, f(x))\right]$ where P' is the distribution of the data $X \times Y$, where Y are the actual labels of X. We note that $\mathcal{R}^*_{L,P} = \mathcal{R}_{L,P'}(f) + \sigma^2$. This is because Gaussian process is a flexible non-parametric regression model that can approximate any function with arbitrary accuracy given a sufficiently large number of samples [44]. Since $\mathcal{R}^*_{L,P}$ is the best possible risk achievable using a Gaussian process assuming access to the true data distribution, fitting a Gaussian process with infinitely many *noise-free* samples will make $\mathcal{R}^*_{L,P}$ equal the risk of black-box pre-trained classifier $\mathcal{R}_{L,P'}(f)$. However, since samples are noisy with noise variance σ^2, we have $\mathcal{R}^*_{L,P}$ exceeding $\mathcal{R}_{L,P'}(f)$ by at most σ^2. Combining this result with Equation (8), we have:

$$\mathcal{R}_{L,P}(g_{\mathcal{D}_f,\lambda}) - \mathcal{R}_{L,P'}(f) \leq \sigma^2 + 9A(\frac{\sigma^2}{N}) + c\frac{a^p\omega}{\sigma^{2p}N^{1-p}}$$

We can see from the above bound that the difference between the risk of our Gaussian process model (i.e. our relabeling function) and that of the pre-trained classifier decreases with the number of samples N in the training set, and asymptotically reduces to σ^2, which has been used to achieve the maximum fairness. This can also be interpreted as a trade-off between the accuracy loss and the fairness improvement.

5. Experiments

In this section, we conduct extensive experiments on four real-world datasets to evaluate the performance (accuracy and fairness) of our proposed method **FCGP**, compared with three state-of-the-art baselines. Our experiments have two settings. (1) We use a training set *without true labels* to learn an optimal relabeling function and then evaluate its performance on the same training set. (2) After learning the optimal relabeling function from the training set, we evaluate its performance on a *hold-out* test set.

5.1. Datasets

We use four standard real-world datasets *Adult* (an income dataset based on a 1994 U.S. Census data), *German* (a credit scoring dataset), *Compas* (a prison recidivism dataset for violent crime), and *Bank* (a direct marketing campaign dataset for term deposit subscription). Since each dataset has two sensitive features, there are eight evaluation datasets in total. These datasets are commonly used to evaluate the performance of a fair classification algorithm [12, 11, 9, 16]. Their characteristics are summarized in Table 1.

5.2. Baselines

We compare our **FCGP** with three state-of-the-art baselines Random, ROC, and IGD. As far as we know, only these three methods can learn a relabeling function based on a training set without requiring true labels.

Table 1: Characteristics of four benchmark datasets. We denote the sensitive feature as S (where $S = 1$ is *favored* group and $S = 0$ is *unfavored* group) and the class feature as y (where $y = 1$ is *positive* outcome and $y = 0$ is *negative* outcome).

Dataset	**# samples**	**# features**	S	$S = 1$	$S = 0$	y	$y = 1$	$y = 0$
Adult	30,162	13	Sex	"male"	"female"	Income	">50K"	"<50K"
			Race	"white"	"non-white"			
German	1,000	20	Age	"adult"	"youth"	Credit	"good"	"bad"
			Sex	"male"	"female"			
Compas	4,010	10	Race	"Caucasian"	"non-Caucasian"	Rearrested	"no"	"yes"
			Sex	"male"	"female"			
Bank	4,521	14	Age	"adult"	"youth"	Subscribed	"yes"	"no"
			Marital	"single"	"not-single"			

1. Random method: at each iteration, this method selects randomly one sample from the training set and relabels it such that the fairness is improved. For example, at iteration t, it draws a sample x. To increase the fairness (i.e. reducing the difference between $P(\hat{y} = 1 \mid S = 1)$ and $P(\hat{y} = 1 \mid S = 0)$), it assigns label $\hat{y}_x = 1$ to x if x has $S_x = 0$. Otherwise, it assigns label $\hat{y}_x = 0$ to x.
2. ROC method [17]: this method relabels *uncertain* samples whose initial predicted scores are in a *critical region* $0.5 - \theta \leq f(x) \leq 0.5 + \theta$, where θ is a *region margin*. Given a uncertain sample x, it is assigned a label $\hat{y}_x = 0$ if it has $S_x = 1$ and is assigned a label $\hat{y}_x = 1$ if it has $S_x = 0$. Other samples whose initial predicted scores are outside the critical region keep the same predicted labels as the initial ones. θ is optimized to achieve the best fairness but as small as possible to minimize the number of relabeled samples (i.e. minimizing the accuracy loss). We implement ROC based on the source code[3] provided in IBM AI Fairness 360 Toolkit [45].
3. IGD method [15]: this method relabels *biased* samples whose bias scores are greater than a *bias threshold* τ. A *bias score* of a sample x is computed as $b_x = f(x \mid S_x = 1) - f(x \mid S_x = 0)$ (i.e. the difference in its initial predicted score if its sensitive feature changes from unfavored to favored group). τ is optimized to achieve the best fairness but as large as possible to minimize the number of relabeled samples (i.e. minimizing the accuracy loss). We implement IGD based on Algorithm 1 in its paper [15].

Our method **FCGP** has two models that differ in the computation of the distance between the relabeling function and the pre-trained classifier. **FCGP-S** model uses the objective function in Equation (1) with predicted *scores* while **FCGP-L** model uses the objective function in Equation (6) with predicted *labels*.

[3] `https://github.com/IBM/AIF360`

5.3. *Implementation details*

We implement our method **FCGP** and all baselines using Python. For a fair comparison, we use a grid search with the same budget of 50 (i.e. the number of iterations) to optimize each model's hyper-parameter, namely the region margin θ in ROC, the bias threshold τ in IGD, and the noise variance σ^2 in our **FCGP**. The standard search range for θ and σ^2 is $[0.0, 0.5]$ while that for τ is $[0.0, 1.0]$. Since the Random method does not have any hyper-parameter, we use the budget as the number of random samples to relabel. Each dataset is randomly split into 90% for training the black-box pre-trained classifier, 5% for the *unlabeled* training set to learn the relabeling function, and 5% for the *hold-out* test set. We train the black-box pre-trained classifier as a neural network with one hidden layer and 32 hidden units. Note that other ML models can be also used for the pre-trained classifier since post-processing approaches are not restricted to any specific classifier. We use the training set without its true labels to learn the optimal relabeling function. We repeat the classification process on each training set and hold-out test set 10 times and report the average accuracy and fairness along with their standard deviations.

5.4. *Performance comparison on training set*

This experiment illustrates how well our models **FCGP-S** and **FCGP-L** perform on a training set. Although this training set is also used to learn the optimal relabeling function, we do not use its true labels during the learning process. Thus, the performance comparison is still reasonable and fair for all methods.

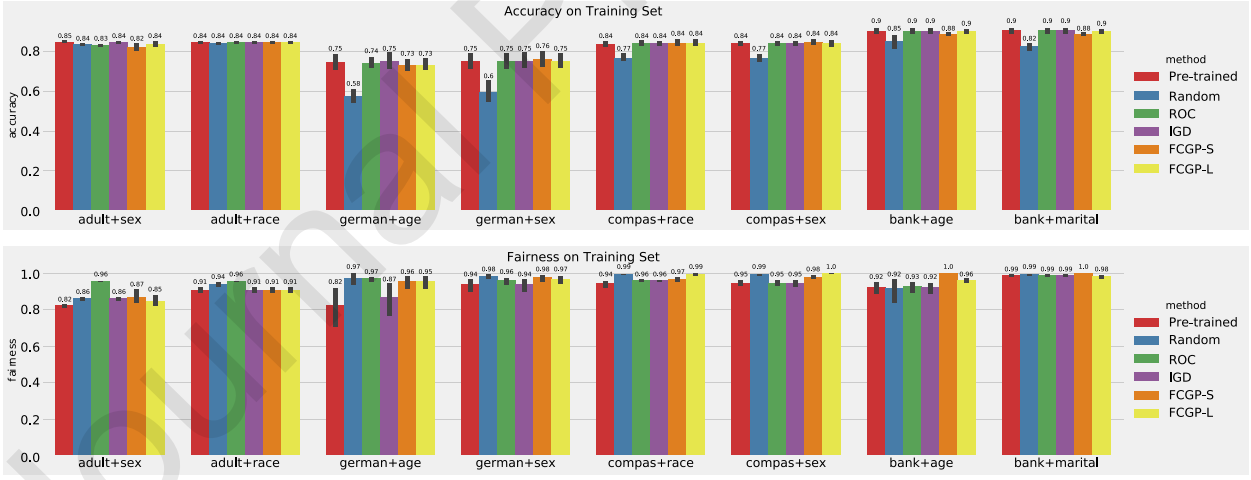


Figure 5: Accuracy and fairness of our models **FCGP-S**, **FCGP-L**, the pre-trained classifier, and state-of-the-art baselines on four datasets (each of them has two sensitive features). The performance of each method is measured on a training set. Although this training set is also used for learning the optimal relabeling function, we do not use its true labels during the learning process. Thus, the performance comparison (e.g. accuracy comparison) is still reasonable and fair for all methods.

From the results shown in Figure 5, we can see that both our models **FCGP-S** and **FCGP-L** improve fairness while maintaining a high accuracy on all datasets except the dataset *adult+race* (i.e. dataset *Adult* with sensitive

feature Race). For example, on *german+age*, **FCGP-S** keeps a similar accuracy as the pre-trained classifier but it improves the fairness significantly (0.82 vs. 0.96).

Random often increases the fairness scores; it however also drops accuracy deeply. For example, on *german+age*, Random increases the fairness up to 0.97 but its accuracy drops 17% (from 0.75 down to 0.58).

ROC works well on most datasets, where it can maintain a high accuracy but improves the fairness. Compared to ROC, our method performs better on four datasets *german+sex*, *compas+race*, *compas+sex*, and *bank+age*.

IGD always keeps the same accuracy as that of the pre-trained classifier. However, it cannot improve the fairness on several datasets. For example, on *adult+race*, *german+sex*, *compas+sex*, *bank+age*, and *bank+marital*, the fairness scores of IGD are unchanged, compared to those of the pre-trained classifier. This problem can be explained by the fact that IGD focuses on relabeling only unfavored samples (e.g. "female" applicants). In contrast, other methods like Random, ROC, and our method can relabel both unfavored and favored samples. Compared to IGD, our model **FCGP-S** is better on seven datasets, namely (*adult+sex*, *german+age*, *german+sex*, *compas+race*, *compas+sex*, *bank+age*, and *bank+marital*) while it is comparable on *adult+race*.

In most cases, our two models behave similarly. However, **FCGP-L** outperforms **FCGP-S** on two datasets *compas+race* and *compas+sex*. After a further investigation, we find that the data points in these two datasets are very close to each other, as shown in Figure 6. Consequently, on such kind of dataset, **FCGP-L** is better than **FCGP-S** in terms of fairness improvement, as explained in Section 3.3.

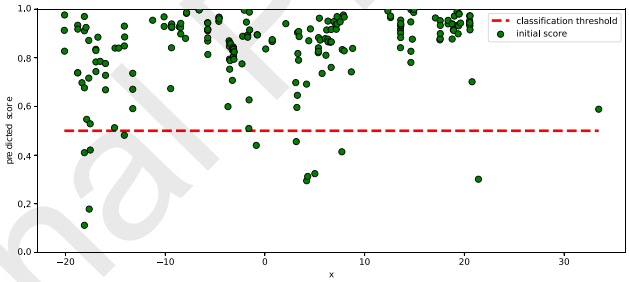


Figure 6: A training set of the dataset *compas+sex* showing its data points are very close to each other. On such kind of dataset, our model **FCGP-L** performs better than **FCGP-S** in terms of fairness improvement, as explained in Section 3.3. To visualize a high-dimension data point, we map it into 1-dimension using t-SNE [46].

5.5. Performance comparison on hold-out test set

After learning the optimal relabeling function in Section 5.4, we evaluate its performance on a *hold-out* test set. Note that different from our method **FCGP** that learns a *real* optimal relabeling function ($\mu_f^*(x)$ in Algorithm 1), both ROC and IGD learn optimal *thresholds* to relabel new unseen samples. In particularly, ROC learns the optimal region margin θ^*, and relabels a test sample x if its initial predicted score belongs to $0.5 - \theta^* \leq f(x) \leq 0.5 + \theta^*$. The new predicted label for x is assigned based on its sensitive feature, as explained in Section 5.2. Similarly, IGD learns the

optimal bias threshold τ^*, and relabels a unfavored test sample x if its bias score $b_x = f(x \mid S_x = 1) - f(x \mid S_x = 0) > \tau^*$.

We see that whenever ROC and IGD perform a relabeling process on a test set, they require access to the pre-trained classifier to obtain its predicted scores on the test set. In many real-word scenarios where the pre-trained classifier is developed and deployed on cloud by a third-party company, accessing the cloud model often costs a certain amount of money. Hence, minimizing the number of accesses to the pre-trained classifier is desirable. Moreover, both ROC and IGD require access to the sensitive feature of test set to perform relabeling. In many real-world applications, especially health-care applications, sensitive features (e.g. patient gender or sexual orientation) are often anonymized due to privacy policies, making the sensitive features invisible. Consequently, neither ROC nor IGD can be applied to these types of sensitive and crucial data.

