Tuning Fairness by Marginalizing Latent Target Labels

Thomas Kehrenberg, Zexun Chen, and Novi Quadrianto*

Predictive Analytics Lab (PAL), University of Sussex, Brighton, UK
{t.kehrenberg,zexun.chen,n.quadrianto}@sussex.ac.uk

Abstract Addressing fairness in machine learning models has recently attracted a lot of attention, as it will ensure continued confidence of the general public in the deployment of machine learning systems. Here, we focus on mitigating harm of a biased system that offers better outputs (e.g. loans, jobs) for certain groups than for others. We show that bias in the output can naturally be handled in probabilistic models by introducing a latent target output that will modulate the likelihood function. This simple formulation has several advantages: first, it is a unified framework for several notions of fairness such as demographic parity and equalized odds; second, it is expressed as marginalization instead of constrained problems; and third, it allows encoding our knowledge of what the bias in outputs should be. Practically, the latter translates to the ability to control the level of fairness by varying directly fairness target rates. In contrast, existing approaches rely on intermediate, arguably unintuitive control parameters such as a covariance threshold.

Keywords: Fair classification · Gaussian process · Label bias

1 Introduction

Algorithmic assessment methods are used for predicting human outcomes such as bail decision and mortgage approval. This contributes, in theory, to a world with decreasing human biases. To achieve this, however, we need advanced machine learning models that are free of algorithmic biases (fair models), despite the fact that they are *written* by humans and trained based on historical and *biased data.*

There is no single accepted definition of algorithmic fairness for automated decision-making though several have been proposed. One definition is referred to as *statistical* or *demographic parity.* Given a binary sensitive attribute (married/unmarried) and a binary decision (yes/no to getting a mortgage), demographic parity requires "yes" decisions of married individuals to be at the same rate as "yes" decisions of *un*married individuals, i.e. $\mathbb{P}(\text{mortgage} = \text{yes}|\text{married}) = \mathbb{P}(\text{mortgage} = \text{yes}|\text{not married})$. Another fairness criterion, *equalized odds* [15], takes into account binary label (yes/no in making a payment), and requires equal true positive rates (TPR) and false

---

* Also with National Research University Higher School of Economics, Moscow.

positive rates (FPR) across married and *un*married groups, i.e. $\mathbb{P}(\text{mortgage} = \text{yes}|\text{married}, \text{payment} = \text{yes}) = \mathbb{P}(\text{mortgage} = \text{yes}|\text{not married}, \text{payment} = \text{yes})$ for equal TPR rates, and accordingly for the FPR rates.

Many models are available to enforce demographic parity or equalized odds (e.g. [1,5,20,33,34]), however none of them give humans the control to set the *rate* of positive predictions (e.g. a PR of 0.6), or the rate of true positives (e.g. a TPR of 0.6). What is the advantage of being able to control PR/TPR/FPR rates? In this paper, we show that we can actually control the level of fairness by tuning directly those target rates. This means machine learning practitioners can trade off fairness and accuracy by directly controlling parameters that are arguably intuitive, understandable to the general public. In contrast, to balance accuracy and fairness, existing approaches use intermediate, unintuitive control parameters such as allowable constraint violation ϵ (e.g. 0.01) in [1], or covariance threshold c (e.g. 0 that is controlled by another parameters τ and $\mu - 0.005$ and 1.2 – to trade off this threshold and accuracy) in [33].

We propose a method for incorporating fairness in probabilistic classifiers. We assume the existence of *unbiased* output decision, which will modulate the likelihood term of the classifier. With this formulation, we can show the theoretical mutual exclusivity of demographic parity and equalized odds (cf. [6,22]) as a by-product of the sum and product probability rules. This is in stark contrast to many existing approaches that embed fairness criteria as constraints in the optimization procedure (e.g. [10,28,33,34]); those methods can then violate mutual exclusivity as there is no mechanism to prevent multiple constraints being added. We instantiate our approach with a parametric logistic regression classifier and a Bayesian nonparametric Gaussian process classifier (GPC). For the latter, as our formulation is not expressed as a constrained problem, we can reuse advancements in automated variational inference [4,13,23] for learning the fair model, and for handling a large amount of data.