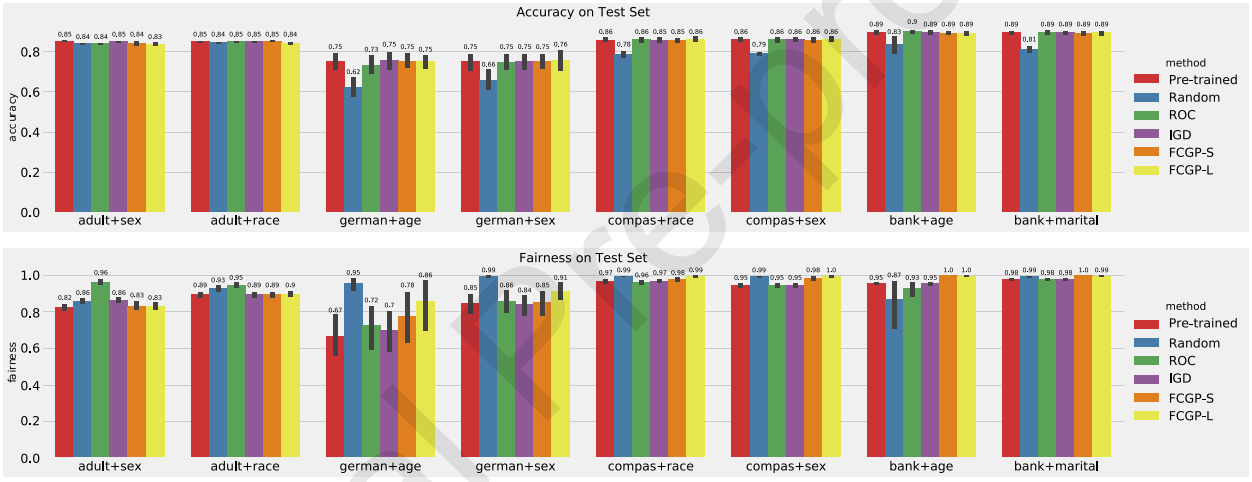


Figure 7: Accuracy and fairness of our models **FCGP-S**, **FCGP-L**, the pre-trained classifier, and state-of-the-art baselines on four datasets (each of them has two sensitive features). The performance of each method is measured on a *hold-out* test set.

Figure 7 reports the accuracy and fairness of all methods on *hold-out* test sets, where we can observe similar results as those in training sets. All methods except Random can maintain a high accuracy, as close as that of the pre-trained classifier. Our models **FCGP-S** and **FCGP-L** often improve the fairness, obtaining very high scores on *german+age*, *german+sex*, *compas+race*, *compas+sex*, and *bank+age*. ROC is often better or comparable with IGD. Our model **FCGP-L** significantly outperforms ROC on five datasets *german+age*, *german+sex*, *compas+race*, *compas+sex*, and *bank+age*. Again, Random often increases the fairness score but fails to maintain high accuracy. This problem is clearly shown in two datasets *german+age* and *german+sex*, where Random significantly drops its accuracy.

We have conducted more experiments with various settings for data split. These results are presented in Appendix A.

6. Conclusion

We have presented a novel *unsupervised* post-processing approach – **FCGP** to improve the fairness of *black-box* pre-trained classifiers. Our method uses the posterior mean function of a Gaussian process to generate a relabeling function, and optimizes it by adjusting the noise variance. Unlike existing unsupervised post-processing methods, our method **FCGP** learns a *real* optimal relabeling function that can be used to classify *unseen* test sets without further accesses to the pre-trained classifier. Moreover, our method does not need to know the sensitive feature on the test set. These two advantages of our method over baselines make it applicable to a wide range of real-world applications, especially in domains where the data are anonymized. We also theoretically analyze the trade-off between accuracy loss and fairness improvement of our method, and prove that our accuracy loss is bounded by the number of samples in the training set. We demonstrate the efficacy of **FCGP** on four standard real-world datasets, each with two sensitive features. The empirical results show that **FCGP** is better or comparable with state-of-the art baselines, improving the fairness of the original black-box pre-trained classifier in most cases..

Discussion: Most existing state-of-the-art methods for fair classification including the baselines ROC and IGD have been developed for binary classification problems [16, 20]. This is because many fairness measures require the concept of "positive outcome" and "negative outcome" in the labels i.e. binary decisions. Aligning with the fairness literature, our method was proposed for binary classification problems.

To adapt to multi-class classification problems, we need to use a relevant fairness measure. One possible option is the *accuracy parity* [16], defined as $1 - |P(\hat{y} = y \mid S = 1) - P(\hat{y} = y \mid S = 0)|$. The accuracy parity encourages the overall accuracy of favored and unfavored groups as close as possible. Our method is straightforwardly applicable to this new fairness measure. By simply replacing the demographic parity in Term-2 of our objective functions (see Equations (1) and (6)) with the accuracy parity, our method can be adapted to multi-class classification tasks.

Future works. In the future, we will investigate the performance of **FCGP** when applied to improving *individual fairness* that requires two samples which are similar with respect to a given measure receive similar classification outcomes [19]. The requirement of individual fairness is naturally incorporated in our framework since one of properties of a Gaussian process is to predict similar labels for similar (nearby) data points.

Acknowledgment

This research was fully supported by the Australian Government through the Australian Research Council's Discovery Projects funding scheme (project DP210102798). The views expressed herein are those of the authors and are not necessarily those of the Australian Government or Australian Research Council.

References

[1] A. Mukerjee, R. Biswas, K. Deb, A. Mathur, Multi–objective evolutionary algorithms for the risk–return trade–off in bank loan management, International Transactions in Operational Research 9 (5) (2002) 583–597.

[2] L. Cohen, Z. Lipton, Y. Mansour, Efficient candidate screening under multiple tests and implications for fairness, arXiv preprint arXiv:1905.11361.

[3] J. Dressel, H. Farid, The accuracy, fairness, and limits of predicting recidivism, Science Advances 4 (1) (2018) eaao5580.

[4] P. Lahoti, K. Gummadi, G. Weikum, iFair: Learning individually fair data representations for algorithmic decision making, in: ICDE, 2019, pp. 1334–1345.

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

[6] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, C. Dwork, Learning fair representations, in: ICML, 2013, pp. 325–333.

[7] F. Calmon, D. Wei, B. Vinzamuri, K. Ramamurthy, K. Varshney, Optimized pre-processing for discrimination prevention, in: NIPS, 2017, pp. 3992–4001.

[8] T. Kamishima, S. Akaho, H. Asoh, J. Sakuma, Fairness-aware classifier with prejudice remover regularizer, in: ECML-PKDD, 2012, pp. 35–50.

[9] M. Zafar, I. Valera, M. Rodriguez, K. Gummadi, A. Weller, From parity to preference-based notions of fairness in classification, in: NIPS, 2017, pp. 229–239.

[10] B. Zhang, B. Lemoine, M. Mitchell, Mitigating unwanted biases with adversarial learning, in: AAAI/ACM Conference on AI, Ethics, and Society, 2018, pp. 335–340.

[11] F. Kamiran, A. Karim, X. Zhang, Decision theory for discrimination-aware classification, in: ICDM, 2012, pp. 924–929.

[12] M. Hardt, E. Price, N. Srebro, Equality of opportunity in supervised learning, in: NIPS, 2016, pp. 3315–3323.

[13] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, K. Weinberger, On fairness and calibration, in: NIPS, 2017, pp. 5680–5689.

[14] M. Kim, A. Ghorbani, J. Zou, Multiaccuracy: Black-box post-processing for fairness in classification, in: AAAI/ACM Conference on AI, Ethics, and Society, 2019, pp. 247–254.

[15] P. Lohia, N. Ramamurthy, M. Bhide, D. Saha, K. Varshney, R. Puri, Bias mitigation post-processing for individual and group fairness, in: International Conference on Acoustics, Speech and Signal Processing, 2019, pp. 2847–2851.

[16] S. Friedler, C. Scheidegger, S. Venkatasubramanian, S. Choudhary, E. Hamilton, D. Roth, A comparative study of fairness-enhancing interventions in machine learning, in: Conference on Fairness, Accountability, and Transparency (FAT), 2019, pp. 329–338.

[17] F. Kamiran, S. Mansha, A. Karim, X. Zhang, Exploiting reject option in classification for social discrimination control, Information Sciences 425 (2018) 18–33.

[18] M. B. Zafar, I. Valera, M. G. Rodriguez, K. Gummadi, Fairness constraints: Mechanisms for fair classification, in: AISTAT, 2017, pp. 962–970.

[19] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, R. Zemel, Fairness through awareness, in: Innovations in Theoretical Computer Science, 2012, pp. 214–226.

[20] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A survey on bias and fairness in machine learning, arXiv preprint arXiv:1908.09635.

[21] L. Oneto, S. Chiappa, Fairness in machine learning, in: Recent Trends in Learning From Data, Springer, 2020, pp. 155–196.

[22] D. Madras, E. Creager, T. Pitassi, R. Zemel, Learning adversarially fair and transferable representations, in: ICML, 2018, pp. 3384–3393.

[23] D. McNamara, C. S. Ong, R. Williamson, Costs and benefits of fair representation learning, in: AAAI/ACM Conference on AI, Ethics, and Society, 2019, pp. 263–270.

[24] S. Barocas, A. Selbst, Big data's disparate impact, California Law Review 104 (2016) 671.

[25] A. Agarwal, A. Beygelzimer, M. Dudik, J. Langford, H. Wallach, A reductions approach to fair classification, in: ICML, 2018, pp. 60–69.

[26] M. Zafar, I. Valera, M. Gomez-Rodriguez, K. Gummadi, Fairness constraints: A flexible approach for fair classification, Journal of Machine Learning Research 20 (75) (2019) 1–42.

[27] Y. Bechavod, K. Ligett, Penalizing unfairness in binary classification, arXiv preprint arXiv:1707.00044.

[28] C. Dwork, N. Immorlica, A. T. Kalai, M. Leiserson, Decoupled classifiers for group-fair and efficient machine learning, in: Conference on

Fairness, Accountability and Transparency (FAT), 2018, pp. 119–133.

[29] R. Nabi, D. Malinsky, I. Shpitser, Learning optimal fair policies, in: ICML, 2019, pp. 4674–4682.

[30] R. Williamson, A. Menon, Fairness risk measures, in: ICML, 2019, pp. 6786–6797.

[31] M. Kearns, S. Neel, A. Roth, Z. S. Wu, Preventing fairness gerrymandering: Auditing and learning for subgroup fairness, arXiv preprint arXiv:1711.05144.

[32] A. Cotter, H. Jiang, K. Sridharan, Two-player games for efficient non-convex constrained optimization, in: Algorithmic Learning Theory, 2019, pp. 300–332.

[33] E. Chzhen, C. Denis, M. Hebiri, L. Oneto, M. Pontil, Leveraging labeled and unlabeled data for consistent fair binary classification, in: NIPS, Vol. 32, 2019, pp. 12739–12750.

[34] A. Noriega-Campero, M. Bakker, B. Garcia-Bulle, A. Pentland, Active fairness in algorithmic decision making, in: AAAI/ACM Conference on AI, Ethics, and Society, 2019, pp. 77–83.

[35] T. Calders, A. Karim, F. Kamiran, W. Ali, X. Zhang, Controlling attribute effect in linear regression, in: ICDM, 2013, pp. 71–80.

[36] R. Berk, H. Heidari, S. Jabbari, M. Joseph, M. Kearns, J. Morgenstern, S. Neel, A. Roth, A convex framework for fair regression, arXiv preprint arXiv:1706.02409.

[37] E. Chzhen, C. Denis, M. Hebiri, L. Oneto, M. Pontil, Fair regression with Wasserstein barycenters, in: NIPS, Vol. 33, 2020, pp. 7321–7331.

[38] J. Snoek, H. Larochelle, R. Adams, Practical bayesian optimization of machine learning algorithms, in: NIPS, 2012, pp. 2951–2959.

[39] B. Shahriari, K. Swersky, Z. Wang, R. Adams, N. Freitas, Taking the human out of the loop: A review of bayesian optimization, Proceedings of the IEEE 104 (1) (2016) 148–175.

[40] D. Nguyen, S. Gupta, S. Rana, A. Shilton, S. Venkatesh, Bayesian optimization for categorical and category-specific continuous inputs, in: AAAI, 2020.

[41] D. Russo, B. Van Roy, A. Kazerouni, I. Osband, Z. Wen, et al., A tutorial on Thompson sampling, Foundations and Trends® in Machine Learning 11 (1) (2018) 1–96.

[42] L. Györfi, M. Kohler, A. Krzyzak, H. Walk, A distribution-free theory of nonparametric regression, Springer Science & Business Media, 2006.

[43] I. Steinwart, D. Hush, C. Scovel, et al., Optimal rates for regularized least squares regression., in: COLT, 2009, pp. 79–93.

[44] B. Sriperumbudur, K. Fukumizu, G. Lanckriet, Universality, characteristic kernels and rkhs embedding of measures, Journal of Machine Learning Research 12 (Jul) (2011) 2389–2410.

[45] R. Bellamy, K. Dey, M. Hind, S. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, et al., AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias, arXiv preprint arXiv:1810.01943.

[46] L. v. d. Maaten, G. Hinton, Visualizing data using t-SNE, Journal of Machine Learning Research 9 (Nov) (2008) 2579–2605.

Appendix A. More Experiments

In this section, we evaluate our method **FCGP** with two additional settings for data split. Our goal is to demonstrate that **FCGP** still works effectively on large test sets, where it improves the fairness score while maintaining an accuracy close to that of the pre-trained classifier.

Appendix A.1. Data split: 70%-15%-15%

For each dataset, we randomly split it into 70% to train the pre-trained classifier, 15% for the *unlabeled* training set to train our method and the baselines, and 15% for the *hold-out* test set. Other settings are the same as those in our previous experiments (see Section 5.3). All methods are evaluated on the *hold-out* test sets.

From Figure A.8, we can see that both our methods **FCGP-S** and **FCGP-L** are the only methods that can achieve fairness improvements without suffering significant accuracy losses on all datasets (except for *bank+marital*). In contrast, two state-of-the-art baselines ROC and IGD cannot increase the fairness scores on some datasets. For example, ROC does not improve the fairness over that of the pre-trained model on *compas+sex*, *bank+age*, and *bank+marital* while IGD suffers the same problem on *adult+race*, *compas+race*, *compas+sex*, *bank+age*, and *bank+marital*.

The baseline method Random often increases the fairness score; it however also loses the maintenance of high accuracy. This serious problem is clearly shown in four datasets *german+age, german+sex*, *bank+age*, and *bank+marital* where Random drops from 3% to 9% in its accuracy.

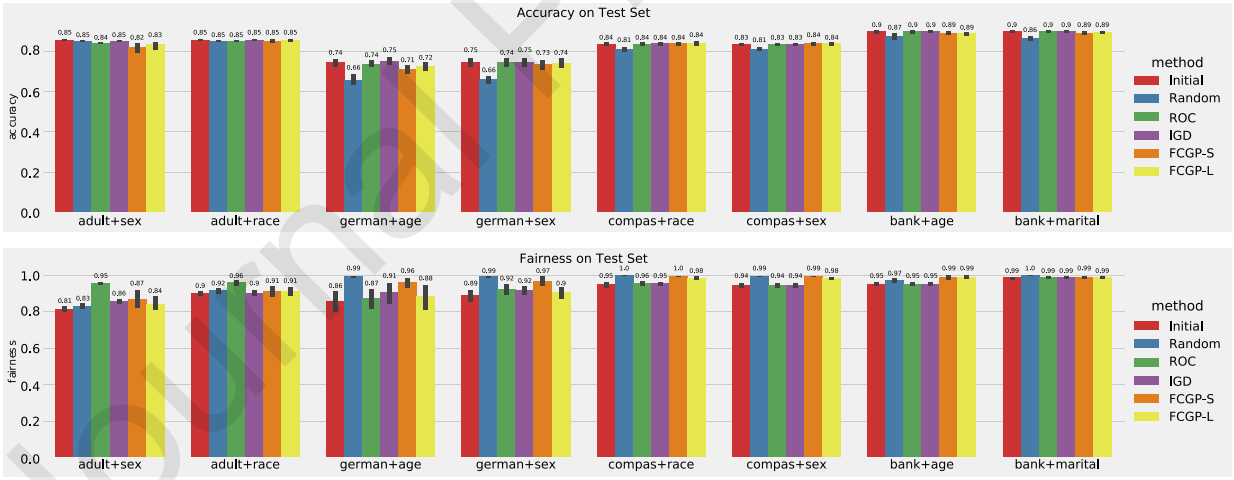


Figure A.8: Accuracy and fairness of our models **FCGP-S**, **FCGP-L**, the pre-trained classifier, and state-of-the-art baselines on four datasets (each of them has two sensitive features) with a data split of 70%-15%-15%. The performance of each method is measured on a *hold-out* test set.

Appendix A.2. Data split: 50%-25%-25%

In this setting, we use the data split 50%-25%-25% for each dataset. Namely, we use 50% of data points to train the pre-trained classifier, 25% of data points without labels to train our method and other baselines, and 25% of data

points as the *hold-out* test set to evaluate all methods.

Figure A.9 reports the accuracy and fairness scores of each method on the *hold-out* test sets. From the figure, we can observe that our method **FCGP** often improves the fairness and keeps a similar accuracy to that of the pre-trained classifier. On three datasets *compas+race*, *compas+sex*, and *bank+age*, **FCGP** can increase the fairness scores up to 1.0 whereas these three datasets are hard for two baselines ROC and IGD to improve the fairness. Neither ROC nor IGD can raise the fairness scores on these three datasets.

We observe a similar behavior for the Random method under this setting of data split, where it often increases the fairness score but decreases the accuracy score.

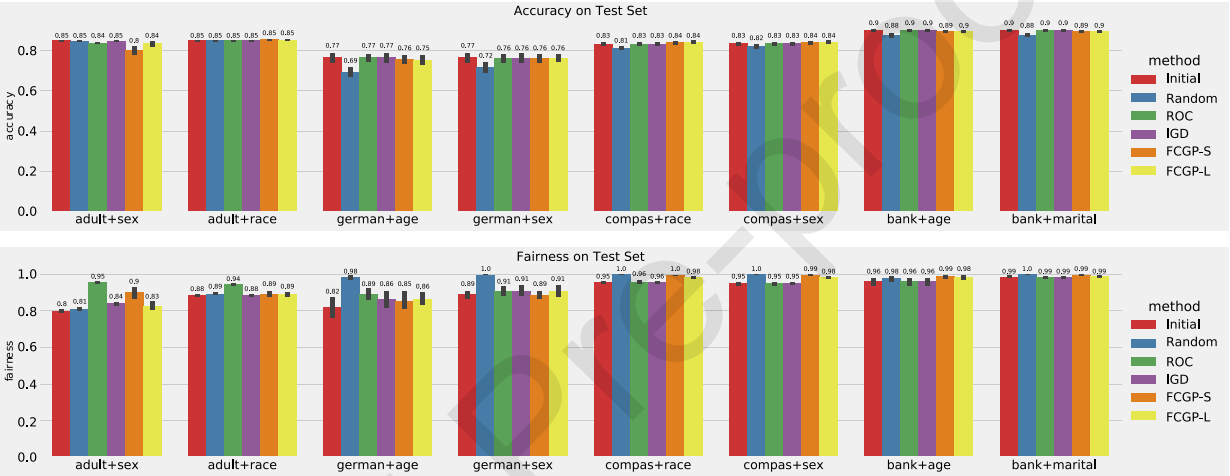


Figure A.9: Accuracy and fairness of our models **FCGP-S**, **FCGP-L**, the pre-trained classifier, and state-of-the-art baselines on four datasets (each of them has two sensitive features) with a data split of 50%-25%-25%. The performance of each method is measured on a *hold-out* test set.

In summary, the experimental results demonstrate that our method **FCGP** works effectively with different settings for data split, where it can enhance the fairness without significantly damaging the accuracy.

Fairness Improvement for Black-box Classifiers with Gaussian Process

Dang Nguyen*, Sunil Gupta, Santu Rana, Alistair Shilton, Svetha Venkatesh

Applied Artificial Intelligence Institute (A^2I^2), Deakin University, Geelong, Australia
{d.nguyen, sunil.gupta, santu.rana, alistair.shilton, svetha.venkatesh}@deakin.edu.au

Abstract

In many real-world applications, it is the fairness, not the accuracy, of a machine learning (ML) classifier that is the crucial factor. Post-processing approaches are widely considered as successful tools to improve the fairness of *black-box* ML classifiers. These aim to learn a *relabeling function* to modify initial predicted labels provided by a pre-trained "unfair" classifier, resulting in fair classification on a given test set. However, many post-processing methods require a training set with true labels to learn the relabeling function. To the best of our knowledge, there have been only two methods that learn the relabeling function without requiring the true labels of training samples. However, both of these methods require access to the predictions of the pre-trained classifier when performing on the test set, even after they learned the optimal relabeling function, and neither offers theoretical guarantees on the trade-off between accuracy loss and fairness improvement. In this paper, we propose a novel post-processing method based on Gaussian process (GP). We first train a GP with *unlabeled* samples, and use its posterior mean function to approximate the predictions of the pre-trained classifier. We then adjust the mean function (i.e. the relabeling function) to achieve two goals: (1) maximize the fairness and (2) minimize the difference between the relabeling function and the pre-trained classifier. By doing this, our method can improve fairness while maintaining high accuracy. We provide a theoretical analysis to derive an upper bound on accuracy loss for our method. We demonstrate our method on four real-world datasets, comparing with state-of-the-art baselines, to demonstrate its ability to achieve both fairness and accuracy.

Keywords: Fair classifier; Post-processing; Gaussian process; Classification; Discrimination-aware data mining; Fairness in machine learning; Black-box pre-trained classifier.

1. Introduction

Recently, machine learning (ML) models (classifiers) have had impressive successes in many real-world domains including loan approvals [1], screening of resumes [2], and decision making in the criminal justice systems [3]. However, like humans, ML models are prone to biases that make their decisions "unfair" to vulnerable sections of our communities (i.e. they make *negative* decisions toward a *unfavored* groups of people identified by a *sensitive* feature). Well-known examples include algorithms being unfairly harsh towards African-American people in granting parole [3], and a Curriculum Vitae screening algorithm unfairly rejecting resumes of capable women applicants [4].

*Corresponding author

Many solutions have been proposed in recent years to address this problem. These can be categorized into three groups: *pre-processing*, *in-processing*, and *post-processing*. The goal of pre-processing approaches is to remove the underlying discrimination from the training data so that any ML model applied to the data will be fair [5, 6, 7]. In-processing approaches modify traditional learning algorithms to remove discrimination during the training phase [8, 9, 10], resulting in a fair classifier regardless of biases in the input. Post-processing approaches take the output of any pre-trained classifier and modify it to be fair on a given test set [11, 12, 13, 14, 15]. Compared to pre-processing and in-processing approaches, post-processing approaches have two significant advantages. First, they do not require access to the original training data used to construct the pre-trained model. Second, since they only treat the pre-trained model as a *black-box* function without requiring access to the internals of learning algorithm, they are applicable to *any* ML model (this is in contrast to in-processing approaches, where the method is often applicable only to a specific ML model). An example of post-processing is illustrated in Figure 1.

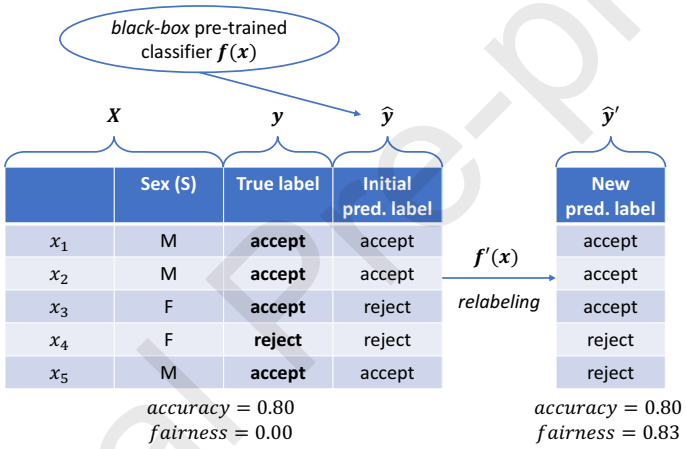


Figure 1: Illustration of post-processing approach for fairness improvement. A *black-box* pre-trained classifier $f(x)$ (e.g. a home loan approval system) achieves a high prediction performance ($accuracy = 0.80$) on a test set $D = [\boldsymbol{X}, y]$ (e.g. a list of applicants), but suffers from a high discrimination between "male" and "female" applicants ($fairness = 0.00$). Specifically, $f(x)$ rejects all applications by women and accepts all applications by men. A post-processing approach finds a *relabeling function* $f'(x)$ to change the initial predicted labels of a small set of samples such that $f'(x)$ maintains good classification performance ($accuracy = 0.80$) but improves fairness ($fairness = 0.83$). Here, the *demographic parity* measure [16] is used: $fairness = 1 - |P(\hat{y} = accept \mid Sex = M) - P(\hat{y} = accept \mid Sex = F)|$ for $f(x)$, which is maximized when the ratio of predictive positive outcomes (i.e. accept) between the two groups ("male" and "female") is balanced.

An interesting observation from Figure 1 is that we can change the initial predicted label of *any* sample in the test set to improve the fairness score since the metric is computed based on predicted labels, not true labels. Consequently, one simple approach is to randomly choose samples and change their initial predicted labels such that the number of samples receiving predicted positive outcomes ("accept") in "male" and "female" groups is balanced. Although this naive approach can improve the fairness, it can potentially reduce the accuracy. Several post-processing approaches have been proposed that take the trade-off between accuracy and fairness into account. Most require a training set

with true labels to generate the *relabeling function* [12, 13, 14], rendering them inapplicable to many real-world applications where the labels either are not given or are difficult to obtain. To the best of our knowledge, there have been only two post-processing methods [11, 15] that can learn the relabeling function based on a training set without requiring true labels. Since these two methods do not use the true labels when learning the relabeling function, they are considered to be *unsupervised learning* methods. *Reject Option based Classification* (ROC) [17, 11] attempts to find an optimal *critical region* where all samples whose initial predicted scores lie in this region are relabeled. Since the region relabeled by ROC is small, the number of relabeled samples will also be small, which ensures that the impact on accuracy caused by the relabeling is minimized. *Individual Group Debiasing* (IGD) [15] improves the fairness based on *biased scores*. It does this by finding *unfavored* samples (e.g. "female" applicants) whose initial predicted scores are different if their sensitive feature is changed (e.g. "female" $\rightarrow$ "male"), and then relabeling them. Although both ROC and IGD can improve fairness while maintaining accuracy, they are *instance-based* methods i.e. they choose samples to relabel based on the initial predicted scores of the black-box pre-trained model. Consequently, they need to query the pre-trained model when they are applied to a test set even after the optimal relabeling function has been learned. Another disadvantage of these methods is that they need to access the sensitive feature of the test set when performing relabeling. In many real-world applications, especially health-care related, sensitive data (e.g. gender of patients) may be anonymized due to privacy policies, rendering these sensitive features invisible. Consequently, neither ROC nor IGD can be applied in such cases. Moreover, neither ROC nor IGD offers a theoretical guarantee on the trade-off between accuracy loss and fairness improvement.