Related work. There are several ways to enforce fairness in machine learning models: as a pre-processing step (e.g. [19,25,26,29,35]), as a post-processing step (e.g. [11,15]), or as a constraint during the learning phase (e.g. [5,10,32,33,34]). Our method enforces fairness during the learning phase, but, unlike other approaches, we do not cast fair learning as a *constrained* optimization problem. Constrained optimization requires a customized optimization procedure. In Goh et al. [14] and Zafar et al. [33,34], suitable majorization-minimization/convex-concave procedures [24] were derived. Furthermore, such constrained optimization approaches may lead to more unstable training, and often yield classifiers with both worse accuracy and more unfair [7]. Our proposed method can be solved by *off-the-shelf* packages, for example, we can use GPC packages by Dezfouli and Bonilla [8] or Gardner et al. [13], which only need conditional likelihood evaluation as a black-box function. Many of the recently proposed methods [1,27,28] attempt to have a unified framework that can be instantiated for either demographic parity or equalized odds criteria. Our method also provides a unified framework. Furthermore, the setting of free parameters in our model transparently highlights the mutual exclusivity of demographic parity and

equalized odds. Approaches closely related to ours were given by Kamiran and Calders [19] who present several pre-processing methods. One of them makes the training data fairer by changing the labels. However, the method requires training a baseline classifier on the original data to determine which labels to flip. Furthermore, this label flipping is all-or-nothing, whereas, in our approach, labels can be probabilistically flipped. Another method from [19] is reweighting the training data. However, it is lacking a target rate mechanism and it is only applicable for demographic parity. The recent work by Agarwal et al. [1] extended ideas from [19] to equalized odds.

2   Target labels for handling label bias

In order to motivate the introduction of our concept of target labels, we consider the fairness criterion demographic parity as an illustrative example. In demographic parity, we demand that the overall probability of being assigned a positive prediction ($\hat{y} = 1$) is the same for all demographic groups s (here with $s \in \{0, 1\}$): $\mathbb{P}(\hat{y} = 1|s = 0) = \mathbb{P}(\hat{y} = 1|s = 1)$. Enforcing this criterion can be understood as learning from a dataset with "incorrect" labels. This can be seen as follows: The above equation does not (in general) hold for the labels in the training set. Therefore, at test time, the fair classifier in the sense of demographic parity makes predictions that are distributed differently than the labels in the dataset. From the perspective of the fair classifier, the training labels are "wrong", because they are biased.

Furthermore, we argue that for *classification*, it is not useful to consider any bias in the features and we will therefore only consider the bias in the labels. This is because when building a fair classifier it is not necessary to explicitly construct a fair version of the features, all that matters in the end is the fair prediction. If the classifier learns from the unbiased labels, it will find an implicit representation of the features that can predict these unbiased labels. There are, however, situations where an explicit fair representation of the features is necessary; e.g. when wanting to sell the data [27].

Inspired by this point of view, we introduce a new variable to represent the unbiased labels (or "true" labels): $\bar{y}$. We call these the *target labels*. The target labels are unknown but using the prior knowledge about what the target labels should look like, we can establish a relationship between the training labels and the target labels. The idea is as follows: In probabilistic models, a classifier outputs a score for $y = 1$: $c(x, \theta) = \mathbb{P}(y = 1|x, \theta)$ where θ is the model parameters. The loss is then computed from the score and the actual label y: $\mathcal{L}(c, y)$, e.g. the negative Bernoulli log-likelihood for binary classification problems, or the negative Categorical log-likelihood for the multi-class setting. In our framework, the classifier predicts $\bar{y}$: $\bar{c}(x, \theta) = \mathbb{P}(\bar{y} = 1|x, \theta)$. However, the loss function remains the same and thus we need a mapping from $\bar{c}$ to c, i.e. $g : \bar{c} \mapsto c$. In fact, it will be the mapping g that defines the target labels.

In the most general case, the mapping function g depends on the features x and the demographic group s, in addition to the predicted score $\bar{c}(x, \theta)$. However,
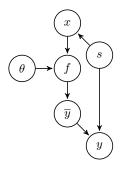
Figure 1: A probabilistic graphical model of our assumptions about biased data. Variable f represents the raw output of the classification model.