Our approach. To overcome the weaknesses of existing approaches, we propose a novel *Gaussian process* (GP) based method for post-processing the predictions of a pre-trained ML classifier to make them fair on the test set. Instead of choosing a set of samples to relabel, our goal is to model the whole pre-trained classifier and modify it to be fair. In particular, we treat the prediction of the pre-trained classifier as a *black-box* function and learn a relabeling function to achieve two goals: (1) minimize the distance (difference) between the pre-trained black-box classifier and the relabeling function to ensure the accuracy will not change too much and (2) maximize the fairness (i.e. minimize the discrimination) w.r.t the sensitive feature. We call our method *Fair Classifier with Gaussian Process* (**FCGP**). Since **FCGP** formalizes the problem as an optimization problem, it can learn a fairer relabeling function, also ensuring that the classification performance does not drop significantly. More importantly, after learning an optimal relabeling function, our method does not need to access either the predictions of pre-trained classifier or the sensitive feature when applied to a *hold-out* test set. This reduces expense as accessing the pre-trained model is often very expensive, if, for example, the pre-trained model is developed and deployed on the cloud by a third-party company who charges per access.

Our contributions. To summarize, we make the following contributions:

1. We propose **FCGP**, a Gaussian process-based method, for post-processing the predictions of a *black-box* pre-trained classifier to improve its fairness. Unlike existing methods, our method does not require access to either

the predictions of the pre-trained classifier or the sensitive feature after the optimal relabeling function has been learned.

2. We provide a theoretical analysis to prove that **FCGP** may reduce the accuracy but this reduction is bounded. As far as we know, our method is the first *unsupervised* post-processing algorithm that provides the upper bound for the trade-off between fairness improvement and accuracy loss.
3. We demonstrate **FCGP** on four standard datasets (each of them has two sensitive features) to show it improves fairness and maintains accuracy close to that of the pre-trained classifier. Our experimental results also show that our method is better than or comparable to state-of-the-art baselines.

The remainder of the paper is organized as follows. In Section 2, related works on fairness measures and fair algorithms are comprehensively reviewed. Our main contributions are presented in Section 3, in which an unsupervised post-processing algorithm for fairness improvement using Gaussian process is described. Section 4 provides the theoretical analyses of our proposed method, including proof that our method may reduce the accuracy of pre-trained classifier, but this reduction is bounded. Experimental results are discussed in Sections 5 while conclusions and future works are represented in Section 6.

2. Related Work

2.1. Notions of fairness

There are two main notions of fairness in decision making: *group fairness* and *individual fairness*. Group fairness uses a *sensitive* feature (e.g. Sex, Race, or Religion) to partition a population into a *favored* group (e.g. "male" applicants) and an *unfavored* group (e.g. "female" applicants), and then aims to ensure that some statistical measure be equal across two groups. Many different statistical measures have been proposed, including *disparate impact* [18], *calibration* [13], or *equalized odds* [12]. Among them, *demographic parity* [16] is one of the most widely-used. Individual fairness aims to ensure that similar individuals are treated similarly (i.e. they should receive similar classification outcomes) [19]. When checking for the individual fairness, one major challenge is to define a notion of the distance between two individuals to measure their similarity. Consequently, group fairness is more commonly used when developing a fair machine learning classifier.

2.2. Bias mitigation algorithms

Works on fair classification can be categorized into three groups: pre-processing, in-processing, and post-processing [20, 21].

Pre-processing. These approaches primarily involve massaging the training data to remove bias. Some examples include [5, 6, 7, 22, 23]. Many of these methods change the true labels and features of training data, which may have legal implications since the ML model is trained on a "fake" dataset [24].

In-processing. These approaches typically modify a specific ML algorithm to create a fair classifier. Most enforce fairness by introducing constraints in the optimization problem [8, 10, 25, 26] or adding penalties to the objective function [27, 28, 29, 30]. Some frame the problem as a two-player game [31, 32].

Post-processing. These approaches focus on relabeling the predictions of a pre-trained classifier to make them fair on a given test set [11, 12, 13, 17, 14, 15, 33]. Compared to pre-processing and in-processing approaches, post-processing approaches have two major advantages. First, they only treat the pre-trained model as a *black-box* function without requiring access to its internal learning algorithm (e.g. model parameters or derivatives), making them applicable to *any* ML model. Second, they do not require access to the original training data of the pre-trained model for learning a fair classifier.

To train the bias mitigation algorithm (i.e. learn the optimal relabeling function), most post-processing approaches require a training set with true labels. For example, [12] assumed that a *labeled* training set for learning the relabeling function was available, and changed the initial decision boundary of the pre-trained model to achieve two fairness measures: *equalized odds* and *equal opportunity*. This framework was later extended with probabilistic classifiers in [13]. The equal opportunity measure was also used in a semi-supervised method that first estimated the class condition probability using labeled training data, and then estimated the unknown decision threshold using unlabeled training data [33]. Two active learning approaches [34] were developed for fair classification, which required a training set with true labels to achieve both group fairness and intra-group fairness. Similarly, [14] also required access to a labeled training set to learn an auditing algorithm to identify which subgroups were biased, then post-processed to ensure a fair classification across all subgroups. Although these methods can improve the fairness of pre-trained classifier, their success relies on the availability of true labels for all training examples, a condition often not met in many real-world applications. As far as we know, only two methods ROC [17, 11] and IGD [15] do not need the ground truth labels of training samples when learning the optimal relabeling function. However, these two methods still require access to the predictions of the pre-trained classifier and the sensitive feature when operating on an *unseen* test set. More importantly, neither ROC nor IGD has theoretical guarantees on the trade-off between accuracy loss and fairness improvement.

Several methods have been proposed for *fair regression* [35, 36, 37]. However, *fair regression* is not our focus in this paper since we propose a novel method for *fair classification*. It is important to note that *fair classification* is totally different from *fair regression.* Since the data in regression tasks do not have labels (they only have real-valued scores), the fairness measures in classification tasks cannot be applied to regression tasks.

3. Framework

3.1. Problem definition

Let $f(x)$ be a classifier and $\mathcal{D} = \{x_i, y_i\}_{i=1}^{N}$ be a dataset. Each sample $x_i \in \mathcal{D}$ has a *sensitive* feature S_i (e.g. Sex or Race), where $S_i = 1$ is the *favored* group (e.g. Sex="male") and $S_i = 0$ is the *unfavored* group (e.g.

Sex="female"). Each $y_i \in \{0, 1\}$ is a binary *true label*, where $y_i = 1$ is the *positive* outcome (e.g. Loan="accept") and $y_i = 0$ is the *negative* outcome (e.g. Loan="reject"). Given a sample $x_i \in \mathcal{D}$, $f(x_i)$ provides a probability (called *predicted score*) that x_i belongs to label 1 (i.e. $f(x_i) = P(y_i = 1 \mid x_i)$ and $f(x_i) \in [0, 1]$). We denote the *predicted label* of x_i as $\hat{y}_i \in \{0, 1\}$, where $\hat{y}_i$ is the rounding of $f(x_i)$ (i.e. $\hat{y}_i = 1$ if $f(x_i) \geq 0.5$, otherwise $\hat{y}_i = 0$).

Definition 1. (***Accuracy***). We define *accuracy* as $P(\hat{y} = y)$, which means the percentage of samples in $\mathcal{D}$ predicted correctly by $f(x)$.

Definition 2. (***Fairness***). We define *fairness* as $1 - |P(\hat{y} = 1 \mid S = 1) - P(\hat{y} = 1 \mid S = 0)|$, which means the samples in both favored and unfavored groups should have equal probability of being assigned to a positive outcome. Here, our definition of fairness is also known as *demographic parity* [16].

Problem statement. Given a *black-box* pre-trained classifier $f(x)$ and a small *unlabeled* training set $\mathcal{D}_f = \{x_i\}_{i=1}^N$, we assume $f(x)$ achieves *high accuracy* but *low fairness* on $\mathcal{D}_f$. This assumption is reasonable since $f(x)$ is assumed to have been pre-trained on a much larger training data compared to $\mathcal{D}_f$, with a focus on high accuracy, not fairness. Our goal is to learn a *relabeling function* $f'(x)$ that modifies predicted labels of $f(x)$ such that $f'(x)$ maintains high accuracy (close to that of $f(x)$) but with improved fairness on $\mathcal{D}_f$.

Like other unsupervised learning methods [11, 15], our method uses the predicted score $f(x_i)$ of pre-trained classifier for each sample $x_i \in \mathcal{D}_f$ to generate an auxiliary training set $\mathcal{D}_f = \{x_i, f(x_i)\}_{i=1}^N$, which is used to learn the relabeling function. Then, both the pre-trained classifier $f(x)$ and the relabeling function $f'(x)$ are evaluated on a *hold-out* test set in terms of accuracy and fairness. The process of learning and evaluating the relabeling function is illustrated in Figure 2.

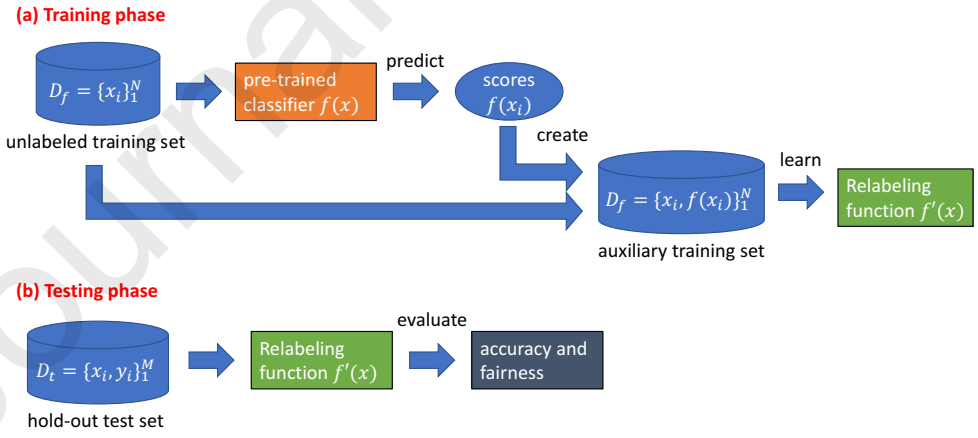


Figure 2: Training and evaluating process for the relabeling function. In the training phase (a), given an unlabeled training set $\mathcal{D}_f = \{x_i\}_1^N$, the black-box pre-trained classifier is used to predict the score $f(x_i)$ for each sample $x_i \in \mathcal{D}_f$. Then, each sample x_i along with its predicted score is used to generate an auxiliary training set to learn the relabeling function $f'(x)$. In the testing phase (b), the learned relabeling function $f'(x)$ is evaluated on a hold-out test set $\mathcal{D}_t$ to compute both accuracy and fairness.

3.2. Proposed method FCGP

To improve the fairness score of $f(x)$, we can simply select random samples in the unfavored group (i.e. $S = 0$) and change their predicted labels from negative outcome (i.e $\hat{y} = 0$) to positive outcome (i.e. $\hat{y} = 1$), which results in the balance between $P(\hat{y} = 1 \mid S = 1)$ and $P(\hat{y} = 1 \mid S = 0)$. However, we expect this simple approach will also reduce the accuracy significantly as there is no mechanism to maintain the accuracy of $f(x)$. Alternative solutions are to carefully select a small set of samples to relabel e.g. ROC [11] chooses *uncertain* samples whose initial predicted scores are close to 0.5 while IGD [15] chooses *biased* samples whose initial predicted scores are different if their sensitive values are changed. However, as discussed in Section 1, both ROC and IGD are *instance-based* methods, making them reliant on the pre-trained classifier even after learning an optimal relabeling function.

Our strategy to solve the problem as discussed in Section 3.1 differs from ROC and IGD. Instead of selecting a subset of training samples to relabel, we learn a relabeling function that approximates the prediction of the pre-trained classifier on the whole space, and adjust this relabeling function to be fair on the training set. Thus, after learning an optimal relabeling function we can use it as a fair classifier on a new test set without further accesses to the pre-trained classifier (i.e. the relabeling function is used *independently* of the pre-trained classifier). We formalize our proposal as an optimization problem as follows:

$$\underset{f'(x)}{\text{argmin}} \underbrace{\frac{1}{N}\sum_{i=1}^{N}|f'(x_i) - f(x_i)|}_{\text{Term-1}} + \underbrace{|P(\hat{y}' = 1 \mid S = 1) - P(\hat{y}' = 1 \mid S = 0)|}_{\text{Term-2}} \tag{1}$$

where $f'(x)$ is the relabeling function. Similar to $f(x)$, given a sample $x_i \in \mathcal{D}_f$, $f'(x_i)$ provides the *predicted score* of x_i and $\hat{y}'_i \in \{0, 1\}$ is the *predicted label* of x_i (i.e. $\hat{y}'_i$ is the rounding of $f'(x_i)$).

Our objective function in Equation (1) has two terms. In **Term-1** (*diff*), we minimize the *difference* between the relabeling function $f'(x)$ and the pre-trained classifier $f(x)$ so that $f'(x)$ can maintain the high accuracy. In **Term-2** (*disc*), we minimize the *discrimination* (i.e. maximize the fairness) score of $f'(x)$ w.r.t the sensitive attribute S.

To find the optimal $f'(x)$, we propose a novel method based on a Gaussian process (GP) [38, 39, 40]. Our method, called *Fair Classifier with Gaussian Process* (**FCGP**), has two main steps: (1) modeling $f(x)$ using a GP and (2) finding the optimal relabeling function $f'(x)$.

3.2.1. Modeling the black-box pre-trained classifier $f(x)$ using GP

Since $f(x)$ is a *black-box* function, we do not know its closed form and can only observe its predicted score $f(x_i) \in [0, 1]$ for each sample $x_i \in \mathcal{D}_f$. To find a relabeling function $f'(x)$ as close as $f(x)$, we first need to model $f(x)$.

To model $f(x)$, we choose to use a GP model, which is one of the most popular methods for modeling black-box functions [38, 39]. We assume that $f(x)$ is a continuous function drawn from a GP i.e. $f(x) \sim \text{GP}(\mu(x), k(x, x'))$,

where $\mu(x)$ is a prior *mean function* (this can be safely assumed to be a zero function) and $k(x, x')$ is a *covariance function* that models the covariance between any two function values $f(x)$ and $f(x')$. A common covariance function is the *squared exponential* kernel [40], $k(x, x') = \sigma_k^2 \exp(-\frac{1}{2l^2}||x - x'||_2^2)$, where σ_k^2 is a parameter dictating the uncertainty in $f(x)$, and l is a length-scale parameter indicating the smoothness of $f(x)$.

To update the belief about $f(x)$, we use the training set $\mathcal{D}_f = \{x_i, f(x_i)\}_{i=1}^N$ (i.e. observations of $f(x)$) to compute the posterior distribution of $f(x)$. The *predictive distribution* for $f(x)$ conditioned on $\mathcal{D}_f$ is also a Gaussian distribution i.e. $f(x) \mid \mathcal{D}_f \sim \mathcal{N}(\mu_f(x), \sigma_f^2(x))$. Following [38, 39], we compute the posterior *mean function* $\mu_f(x)$ and the *variance function* $\sigma_f^2(x)$ as follows:

$$\mu_f(x) = \boldsymbol{k}^\mathsf{T} K^{-1} f_{1:N} \tag{2}$$

$$\sigma_f^2(x) = k(x, x) - \boldsymbol{k}^\mathsf{T} K^{-1} \boldsymbol{k} \tag{3}$$

where $f_{1:N} = [f(x_1), f(x_2), ..., f(x_N)]$ is a vector of predicted scores of the pre-trained classifier, K is a covariance matrix of size $N \times N$ with (i, j)-th element defined as $k(x_i, x_j)$ ($x_i, x_j \in \mathcal{D}_f$), and $\boldsymbol{k} = [k(x_i, x)]_{\forall x_i \in \mathcal{D}_f}$ is a vector containing the corvariance between a new point x and all observed points x_i in the training set $\mathcal{D}_f$.

Since the mean function $\mu_f(x)$ in Equation (2) can approximate the predicted score of $f(x)$ at *any* point x, we use it as the relabeling function $f'(x)$[2].

3.2.2. Finding the optimal relabeling function $f'(x)$

After obtaining a GP mean function $\mu_f(x)$ to approximate the pre-trained classifier $f(x)$ in Section 3.2.1, the next question is how we can adjust $\mu_f(x)$ to obtain a fair classification on $\mathcal{D}_f$. From Figure 3, we see that, given a set of samples in the training set $\mathcal{D}_f = \{x_i, f(x_i)\}_{i=1}^N$, the predicted scores of $f(x)$ and $\mu_f(x)$ are exactly the same at each sample. This is because the mean function $\mu_f(x)$ computed in Equation (2) treats the predicted scores of $f(x)$ as *noise-free* observations. Although this behavior achieves the goal of Term-1 in Equation (1) (i.e. minimizing the difference between the relabeling function and the pre-trained classifier down to 0), it cannot achieve the goal of Term-2. In other words, we cannot use $\mu_f(x)$ to change the predicted labels given by $f(x)$. This leads to the problem that $\mu_f(x)$ can maintain the accuracy of $f(x)$ but cannot improve the fairness score.

[2]For the remaining of paper, we use the mean function $\mu_f(x)$ and the relabeling function $f'(x)$ interchangeably.

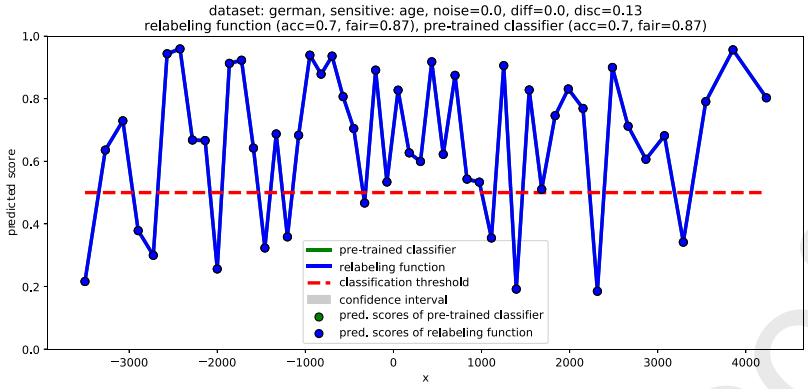


Figure 3: Modeling the black-box pre-trained classifier $f(x)$ (green line) via a GP mean function $\mu_f(x)$ (blue line). Since the evaluation of $f(x)$ is *noise-free*, the mean function $\mu_f(x)$ has the same predicted scores as $f(x)$ at all samples in the training set $\mathcal{D}_f$. Although the difference between two functions is minimized ($diff = \frac{1}{N}\sum_{i=1}^{N}|\mu_f(x_i) - f(x_i)| = 0$) i.e. $\mu_f(x)$ maintains the same accuracy ($acc = 0.7$) as $f(x)$, the fairness score ($fair = 0.87$) of $\mu_f(x)$ is not improved.

To solve this problem, we assume that each predicted score given by $f(x_i)$ is perturbed by noise $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$. Thus, when fitting a GP to the set of noisy samples $\mathcal{D}_f = \{x_i, f(x_i) + \epsilon_i\}_{i=1}^{N}$, we have an in-built error level for each predicted score of $\mu_f(x)$. In other words, the mean function $\mu_f(x)$ predicts a different score from $f(x)$ at each sample in the training set $\mathcal{D}_f$, depending on the noise variance σ^2. This allows us to optimize σ^2 to generate $\mu_f(x)$ such that it achieves a better fairness score while maintaining good accuracy.

To compute the mean function $\mu_f(x)$ with the noise variance σ^2, we replace the covariance matrix K by the following matrix:

$$K = \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_N) \\ \dots & \dots & \dots \\ k(x_N, x_1) & \dots & k(x_N, x_N) \end{bmatrix} + \sigma^2 I,$$

where I is an identity matrix with the same dimension as K.

With the new covariance matrix K and noise variance σ^2, the posterior *mean function* $\mu_f(x)$ and the *variance function* $\sigma_f^2(x)$ are computed as:

$$\mu_f(x) = \boldsymbol{k}^{\mathsf{T}}[K + \sigma^2 I]^{-1} f_{1:N} \tag{4}$$

$$\sigma_f^2(x) = k(x, x) - \boldsymbol{k}^{\mathsf{T}}[K + \sigma^2 I]^{-1}\boldsymbol{k} \tag{5}$$

Our idea of leveraging a mean function $\mu_f(x)$ with noise variance σ^2 to improve the fairness score is illustrated in Figure 4.

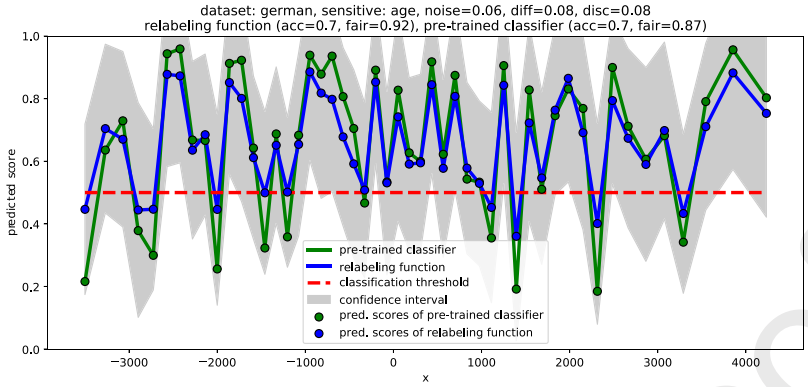


Figure 4: With an assumption that each predicted score of the pre-trained classifier $f(x)$ (green line) has a noise $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ ($\sigma^2 = 0.06$), the mean function $\mu_f(x)$ (blue line) provides different predicted scores (blue dots) from those of $f(x)$ (green dots) for samples in the training set $\mathcal{D}_f$. Some samples are relabeled when the predicted scores of $\mu_f(x)$ are equal or greater than the classification threshold $\alpha = 0.5$ (red dot line). This results in the difference between two functions being increased ($diff = 0.08$) but the fairness score is also improved from 0.87 to 0.92.

As shown in Figure 4, with different noise levels set by σ^2 we obtain different relabeling functions $f'(x)$. Our next step is to optimize σ^2 to obtain the optimal $f'(x)$ that minimizes the objective function in Equation (1) (i.e. minimizing both difference and discrimination). To optimize σ^2, we use grid search as in [11, 15]. Alternative solutions can be a local optimizer with multi-starts or Thompson sampling in Bayesian optimization [41].

Our **FCGP** is summarized in Algorithm 1. **FCGP** receives a training set (without true labels) $\mathcal{D}_f$ as input. It uses a grid search to find the optimal noise variance σ^2_*, and then computes the optimal relabeling function $\mu^*_f(x)$. Since **FCGP** returns the optimal relabeling function $\mu^*_f(x)$ as output, $\mu^*_f(x)$ is used as a *fair classifier* to relabel samples in the training set $\mathcal{D}_f$ or new samples in an *unseen* test set.

3.3. A variant of FCGP focusing on fairness improvement

As shown in Equation (1), our objective function tries to balance accuracy (Term-1) and fairness (Term-2) i.e. it finds a relabeling function $f'(x)$ that improves the fairness score but remains close to the pre-trained classifier $f(x)$. This objective function is intuitive and works well in most cases where it can improve the fairness. However, there is one case where the relabeling function $f'(x)$ may not improve the fairness score. If the samples in the training set are dense i.e. they are nearby each other, the objective function tends to find $f'(x)$ that is very similar to $f(x)$, which keeps both accuracy and fairness unchanged. This is because whenever $f'(x)$ changes the predicted score of an sample x_i, it also changes the predicted scores of x_i's neighbor, leading to the difference between two functions increasing significantly.

Since our goal is to improve the fairness of pre-trained classifier $f(x)$, we propose a new objective function as

Input: $\mathcal{D}_f = \{x_i, f(x_i)\}_{i=1}^N$: training set without true labels, T: # of iterations

begin

define a grid of noise variances $[\sigma_1^2, \sigma_2^2, ..., \sigma_T^2]$;

fit a GP (i.e. $P(f(x) \mid \mathcal{D}_f)$) using $\mathcal{D}_f$;

for $t = 1, 2, ..., T$ **do**

compute $diff = \frac{1}{N}\sum_{i=1}^N |\mu_f(x_i) - f(x_i)|$, with $\mu_f(x_i) = \boldsymbol{k}^\mathsf{T}[K + \sigma_t^2 I]^{-1} f_{1:N}$ Eq. (4);

compute $\hat{y}' = \mathbb{I}_{\forall x \in \mathcal{D}_f}(\mu_f(x) \geq 0.5)$;

compute $disc = |P(\hat{y}' = 1 \mid S = 1) - P(\hat{y}' = 1 \mid S = 0)|$;

compute $score_{\sigma_t^2} = diff + disc$ Eq. (1);

end

choose optimal noise variance $\sigma_*^2 = \arg\min_{\sigma_t^2} score_{\sigma_t^2}$;

compute optimal relabeling function $\mu_f^*(x) = \boldsymbol{k}^\mathsf{T}[K + \sigma_*^2 I]^{-1} f_{1:N}, \forall x \in \mathcal{D}_f$;

end

Algorithm 1: The proposed **FCGP** algorithm.

follows:

$$\underset{f'(x)}{\text{argmin}} \underbrace{\frac{1}{N}\sum_{i=1}^N |\hat{y}'_{x_i} - \hat{y}_{x_i}| + \mathbb{I}(\hat{y}' = \hat{y})}_{\text{Term-1}} + \underbrace{|P(\hat{y}' = 1 \mid S = 1) - P(\hat{y}' = 1 \mid S = 0)|}_{\text{Term-2}} \tag{6}$$

where $f'(x)$ is a relabeling function. Given a sample x_i, $f'(x_i)$ returns a predicted score for x_i and $\hat{y}'_{x_i}$ is a predicted label of x_i (i.e. $\hat{y}'_{x_i} = \mathbb{I}(f'(x_i) \geq 0.5)$).