Algorithm 1: Training loop with Target Labels

Input: Training set $\mathcal{D} = \{(x_i, y_i, s_i)\}_{i=1}^N$, debiasing parameters $d_{\bar{y}=0}^{s=0}$, $d_{\bar{y}=1}^{s=0}$, $d_{\bar{y}=0}^{s=1}$, $d_{\bar{y}=1}^{s=1}$

Output: fair model parameters θ

1: Initialize θ (randomly)
2: **for all** x_i, y_i, s_i **do**
3:     $P_{\bar{y}=1} \leftarrow \bar{c}(x_i, \theta)$ (e.g. logistic($\langle x, \theta \rangle$))
4:     $P_{\bar{y}=0} \leftarrow 1 - P_{\bar{y}=1}$
5:     **if** $s_i = 0$ **then**
6:         $P_{y=1} \leftarrow d_{\bar{y}=0}^{s=0} \cdot P_{\bar{y}=0} + d_{\bar{y}=1}^{s=0} \cdot P_{\bar{y}=1}$
7:     **else**
8:         $P_{y=1} \leftarrow d_{\bar{y}=0}^{s=1} \cdot P_{\bar{y}=0} + d_{\bar{y}=1}^{s=1} \cdot P_{\bar{y}=1}$
9:     **end if**
10:     $\ell \leftarrow y_i \cdot P_{y=1} + (1 - y_i) \cdot (1 - P_{y=1})$
11:     update θ to maximize likelihood ℓ
12: **end for**

---

in this paper, we consider only the simplest non-trivial case where g only depends on $\bar{c}(x, \theta)$ and s. With this choice of mapping function, we therefore assume that there is no additional information in the features x about the bias in y if s *is already known*: $y \perp x | \bar{y}, s$. In other words, we assume that the score $\bar{c}(x, \theta)$ contains all the relevant information from x for reconstructing y, given s. Lettings g depend (*directly*) on x is also possible in principle, but in practice it means learning a complicated function of input x, where we do not have the true score $\bar{c}$ that would be needed to learn it. Furthermore, we concentrate on the case where the transformation is *linear* because this allows us to interpret the mapping as marginalization of a joint probability. These probabilities are interpretable and allow us to define a mapping that will enforce various fairness constraints.

The mapping from $\bar{c}$ to c is then:

$$\mathbb{P}(y = 1 | x, s, \theta) = g(\bar{c}(x, \theta), s) = m_s \cdot \mathbb{P}(\bar{y} = 1 | x, s, \theta) + b_s$$

$$= \sum_{\bar{y} \in \{0,1\}} \mathbb{P}(y = 1 | \bar{y}, s) \mathbb{P}(\bar{y} | x, \theta) = \sum_{\bar{y} \in \{0,1\}} \mathbb{P}(y = 1, \bar{y} | x, s, \theta) \quad (1)$$

with $m_s = \mathbb{P}(y = 1 | \bar{y} = 1, s) - \mathbb{P}(y = 1 | \bar{y} = 0, s)$ and $b_s = \mathbb{P}(y = 1 | \bar{y} = 0, s)$. On the last line in (1), we used $\mathbb{P}(y | \bar{y}, s) = \mathbb{P}(y | \bar{y}, x, \theta, s)$ and $\mathbb{P}(\bar{y} | x, \theta) = \mathbb{P}(\bar{y} | x, \theta, s)$; the former follows from the previously stated conditional independence assumption and the latter reflects the desideratum that the fair prediction $\bar{y}$ should not depend on s, such that s is not needed to make predictions at test time. This requirement is not strictly necessary and, in fact, s can simply be added as an additional feature to the input x if so desired. From our experience, doing so improves accuracy as well as fairness. In that case, s needs to be available at training and at test time. However, not using s at prediction time can be desir-

able in order to avoid *Disparate Treatment*[1] [3]. The graphical model in Fig. 1 summarizes our assumptions.