In Term-1, instead of using the predicted scores, we compute the distance between two functions $f'(x)$ and $f(x)$ using their predicted labels. By doing this, we solve the above problem since we do not care about the difference between $f'(x)$ and $f(x)$ at samples whose labels are unchanged. As long as $f'(x)$ relabels a sample x_i, the difference between $f'(x)$ and $f(x)$ at x_i's neighbor is no longer important. We also add a penalty term $\mathbb{I}(\hat{y}' = \hat{y})$ to Term-1 to ignore relabeling functions that are identical to $f(x)$ (i.e. they do not improve the fairness). When $f'(x) \equiv f(x)$, $\frac{1}{N}\sum_{i=1}^N |\hat{y}'_{x_i} - \hat{y}_{x_i}| = 0$ whereas $\mathbb{I}(\hat{y}' = \hat{y}) = 1$, maximizing Term-1, which is in conflict with our objective to minimize Term-1.

Term-1 consists of $\frac{1}{N}\sum_i |\hat{y}'_{x_i} - \hat{y}_{x_i}|$ and the penalty term $\mathbb{I}(\hat{y}' = \hat{y})$. We minimize $\frac{1}{N}\sum_i |\hat{y}'_{x_i} - \hat{y}_{x_i}|$ to make the accuracy of relabeling function close to that of the pre-trained classifier. We add the penalty term $\mathbb{I}(\hat{y}' = \hat{y})$ to ignore relabeling functions that provide the same predictions as those of the pre-trained classifier because they do not improve the fairness score. For example, if $\hat{y}' = [1, 0, 1, 1]$ and $\hat{y} = [1, 0, 1, 1]$, then there is no different prediction between $\hat{y}'$ and $\hat{y}$, and $\sum_i |\hat{y}'_{x_i} - \hat{y}_{x_i}| = 0$. In this case, $\mathbb{I}(\hat{y}' = \hat{y})$ becomes 1, making the whole Term-1

$= \frac{1}{N}\sum_i |\hat{y}'_{x_i} - \hat{y}_{x_i}| + \mathbb{I}(\hat{y}' = \hat{y}) = 1$. Since our goal is to minimize both Term-1 and Term-2, and the value range of Term-1 is $[0, 1]$, Term-1 $= 1$ (the maximum value) will conflict with our objective, so $\hat{y}' = [1, 0, 1, 1]$ will not be selected by our method.

4. Theoretical Analysis

This section provides a theoretical analysis of our method. Assuming that the *black-box* pre-trained classifier is trained with accurate labels, fairness and accuracy are two conflicting goals, and so increasing the classifier's fairness may lead to some increase in its error rate. In this section, we provide an upper bound on the additional classification error incurred due to our relabeling process that aims to maximize fairness. A sketch of our proof is outlined below.

Since our Gaussian process is a flexible non-parametric regression model that we train using the output of the black-box classifier, without noise its lowest training error rate will be the same as the generalization error rate of the black-box classifier. When treating black-box outputs as noisy observations (which is required in our case to improve the fairness), the Gaussian process has an asymptotic error rate that differs from that of the black-box classifier by at most the noise variance. We utilize a generalization error bound for Gaussian process to bound its error rate in terms of the number of samples in the training set. An interesting property of this bound is that the additional error rate of our Gaussian process asymptotically reduces to the noise variance. Details are as follows.

Let us use $\mathcal{D}_f = \{x_i, f(x_i)\}_{i=1}^N$ to denote a training set sampled from a data distribution P on $X \times F$, where $F \in [0, 1]$ are the predicted scores of X. In our case, $f(x_i)$ is the predicted score of the black-box pre-trained classifier for an instance $x_i \in \mathcal{D}_f$. Then the goal of any non-parametric least squares regression is to find a function $g : X \to \mathbb{R}$ by minimizing a risk functional of the form $\mathcal{R}_{L,P}(g) = \mathbb{E}_{X,F\sim P}\left[L(f(x), g(x))\right] = \int_{X\times F} L(f(x), g(x))dP(x, f(x))$, where L is the least square loss, i.e. $L(a, b) = (a-b)^2$. Assuming a function class such as reproducing kernel Hilbert space (RKHS), the optimal risk is given as:

$$\mathcal{R}^*_{L,P} = \inf\{\mathcal{R}_{L,P}(g) \mid g : X \to \mathbb{R}\}$$

There are many ways to solve the non-parametric least squares regression problem (see [42]). In this work, we use a Gaussian process that is a kernel-based method. Gaussian process regression can be shown to estimate a function through the following formulation:

$$g_{\mathcal{D}_f,\lambda} = \underset{g\in H}{\operatorname{argmin}} \left(\lambda||g||^2_H + \mathcal{R}_{L,\mathcal{D}_f}(g)\right) \tag{7}$$

A solution for the above optimization problem in Equation (7) has the form $g_{\mathcal{D}_f,\lambda}(x) = \boldsymbol{k}^\mathsf{T}[K + \lambda I]^{-1} f_{1:N}$, where $\mathbf{k} = [k(x, x_1), \ldots, k(x, x_N)]$ and $f_{1:N} = [f(x_1), \ldots, f(x_N)]$. In our case, since we fit a Gaussian process with noise variance σ^2, we have $\lambda = \frac{\sigma^2}{N}$. We adapt a generalization bound from [43] to our problem and write as:

$$\mathcal{R}_{L,P}(g_{\mathcal{D}_f,\lambda}) - \mathcal{R}^*_{L,P} \leq 9A(\frac{\sigma^2}{N}) + c\frac{a^p\omega}{\sigma^{2p}N^{1-p}}, \tag{8}$$

where $A(r) = cr^{\beta}$ with $\beta \in (0,1)$ and c being a constant only depending on p and $C \geq 1$ such that $||g||_{\infty} \leq C||g||_H^p \cdot ||g||_{L_2(P_X)}^{1-p}$; and $a \geq 16, \omega > 0, p \in (0,1)$ are constants.

Next, we define the risk of black-box pre-trained classifier $f(x)$ as $\mathcal{R}_{L,P'}(f) = \mathbb{E}_{X,Y\sim P'}\left[L(y, f(x))\right]$ where P' is the distribution of the data $X \times Y$, where Y are the actual labels of X. We note that $\mathcal{R}^*_{L,P} = \mathcal{R}_{L,P'}(f) + \sigma^2$. This is because Gaussian process is a flexible non-parametric regression model that can approximate any function with arbitrary accuracy given a sufficiently large number of samples [44]. Since $\mathcal{R}^*_{L,P}$ is the best possible risk achievable using a Gaussian process assuming access to the true data distribution, fitting a Gaussian process with infinitely many *noise-free* samples will make $\mathcal{R}^*_{L,P}$ equal the risk of black-box pre-trained classifier $\mathcal{R}_{L,P'}(f)$. However, since samples are noisy with noise variance σ^2, we have $\mathcal{R}^*_{L,P}$ exceeding $\mathcal{R}_{L,P'}(f)$ by at most σ^2. Combining this result with Equation (8), we have:

$$\mathcal{R}_{L,P}(g_{\mathcal{D}_f,\lambda}) - \mathcal{R}_{L,P'}(f) \leq \sigma^2 + 9A(\frac{\sigma^2}{N}) + c\frac{a^p\omega}{\sigma^{2p}N^{1-p}}$$

We can see from the above bound that the difference between the risk of our Gaussian process model (i.e. our relabeling function) and that of the pre-trained classifier decreases with the number of samples N in the training set, and asymptotically reduces to σ^2, which has been used to achieve the maximum fairness. This can also be interpreted as a trade-off between the accuracy loss and the fairness improvement.

5. Experiments

In this section, we conduct extensive experiments on four real-world datasets to evaluate the performance (accuracy and fairness) of our proposed method **FCGP**, compared with three state-of-the-art baselines. Our experiments have two settings. (1) We use a training set *without true labels* to learn an optimal relabeling function and then evaluate its performance on the same training set. (2) After learning the optimal relabeling function from the training set, we evaluate its performance on a *hold-out* test set.

5.1. Datasets

We use four standard real-world datasets *Adult* (an income dataset based on a 1994 U.S. Census data), *German* (a credit scoring dataset), *Compas* (a prison recidivism dataset for violent crime), and *Bank* (a direct marketing campaign dataset for term deposit subscription). Since each dataset has two sensitive features, there are eight evaluation datasets in total. These datasets are commonly used to evaluate the performance of a fair classification algorithm [12, 11, 9, 16]. Their characteristics are summarized in Table 1.

5.2. Baselines

We compare our **FCGP** with three state-of-the-art baselines Random, ROC, and IGD. As far as we know, only these three methods can learn a relabeling function based on a training set without requiring true labels.

Table 1: Characteristics of four benchmark datasets. We denote the sensitive feature as S (where $S = 1$ is *favored* group and $S = 0$ is *unfavored* group) and the class feature as y (where $y = 1$ is *positive* outcome and $y = 0$ is *negative* outcome).

Dataset	**# samples**	**# features**	S	$S = 1$	$S = 0$	y	$y = 1$	$y = 0$
Adult	30,162	13	Sex	"male"	"female"	Income	">50K"	"<50K"
			Race	"white"	"non-white"			
German	1,000	20	Age	"adult"	"youth"	Credit	"good"	"bad"
			Sex	"male"	"female"			
Compas	4,010	10	Race	"Caucasian"	"non-Caucasian"	Rearrested	"no"	"yes"
			Sex	"male"	"female"			
Bank	4,521	14	Age	"adult"	"youth"	Subscribed	"yes"	"no"
			Marital	"single"	"not-single"			

1. Random method: at each iteration, this method selects randomly one sample from the training set and relabels it such that the fairness is improved. For example, at iteration t, it draws a sample x. To increase the fairness (i.e. reducing the difference between $P(\hat{y} = 1 \mid S = 1)$ and $P(\hat{y} = 1 \mid S = 0)$), it assigns label $\hat{y}_x = 1$ to x if x has $S_x = 0$. Otherwise, it assigns label $\hat{y}_x = 0$ to x.
2. ROC method [17]: this method relabels *uncertain* samples whose initial predicted scores are in a *critical region* $0.5 - \theta \leq f(x) \leq 0.5 + \theta$, where θ is a *region margin*. Given a uncertain sample x, it is assigned a label $\hat{y}_x = 0$ if it has $S_x = 1$ and is assigned a label $\hat{y}_x = 1$ if it has $S_x = 0$. Other samples whose initial predicted scores are outside the critical region keep the same predicted labels as the initial ones. θ is optimized to achieve the best fairness but as small as possible to minimize the number of relabeled samples (i.e. minimizing the accuracy loss). We implement ROC based on the source code[3] provided in IBM AI Fairness 360 Toolkit [45].
3. IGD method [15]: this method relabels *biased* samples whose bias scores are greater than a *bias threshold* τ. A *bias score* of a sample x is computed as $b_x = f(x \mid S_x = 1) - f(x \mid S_x = 0)$ (i.e. the difference in its initial predicted score if its sensitive feature changes from unfavored to favored group). τ is optimized to achieve the best fairness but as large as possible to minimize the number of relabeled samples (i.e. minimizing the accuracy loss). We implement IGD based on Algorithm 1 in its paper [15].

Our method **FCGP** has two models that differ in the computation of the distance between the relabeling function and the pre-trained classifier. **FCGP-S** model uses the objective function in Equation (1) with predicted *scores* while **FCGP-L** model uses the objective function in Equation (6) with predicted *labels*.

[3] `https://github.com/IBM/AIF360`

5.3. *Implementation details*

We implement our method **FCGP** and all baselines using Python. For a fair comparison, we use a grid search with the same budget of 50 (i.e. the number of iterations) to optimize each model's hyper-parameter, namely the region margin θ in ROC, the bias threshold τ in IGD, and the noise variance σ^2 in our **FCGP**. The standard search range for θ and σ^2 is $[0.0, 0.5]$ while that for τ is $[0.0, 1.0]$. Since the Random method does not have any hyper-parameter, we use the budget as the number of random samples to relabel. Each dataset is randomly split into 90% for training the black-box pre-trained classifier, 5% for the *unlabeled* training set to learn the relabeling function, and 5% for the *hold-out* test set. We train the black-box pre-trained classifier as a neural network with one hidden layer and 32 hidden units. Note that other ML models can be also used for the pre-trained classifier since post-processing approaches are not restricted to any specific classifier. We use the training set without its true labels to learn the optimal relabeling function. We repeat the classification process on each training set and hold-out test set 10 times and report the average accuracy and fairness along with their standard deviations.

5.4. *Performance comparison on training set*

This experiment illustrates how well our models **FCGP-S** and **FCGP-L** perform on a training set. Although this training set is also used to learn the optimal relabeling function, we do not use its true labels during the learning process. Thus, the performance comparison is still reasonable and fair for all methods.

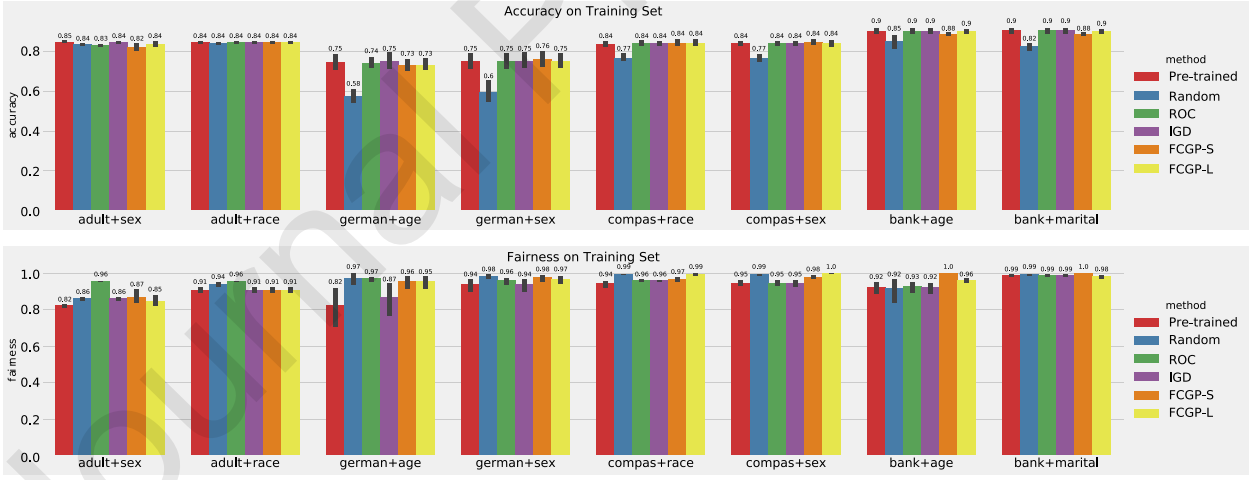


Figure 5: Accuracy and fairness of our models **FCGP-S**, **FCGP-L**, the pre-trained classifier, and state-of-the-art baselines on four datasets (each of them has two sensitive features). The performance of each method is measured on a training set. Although this training set is also used for learning the optimal relabeling function, we do not use its true labels during the learning process. Thus, the performance comparison (e.g. accuracy comparison) is still reasonable and fair for all methods.

From the results shown in Figure 5, we can see that both our models **FCGP-S** and **FCGP-L** improve fairness while maintaining a high accuracy on all datasets except the dataset *adult+race* (i.e. dataset *Adult* with sensitive

feature Race). For example, on *german+age*, **FCGP-S** keeps a similar accuracy as the pre-trained classifier but it improves the fairness significantly (0.82 vs. 0.96).

Random often increases the fairness scores; it however also drops accuracy deeply. For example, on *german+age*, Random increases the fairness up to 0.97 but its accuracy drops 17% (from 0.75 down to 0.58).

ROC works well on most datasets, where it can maintain a high accuracy but improves the fairness. Compared to ROC, our method performs better on four datasets *german+sex*, *compas+race*, *compas+sex*, and *bank+age*.

IGD always keeps the same accuracy as that of the pre-trained classifier. However, it cannot improve the fairness on several datasets. For example, on *adult+race*, *german+sex*, *compas+sex*, *bank+age*, and *bank+marital*, the fairness scores of IGD are unchanged, compared to those of the pre-trained classifier. This problem can be explained by the fact that IGD focuses on relabeling only unfavored samples (e.g. "female" applicants). In contrast, other methods like Random, ROC, and our method can relabel both unfavored and favored samples. Compared to IGD, our model **FCGP-S** is better on seven datasets, namely (*adult+sex*, *german+age*, *german+sex*, *compas+race*, *compas+sex*, *bank+age*, and *bank+marital*) while it is comparable on *adult+race*.

In most cases, our two models behave similarly. However, **FCGP-L** outperforms **FCGP-S** on two datasets *compas+race* and *compas+sex*. After a further investigation, we find that the data points in these two datasets are very close to each other, as shown in Figure 6. Consequently, on such kind of dataset, **FCGP-L** is better than **FCGP-S** in terms of fairness improvement, as explained in Section 3.3.

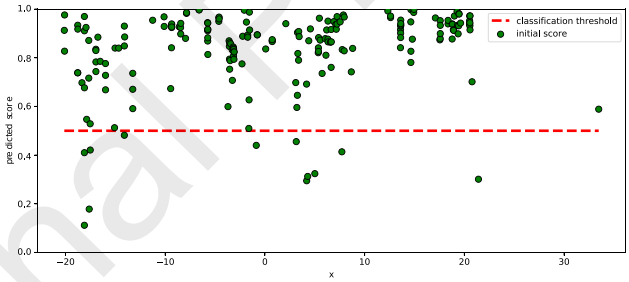


Figure 6: A training set of the dataset *compas+sex* showing its data points are very close to each other. On such kind of dataset, our model **FCGP-L** performs better than **FCGP-S** in terms of fairness improvement, as explained in Section 3.3. To visualize a high-dimension data point, we map it into 1-dimension using t-SNE [46].

5.5. Performance comparison on hold-out test set

After learning the optimal relabeling function in Section 5.4, we evaluate its performance on a *hold-out* test set. Note that different from our method **FCGP** that learns a *real* optimal relabeling function ($\mu_f^*(x)$ in Algorithm 1), both ROC and IGD learn optimal *thresholds* to relabel new unseen samples. In particularly, ROC learns the optimal region margin θ^*, and relabels a test sample x if its initial predicted score belongs to $0.5 - \theta^* \leq f(x) \leq 0.5 + \theta^*$. The new predicted label for x is assigned based on its sensitive feature, as explained in Section 5.2. Similarly, IGD learns the

optimal bias threshold τ^*, and relabels a unfavored test sample x if its bias score $b_x = f(x \mid S_x = 1) - f(x \mid S_x = 0) > \tau^*$.

We see that whenever ROC and IGD perform a relabeling process on a test set, they require access to the pre-trained classifier to obtain its predicted scores on the test set. In many real-word scenarios where the pre-trained classifier is developed and deployed on cloud by a third-party company, accessing the cloud model often costs a certain amount of money. Hence, minimizing the number of accesses to the pre-trained classifier is desirable. Moreover, both ROC and IGD require access to the sensitive feature of test set to perform relabeling. In many real-world applications, especially health-care applications, sensitive features (e.g. patient gender or sexual orientation) are often anonymized due to privacy policies, making the sensitive features invisible. Consequently, neither ROC nor IGD can be applied to these types of sensitive and crucial data.

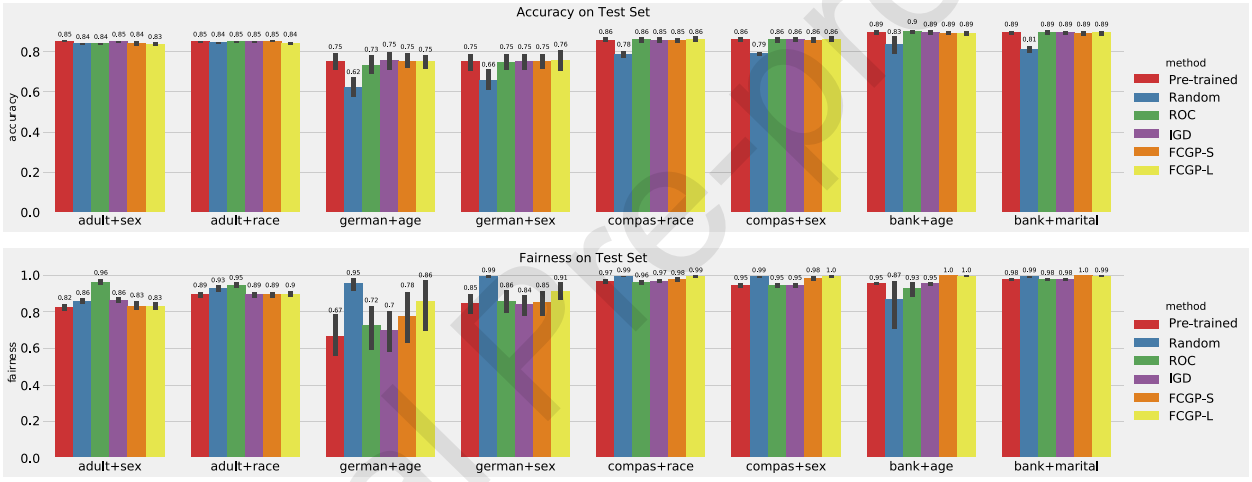


Figure 7: Accuracy and fairness of our models **FCGP-S**, **FCGP-L**, the pre-trained classifier, and state-of-the-art baselines on four datasets (each of them has two sensitive features). The performance of each method is measured on a *hold-out* test set.

Figure 7 reports the accuracy and fairness of all methods on *hold-out* test sets, where we can observe similar results as those in training sets. All methods except Random can maintain a high accuracy, as close as that of the pre-trained classifier. Our models **FCGP-S** and **FCGP-L** often improve the fairness, obtaining very high scores on *german+age*, *german+sex*, *compas+race*, *compas+sex*, and *bank+age*. ROC is often better or comparable with IGD. Our model **FCGP-L** significantly outperforms ROC on five datasets *german+age*, *german+sex*, *compas+race*, *compas+sex*, and *bank+age*. Again, Random often increases the fairness score but fails to maintain high accuracy. This problem is clearly shown in two datasets *german+age* and *german+sex*, where Random significantly drops its accuracy.

We have conducted more experiments with various settings for data split. These results are presented in Appendix A.

6. Conclusion

We have presented a novel *unsupervised* post-processing approach – **FCGP** to improve the fairness of *black-box* pre-trained classifiers. Our method uses the posterior mean function of a Gaussian process to generate a relabeling function, and optimizes it by adjusting the noise variance. Unlike existing unsupervised post-processing methods, our method **FCGP** learns a *real* optimal relabeling function that can be used to classify *unseen* test sets without further accesses to the pre-trained classifier. Moreover, our method does not need to know the sensitive feature on the test set. These two advantages of our method over baselines make it applicable to a wide range of real-world applications, especially in domains where the data are anonymized. We also theoretically analyze the trade-off between accuracy loss and fairness improvement of our method, and prove that our accuracy loss is bounded by the number of samples in the training set. We demonstrate the efficacy of **FCGP** on four standard real-world datasets, each with two sensitive features. The empirical results show that **FCGP** is better or comparable with state-of-the art baselines, improving the fairness of the original black-box pre-trained classifier in most cases.

Discussion: Most existing state-of-the-art methods for fair classification including the baselines ROC and IGD have been developed for binary classification problems [16, 20]. This is because many fairness measures require the concept of "positive outcome" and "negative outcome" in the labels i.e. binary decisions. Aligning with the fairness literature, our method was proposed for binary classification problems.

To adapt to multi-class classification problems, we need to use a relevant fairness measure. One possible option is the *accuracy parity* [16], defined as $1 - |P(\hat{y} = y \mid S = 1) - P(\hat{y} = y \mid S = 0)|$. The accuracy parity encourages the overall accuracy of favored and unfavored groups as close as possible. Our method is straightforwardly applicable to this new fairness measure. By simply replacing the demographic parity in Term-2 of our objective functions (see Equations (1) and (6)) with the accuracy parity, our method can be adapted to multi-class classification tasks.

Future works. In the future, we will investigate the performance of **FCGP** when applied to improving *individual fairness* that requires two samples which are similar with respect to a given measure receive similar classification outcomes [19]. The requirement of individual fairness is naturally incorporated in our framework since one of properties of a Gaussian process is to predict similar labels for similar (nearby) data points.

Acknowledgment

This research was fully supported by the Australian Government through the Australian Research Council's Discovery Projects funding scheme (project DP210102798). The views expressed herein are those of the authors and are not necessarily those of the Australian Government or Australian Research Council.

References

[1] A. Mukerjee, R. Biswas, K. Deb, A. Mathur, Multi–objective evolutionary algorithms for the risk–return trade–off in bank loan management, International Transactions in Operational Research 9 (5) (2002) 583–597.

[2] L. Cohen, Z. Lipton, Y. Mansour, Efficient candidate screening under multiple tests and implications for fairness, arXiv preprint arXiv:1905.11361.

[3] J. Dressel, H. Farid, The accuracy, fairness, and limits of predicting recidivism, Science Advances 4 (1) (2018) eaao5580.

[4] P. Lahoti, K. Gummadi, G. Weikum, iFair: Learning individually fair data representations for algorithmic decision making, in: ICDE, 2019, pp. 1334–1345.

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

[6] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, C. Dwork, Learning fair representations, in: ICML, 2013, pp. 325–333.

[7] F. Calmon, D. Wei, B. Vinzamuri, K. Ramamurthy, K. Varshney, Optimized pre-processing for discrimination prevention, in: NIPS, 2017, pp. 3992–4001.

[8] T. Kamishima, S. Akaho, H. Asoh, J. Sakuma, Fairness-aware classifier with prejudice remover regularizer, in: ECML-PKDD, 2012, pp. 35–50.

[9] M. Zafar, I. Valera, M. Rodriguez, K. Gummadi, A. Weller, From parity to preference-based notions of fairness in classification, in: NIPS, 2017, pp. 229–239.