We refer to these parameters m_s and b_s as the *debiasing parameters*. We give an intuition for the meaning of these parameters and derive concrete values in Section 3. For a binary sensitive attribute s (and binary label y), there are 4 debiasing parameters (see Algorithm 1 where $d_{\bar{y}=i}^{s=j} := \mathbb{P}(y = 1|\bar{y} = i, s = j)$):

$$\mathbb{P}(y = 1|\bar{y} = 0, s = 0), \qquad \mathbb{P}(y = 1|\bar{y} = 1, s = 0) \qquad (2)$$
$$\mathbb{P}(y = 1|\bar{y} = 0, s = 1), \qquad \mathbb{P}(y = 1|\bar{y} = 1, s = 1) . \qquad (3)$$

The above derivation was given for the case of binary classification but it can be easily extended to multi-class predictions.

3   Realization of concrete fairness constraints

This section focuses on how to set values of the debiasing parameters for tuning a variety of fairness target rates.

3.1   Targeting an acceptance rate/positive rate

Before we consider concrete values, we give a quick overview on the intuition behind the debiasing parameters. Let $s = 0$ refer to the disadvantaged group. For this group, we want to make more positive predictions than the dataset labels indicate. Variable $\bar{y}$ is supposed to be our fair label. Thus, in order to make more positive predictions, some of the $y = 0$ labels should be associated with $\bar{y} = 1$. However, we do not know which. So, if our model predicts $\bar{y} = 1$ (high $\mathbb{P}(\bar{y} = 1|x, \theta)$) while the dataset label is $y = 0$, then we allow for the possibility that this is actually correct. That is, $\mathbb{P}(y = 0|\bar{y} = 1, s = 0)$ is not 0. If we choose, for example, $\mathbb{P}(y = 0|\bar{y} = 1, s = 0) = 0.3$ then that means that 30% of positive virtual labels $\bar{y} = 1$ may correspond to negative dataset labels $y = 0$. This way we can have more $\bar{y} = 1$ than $y = 1$, overall. On the other hand, predicting $\bar{y} = 0$ when $y = 1$ holds, will always be deemed incorrect: $\mathbb{P}(y = 1|\bar{y} = 0, s = 0) = 0$; this is because we do not want any additional negative labels.

For the advantaged group $s = 1$, we have the exact opposite situation. If anything, we have too many positive labels. So, if our model predicts $\bar{y} = 0$ (high $\mathbb{P}(\bar{y} = 0|x, \theta)$) while the dataset label is $y = 1$, then we should again allow for the possibility that this is actually correct. That is, $\mathbb{P}(y = 1|\bar{y} = 0, s = 1)$ should not be 0. On the other hand, $\mathbb{P}(y = 0|\bar{y} = 1, s = 1)$ should be 0 because we do not want additional positive labels for $s = 1$. It could also be that the number of positive labels is exactly as it should be, in which case we can just set $y = \bar{y}$ for all data points with $s = 1$.

---

[1] Note that knowing s is needed for training in order to map from $\bar{y}$ to y but is not needed at test time where we just predict $\bar{y}$.

We now give concrete values for the debiasing parameters in Eqs (2)-(3). We first apply Bayes' rule to the debiasing parameters:

$$\mathbb{P}(y = 0|\bar{y} = 0, s) = \frac{\mathbb{P}(\bar{y} = 0|y = 0, s)\mathbb{P}(y = 0|s)}{\mathbb{P}(\bar{y} = 0|s)} \tag{4}$$

$$\mathbb{P}(y = 1|\bar{y} = 1, s) = \frac{\mathbb{P}(\bar{y} = 1|y = 1, s)\mathbb{P}(y = 1|s)}{\mathbb{P}(\bar{y} = 1|s)} \tag{5}$$

In the following, $s = i$ refers to either $s = 0$ or $s = 1$. The term $\mathbb{P}(y = 1|s = i)$ is the acceptance rate in group i of the biased training labels. We call it the *biased acceptance rate*. This quantity can be estimated from the training data:

$$\mathbb{P}(y = 1|s = i) = \frac{\text{number of points with } y = 1 \text{ in group } i}{\text{number of points in group } i} .$$

The term $\mathbb{P}(\bar{y} = 1|s = i)$ is the *target acceptance rate*. This can*not* be estimated from the data; it has to be known about the unbiased data. We will later discuss strategies to choose this. This target acceptance rate is related to the other parameters in the following way:

$$\mathbb{P}(\bar{y}|s = i) = \sum_{j \in \{0,1\}} \mathbb{P}(\bar{y}|y = j, s = i)\mathbb{P}(y = j|s = i) . \tag{6}$$

With a given target acceptance rate and a given biased acceptance rate, this constraint still leaves two degrees of freedom that can be represented as $\mathbb{P}(\bar{y} = 1|y = 1, s = i)$ for $s \in \{0, 1\}$. If we consider y the label and $\bar{y}$ the prediction, then this would be the true positive rate. As such, we call these two degrees of freedom the *biased* TPRs. It is tempting to use these biased TPRs to enforce fairness constraints based on the TPR (e.g. equality of opportunity) but after choosing a target acceptance rate, there are additional constraints on these biased TPRs.

The biased TPRs will affect the accuracy with respect to the biased labels. For example, $\mathbb{P}(\bar{y} = 1|y = 1, s) = 50\%$ will lead to predictions that look random with respect to the biased labels. In order to minimize the drop in accuracy w.r.t. the biased labels, we need to maximize the biased TPRs.

For $\mathbb{P}(\bar{y} = 1|s = i) > \mathbb{P}(y = 1|s = i)$, the terms $\mathbb{P}(\bar{y} = 1|y = 1, s = i)$ can be set to 1. In the case of $\mathbb{P}(\bar{y} = 1|s = i) < \mathbb{P}(y = 1|s = i)$, it follows that $\mathbb{P}(\bar{y} = 0|s = i) > \mathbb{P}(y = 0|s = i)$. Here we set $\mathbb{P}(\bar{y} = 0|y = 0, s = i)$ (the biased TNR) to 1 and compute the biased TPR via the constraint in Eq (6) which leads to the maximum possible value for the biased TPR. Algorithm 2 shows the pseudocode describing this procedure.

Demographic Parity. A simple strategy for the target rate is demographic parity where we want to enforce

$$\mathbb{P}(\bar{y}|s = 0) = \mathbb{P}(\bar{y}|s = 1) = \mathbb{P}(\bar{y}) . \tag{7}$$

This means we only choose one target rate for both groups. We denote this by $PR_t := \mathbb{P}(\bar{y} = 1)$ (PR: positive rate).

Algorithm 2: Targeting PR	**Algorithm 3:** Targeting TPR/TNR
Input: target rate PR_t, biased acceptance rate $\mathbb{P}(y = 1\|s = i)$	**Input:** target rates TPR_t, TNR_t, biased acceptance rate $\mathbb{P}(y = 1\|s = i)$
Output: debiasing parameter $d_{\bar{y}=j}^{s=i}$	**Output:** debiasing parameter $d_{\bar{y}=j}^{s=i}$
1: **if** j=0 **then**	1: $\mathbb{P}(\bar{y} = 1, y = 0\|s = i) \leftarrow (1 - TNR_t) \cdot$
2: $\mathbb{P}(\bar{y} = j\|s = i) \leftarrow 1 - PR_t$	$(1 - \mathbb{P}(y = 1\|s = i))$
3: **else if** j=1 **then**	2: $\mathbb{P}(\bar{y} = 1, y = 1\|s = i) \leftarrow TPR_t \cdot \mathbb{P}(y =$
4: $\mathbb{P}(\bar{y} = j\|s = i) \leftarrow PR_t$	$1\|s = i)$
5: **end if**	3: $\mathbb{P}(\bar{y} = 1\|s = i) \leftarrow \mathbb{P}(\bar{y} = 1, y = 0\|s =$
6: **if** $\mathbb{P}(\bar{y} = j\|s = i) > \mathbb{P}(y = 1\|s = i)$ **then**	$i) + \mathbb{P}(\bar{y} = 1, y = 1\|s = i)$
7: $d_{\bar{y}=j}^{s=i} \leftarrow \frac{\mathbb{P}(y=1\|s=i)}{\mathbb{P}(\bar{y}=j\|s=i)}$	4: **if** j=0 **then**
8: **else**	5: $d_{\bar{y}=0}^{s=i} \leftarrow \frac{1-\mathbb{P}(\bar{y}=1,y=1\|s=i)}{1-\mathbb{P}(\bar{y}=1\|s=i)}$
9: $d_{\bar{y}=j}^{s=i} \leftarrow 1$	6: **else if** j=1 **then**
10: **end if**	7: $d_{\bar{y}=1}^{s=i} \leftarrow \frac{\mathbb{P}(\bar{y}=1,y=1\|s=i)}{\mathbb{P}(\bar{y}=1\|s=i)}$
	8: **end if**