[10] B. Zhang, B. Lemoine, M. Mitchell, Mitigating unwanted biases with adversarial learning, in: AAAI/ACM Conference on AI, Ethics, and Society, 2018, pp. 335–340.

[11] F. Kamiran, A. Karim, X. Zhang, Decision theory for discrimination-aware classification, in: ICDM, 2012, pp. 924–929.

[12] M. Hardt, E. Price, N. Srebro, Equality of opportunity in supervised learning, in: NIPS, 2016, pp. 3315–3323.

[13] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, K. Weinberger, On fairness and calibration, in: NIPS, 2017, pp. 5680–5689.

[14] M. Kim, A. Ghorbani, J. Zou, Multiaccuracy: Black-box post-processing for fairness in classification, in: AAAI/ACM Conference on AI, Ethics, and Society, 2019, pp. 247–254.

[15] P. Lohia, N. Ramamurthy, M. Bhide, D. Saha, K. Varshney, R. Puri, Bias mitigation post-processing for individual and group fairness, in: International Conference on Acoustics, Speech and Signal Processing, 2019, pp. 2847–2851.

[16] S. Friedler, C. Scheidegger, S. Venkatasubramanian, S. Choudhary, E. Hamilton, D. Roth, A comparative study of fairness-enhancing interventions in machine learning, in: Conference on Fairness, Accountability, and Transparency (FAT), 2019, pp. 329–338.

[17] F. Kamiran, S. Mansha, A. Karim, X. Zhang, Exploiting reject option in classification for social discrimination control, Information Sciences 425 (2018) 18–33.

[18] M. B. Zafar, I. Valera, M. G. Rodriguez, K. Gummadi, Fairness constraints: Mechanisms for fair classification, in: AISTAT, 2017, pp. 962–970.

[19] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, R. Zemel, Fairness through awareness, in: Innovations in Theoretical Computer Science, 2012, pp. 214–226.

[20] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, A. Galstyan, A survey on bias and fairness in machine learning, arXiv preprint arXiv:1908.09635.

[21] L. Oneto, S. Chiappa, Fairness in machine learning, in: Recent Trends in Learning From Data, Springer, 2020, pp. 155–196.

[22] D. Madras, E. Creager, T. Pitassi, R. Zemel, Learning adversarially fair and transferable representations, in: ICML, 2018, pp. 3384–3393.

[23] D. McNamara, C. S. Ong, R. Williamson, Costs and benefits of fair representation learning, in: AAAI/ACM Conference on AI, Ethics, and Society, 2019, pp. 263–270.

[24] S. Barocas, A. Selbst, Big data's disparate impact, California Law Review 104 (2016) 671.

[25] A. Agarwal, A. Beygelzimer, M. Dudik, J. Langford, H. Wallach, A reductions approach to fair classification, in: ICML, 2018, pp. 60–69.

[26] M. Zafar, I. Valera, M. Gomez-Rodriguez, K. Gummadi, Fairness constraints: A flexible approach for fair classification, Journal of Machine Learning Research 20 (75) (2019) 1–42.

[27] Y. Bechavod, K. Ligett, Penalizing unfairness in binary classification, arXiv preprint arXiv:1707.00044.

[28] C. Dwork, N. Immorlica, A. T. Kalai, M. Leiserson, Decoupled classifiers for group-fair and efficient machine learning, in: Conference on

Fairness, Accountability and Transparency (FAT), 2018, pp. 119–133.

[29] R. Nabi, D. Malinsky, I. Shpitser, Learning optimal fair policies, in: ICML, 2019, pp. 4674–4682.

[30] R. Williamson, A. Menon, Fairness risk measures, in: ICML, 2019, pp. 6786–6797.

[31] M. Kearns, S. Neel, A. Roth, Z. S. Wu, Preventing fairness gerrymandering: Auditing and learning for subgroup fairness, arXiv preprint arXiv:1711.05144.

[32] A. Cotter, H. Jiang, K. Sridharan, Two-player games for efficient non-convex constrained optimization, in: Algorithmic Learning Theory, 2019, pp. 300–332.

[33] E. Chzhen, C. Denis, M. Hebiri, L. Oneto, M. Pontil, Leveraging labeled and unlabeled data for consistent fair binary classification, in: NIPS, Vol. 32, 2019, pp. 12739–12750.

[34] A. Noriega-Campero, M. Bakker, B. Garcia-Bulle, A. Pentland, Active fairness in algorithmic decision making, in: AAAI/ACM Conference on AI, Ethics, and Society, 2019, pp. 77–83.

[35] T. Calders, A. Karim, F. Kamiran, W. Ali, X. Zhang, Controlling attribute effect in linear regression, in: ICDM, 2013, pp. 71–80.

[36] R. Berk, H. Heidari, S. Jabbari, M. Joseph, M. Kearns, J. Morgenstern, S. Neel, A. Roth, A convex framework for fair regression, arXiv preprint arXiv:1706.02409.

[37] E. Chzhen, C. Denis, M. Hebiri, L. Oneto, M. Pontil, Fair regression with Wasserstein barycenters, in: NIPS, Vol. 33, 2020, pp. 7321–7331.

[38] J. Snoek, H. Larochelle, R. Adams, Practical bayesian optimization of machine learning algorithms, in: NIPS, 2012, pp. 2951–2959.

[39] B. Shahriari, K. Swersky, Z. Wang, R. Adams, N. Freitas, Taking the human out of the loop: A review of bayesian optimization, Proceedings of the IEEE 104 (1) (2016) 148–175.

[40] D. Nguyen, S. Gupta, S. Rana, A. Shilton, S. Venkatesh, Bayesian optimization for categorical and category-specific continuous inputs, in: AAAI, 2020.

[41] D. Russo, B. Van Roy, A. Kazerouni, I. Osband, Z. Wen, et al., A tutorial on Thompson sampling, Foundations and Trends® in Machine Learning 11 (1) (2018) 1–96.

[42] L. Györfi, M. Kohler, A. Krzyzak, H. Walk, A distribution-free theory of nonparametric regression, Springer Science & Business Media, 2006.

[43] I. Steinwart, D. Hush, C. Scovel, et al., Optimal rates for regularized least squares regression., in: COLT, 2009, pp. 79–93.

[44] B. Sriperumbudur, K. Fukumizu, G. Lanckriet, Universality, characteristic kernels and rkhs embedding of measures, Journal of Machine Learning Research 12 (Jul) (2011) 2389–2410.

[45] R. Bellamy, K. Dey, M. Hind, S. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, et al., AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias, arXiv preprint arXiv:1810.01943.

[46] L. v. d. Maaten, G. Hinton, Visualizing data using t-SNE, Journal of Machine Learning Research 9 (Nov) (2008) 2579–2605.

Appendix A. More Experiments

In this section, we evaluate our method **FCGP** with two additional settings for data split. Our goal is to demonstrate that **FCGP** still works effectively on large test sets, where it improves the fairness score while maintaining an accuracy close to that of the pre-trained classifier.

Appendix A.1. Data split: 70%-15%-15%

For each dataset, we randomly split it into 70% to train the pre-trained classifier, 15% for the *unlabeled* training set to train our method and the baselines, and 15% for the *hold-out* test set. Other settings are the same as those in our previous experiments (see Section 5.3). All methods are evaluated on the *hold-out* test sets.

From Figure A.8, we can see that both our methods **FCGP-S** and **FCGP-L** are the only methods that can achieve fairness improvements without suffering significant accuracy losses on all datasets (except for *bank+marital*). In contrast, two state-of-the-art baselines ROC and IGD cannot increase the fairness scores on some datasets. For example, ROC does not improve the fairness over that of the pre-trained model on *compas+sex*, *bank+age*, and *bank+marital* while IGD suffers the same problem on *adult+race*, *compas+race*, *compas+sex*, *bank+age*, and *bank+marital*.

The baseline method Random often increases the fairness score; it however also loses the maintenance of high accuracy. This serious problem is clearly shown in four datasets *german+age, german+sex*, *bank+age*, and *bank+marital* where Random drops from 3% to 9% in its accuracy.

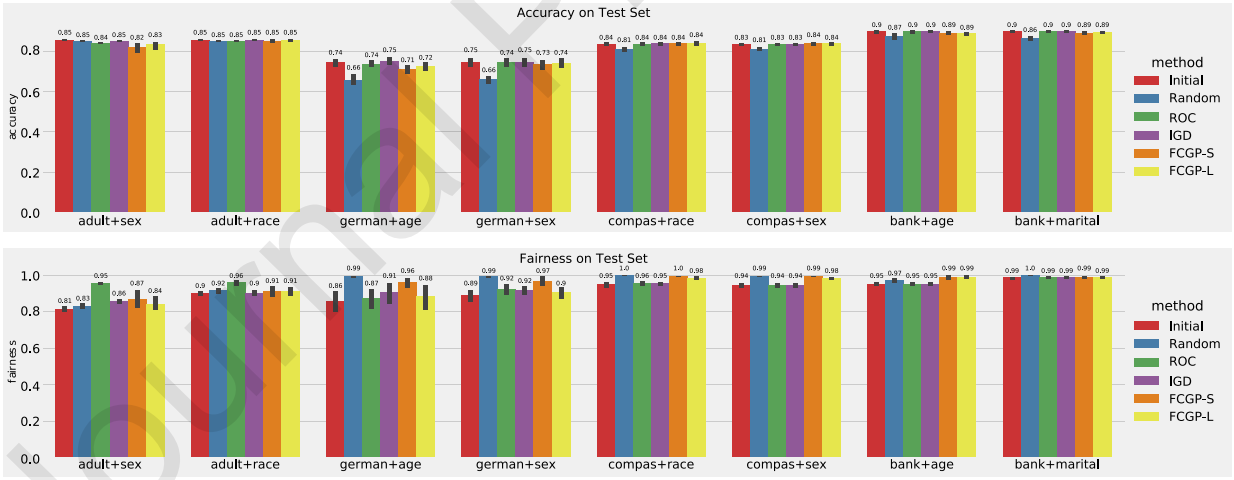


Figure A.8: Accuracy and fairness of our models **FCGP-S**, **FCGP-L**, the pre-trained classifier, and state-of-the-art baselines on four datasets (each of them has two sensitive features) with a data split of 70%-15%-15%. The performance of each method is measured on a *hold-out* test set.

Appendix A.2. Data split: 50%-25%-25%

In this setting, we use the data split 50%-25%-25% for each dataset. Namely, we use 50% of data points to train the pre-trained classifier, 25% of data points without labels to train our method and other baselines, and 25% of data

points as the *hold-out* test set to evaluate all methods.

Figure A.9 reports the accuracy and fairness scores of each method on the *hold-out* test sets. From the figure, we can observe that our method **FCGP** often improves the fairness and keeps a similar accuracy to that of the pre-trained classifier. On three datasets *compas+race*, *compas+sex*, and *bank+age*, **FCGP** can increase the fairness scores up to 1.0 whereas these three datasets are hard for two baselines ROC and IGD to improve the fairness. Neither ROC nor IGD can raise the fairness scores on these three datasets.

We observe a similar behavior for the Random method under this setting of data split, where it often increases the fairness score but decreases the accuracy score.

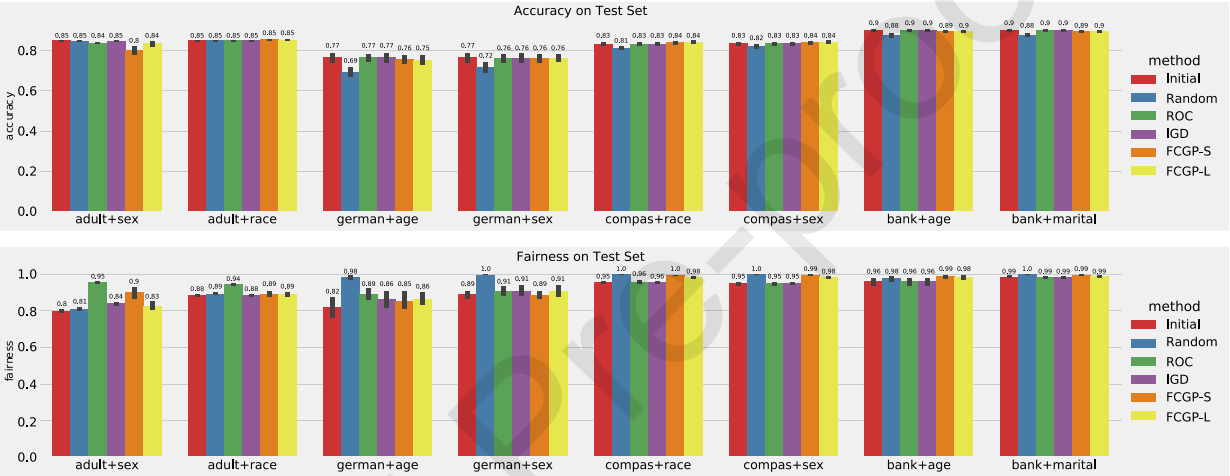


Figure A.9: Accuracy and fairness of our models **FCGP-S**, **FCGP-L**, the pre-trained classifier, and state-of-the-art baselines on four datasets (each of them has two sensitive features) with a data split of 50%-25%-25%. The performance of each method is measured on a *hold-out* test set.

In summary, the experimental results demonstrate that our method **FCGP** works effectively with different settings for data split, where it can enhance the fairness without significantly damaging the accuracy.