When choosing the target rate, we again take into account what the effect is on the accuracy with respect to the biased labels. $\mathbb{P}(\bar{y}) \neq \mathbb{P}(y)$ in the predictions necessarily implies that, for some input x, $\bar{y} \neq y$. To keep the drop in accuracy to a minimum, PR_t has to be between $\mathbb{P}(y = 1|s = 0)$ and $\mathbb{P}(y = 1|s = 1)$.

Natural options are,

$$PR_t^{avg} = \tfrac{1}{2} \cdot (\mathbb{P}(y = 1|s = 0) + \mathbb{P}(y = 1|s = 1))$$
$$PR_t^{max} = \max (\mathbb{P}(y = 1|s = 0), \mathbb{P}(y = 1|s = 1))$$
$$PR_t^{min} = \min (\mathbb{P}(y = 1|s = 0), \mathbb{P}(y = 1|s = 1))$$

where $\mathbb{P}(y = 1|s = i)$ is the estimated biased acceptance rate in group i. We find that using the mean for the target (PR_t^{avg}) is a safer choice than PR_t^{min} and PR_t^{max}. It is easier for the targeting mechanism to move both PR_t^{min} and PR_t^{max} to PR_t^{avg} than to move PR_t^{min} to PR_t^{max} or vice versa. We investigate this in Section 5.

3.2 Targeting a true positive rate

Whereas for demographic parity we enforce a constraint on $\mathbb{P}(\bar{y}|s)$, for equality of opportunity the constraint is on the TPR (true positive rate) $\mathbb{P}(\bar{y} = 1|y = 1, s)$. By Eqs (4) and (5), the debiasing parameters are *fully determined* by $\mathbb{P}(\bar{y} = 0|y = 0, s)$, $\mathbb{P}(\bar{y} = 1|y = 1, s)$ and $\mathbb{P}(y|s)$. The last of which can be estimated from the training data. This highlights the general *mutual exclusivity* of demographic parity and equality of opportunity [22,6].

Equality of opportunity demands that

$$\mathbb{P}(\hat{y} = 1|y = 1, s = 0) = \mathbb{P}(\hat{y} = 1|y = 1, s = 1) \tag{8}$$

where $\hat{y}$ is the prediction of the classifier. Assuming that the classifier (GPC or logistic regression models) perfectly learns the target labels $\bar{y}$, we can fulfill

this demand by enforcing (8) on $\bar{y}$. At first glance, it seems desirable to set both the target TNR (true negative rate), $TNR_t = \mathbb{P}(\bar{y} = 0|y = 0, s)$, and the target TPR, $TPR_t = \mathbb{P}(\bar{y} = 1|y = 1, s)$, to 1, because any value lower than 1 necessarily reduces the accuracy. However, when target TNRs and target TPRs are all set to 1, then the debiasing parameters are likewise all 1, which is equivalent to $\bar{y} = y$ for all data-points. This is just the standard classification model (without considering fairness). This problem can be understood in the following way. A perfect predictor would predict all labels correctly, that is $\hat{y} = y$. This automatically fulfills equality of opportunity. Generally, our predictors are not perfect, however, so they make some classification errors. What equality of opportunity demands is that this classification error is the same for all specified groups. By setting the target TPR (TPR_t) to a lower value that is the same for all groups, we purposefully sacrifice some accuracy to make the errors the same. This sacrifice should be as small as possible. Algorithm 3 shows the pseudocode for computing the debiasing parameters from TPR_t and TNR_t.

Choosing values for the TPR target rate (and the TNR target rate) is not as straightforward as in the case of targeting an acceptance rate because TPR and TNR are inextricably linked to the classifier that is used. We additionally found that the achieved TPR does not just depend on the target TPR, but also on the target TNR. More specifically, targeting a lower TNR makes it easier to achieve a higher TPR. This is not surprising, as lowering the TNR will result in more positive predictions ($\hat{y} = 1$), which means that the general threshold for a positive predictions is lowered. This lowered threshold makes it more likely that a given false negative prediction will be flipped; i.e., becomes a true positive prediction. A decrease of false negatives coupled with an increase in true positives will increase the TPR. We investigate this trade-off between TPR and TNR in our experiments (e.g. Fig. 5(c) and (d)).

We use the following method to find a good value for the target TNR. We train a standard (unfair) classification model on the training set and evaluate the model on the test set. From this evaluation we compute the achievable TNR and TPR separately for each group. Of the two achievable TNRs (one for $s = 0$, one for $s = 1$), we take the minimum as the target TNR for all groups. We set the TNR to be the same for both groups so that the effect on the TPR is the same as well. This makes it easier to achieve equal TPRs. Technically, this enforces the fairness criterion *equalized odds* in which both TPR and TNR must be the same:

$$\mathbb{P}(\hat{y} = 1|y = 1, s = 0) = \mathbb{P}(\hat{y} = 1|y = 1, s = 1)$$
$$\mathbb{P}(\hat{y} = 0|y = 0, s = 0) = \mathbb{P}(\hat{y} = 0|y = 0, s = 1) \ .$$

However, *equalized odds* implies *equality of opportunity*.

For the target TNR, we take here the minimum of the two observed TNRs. This is not strictly necessary and was a choice we made to attain higher TPRs. An equivalent approach can be used for selecting the TPRs. We could choose the maximum of the two observed TPRs, after choosing the minimum for the

Table 1: Accuracy and fairness (with respect to *demographic parity*) for various methods on the Adult dataset. Fairness is defined as $PR_{s=0}/PR_{s=1}$ (a completely fair model would achieve a value of 1.0). Left: using **race** as the sensitive attribute. Right: using **gender** as the sensitive attribute. The mean and std of 10 repeated experiments.

Algorithm	Fair $\to 1.0 \leftarrow$	Accuracy $\uparrow$	Fair $\to 1.0 \leftarrow$	Accuracy $\uparrow$
GP	0.56 ± 0.02	0.853 ± 0.002	0.32 ± 0.03	0.854 ± 0.003
GP, use s	0.50 ± 0.03	0.854 ± 0.003	0.31 ± 0.02	0.854 ± 0.002
LR	0.57 ± 0.03	0.846 ± 0.003	0.33 ± 0.02	0.847 ± 0.002
LR, use s	0.74 ± 0.03	0.846 ± 0.002	0.34 ± 0.02	0.847 ± 0.003
SVM	0.61 ± 0.02	0.859 ± 0.002	0.26 ± 0.02	0.857 ± 0.003
FairGP (ours)	0.62 ± 0.03	0.853 ± 0.003	0.65 ± 0.04	0.846 ± 0.004
FairGP, use s (ours)	0.89 ± 0.05	0.850 ± 0.003	0.71 ± 0.04	0.845 ± 0.004
FairLR (ours)	0.73 ± 0.04	0.844 ± 0.003	0.67 ± 0.03	0.839 ± 0.003
FairLR, use s (ours)	1.03 ± 0.08	0.843 ± 0.002	0.71 ± 0.03	0.838 ± 0.003
ZafarAccuracy [34]	1.31 ± 0.36	0.800 ± 0.012	1.22 ± 0.40	0.793 ± 0.009
ZafarFairness [34]	0.58 ± 0.14	0.846 ± 0.003	0.35 ± 0.10	0.846 ± 0.003
Kamiran&Calders [19]	0.60 ± 0.03	0.831 ± 0.003	0.64 ± 0.03	0.847 ± 0.003
Agarwal et al. [1]	0.61 ± 0.06	0.847 ± 0.003	0.35 ± 0.03	0.847 ± 0.003

TNR. In our experiments, we do not restrict ourselves to this choice but rather investigate using different values for the target TPR.

4   Implementation

The proposed method works with any likelihood-based algorithm but will work best if the predicted probabilities are well-calibrated. We consider both a nonparametric and a parametric model. The nonparametric model is a Gaussian process model and is our main algorithm, given its good calibration. Logistic regression is the parametric counterpart. Our fairness approach is not framed as a constrained optimization problem and thus is well-suited to be used with Gaussian process models. A Gaussian process (GP) defines a distribution over functions f, that is, a sample from a Gaussian process is a function. This allows us to work directly on the space of functions. Gaussian process models are a powerful and versatile tool for solving a variety of machine learning problems, including classification; please refer to [30] for a review. To fully characterize a Gaussian process, we need to define its mean function $m(x) := \mathbb{E}[f(x)]$ and its covariance function $k(x, x') := \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$. In order to train Gaussian process models, we look for the hyperparameters that maximize the posterior of the latent function f given the training data: $\mathbb{P}(f|y, x)$. This posterior can be derived from the likelihood $\mathbb{P}(y|f)$ and the prior $\mathbb{P}(f|x)$ with Bayes' Rule. The important thing to note is that *posterior inference* in the classification model is very challenging due to non-Gaussian likelihood functions.

Recently, there have been several attempts to develop *black-box* inference techniques for Gaussian process models [8,16,17,31]. Variational inference [18]

is a widely-used technique as it enables use of stochastic optimization procedures. In the variational inference framework, all the parameters, including hyperparameters in the covariance function, variational parameters and likelihood parameters, are learned by maximizing the evidence lower bound (ELBO), as a lower bound of the marginal likelihood, $(\mathcal{L}_{elbo})$, that is: $\mathcal{L}_{\text{elbo}} = \mathcal{L}_{\text{ent}} + \mathcal{L}_{\text{cross}} + \mathcal{L}_{\text{ell}}$. Here, $\mathcal{L}_{\text{ent}}$, $\mathcal{L}_{\text{cross}}$ and $\mathcal{L}_{\text{ell}}$ are the entropy, the cross-entropy of variational distribution and the expected log-likelihood respectively. Since the first two terms correspond to the (negative) KL-divergence between the approximate posterior and prior, they do not rely on the observed data, including the sensitive attributes. Therefore, in the black-box inference framework, we only need to provide the evaluation of expected log likelihood, which, for our model, is described in Algorithm 1. Instead of $\mathbb{P}(\bar{y}|x, \theta)$ as the score function $(\bar{c}(x, \theta))$, we now have $\mathbb{P}(\bar{y}|f)$ where f is a function drawn from the GP. This does not change the calculation of the label mapping function g. Our method is therefore simple as we can reuse advancements in automated variational inference.

For our GP implementation, we are using the GPyTorch library by Gardner et al. [13], which incorporates recent advances in scalable variational inference including variational inducing variables and likelihood ratio/REINFORCE estimators for computing the gradient of $\mathcal{L}_{\text{ell}}$ term.

In the case of logistic regression model, the function f is parameterized by a weight vector and a bias term (we have previously denoted those parameters as θ). The score function in Algorithm 1 is then given by $\mathbb{P}(\bar{y} = 1|x, \theta) = \sigma(\langle x, \theta \rangle + \theta_0)$, where $\sigma(\cdot)$ is the logistic function and θ_0 is the bias.
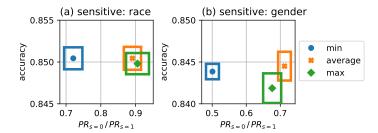
5   Experiments

We compare the performance of our target-label model with other existing models based on two real-world datasets. These datasets have been previously considered in the fairness-aware machine learning literature.

Data. The first dataset is the **Adult Income** dataset [9]. It contains 33,561 data points with census information from US citizens. The labels indicate whether the individual earns more ($y = 1$) or less ($y = 0$) than \$50,000 per year. We use the dataset with *race* and *gender* as the sensitive attribute. The input dimension, excluding the sensitive attributes, is 12 in the raw data; the categorical features are then one-hot encoded. For the experiments, we removed 2,399 instances with missing data and used only the training data, which we split randomly for each trial run. The second dataset is the **ProPublica recidivism** dataset. It contains data from 6,167 individuals that were arrested. The data was collected for the COMPAS risk assessment tool [2]. The task is to predict whether the person was rearrested within two years ($y = 1$ if they were rearrested, $y = 0$ otherwise). We again use the dataset with *race* and *gender* as the sensitive attributes.

Method. We evaluate two versions of our target label model[2]: *FairGP*, which is based on Gaussian Process models, and *FairLR*, which is based on logistic

---

[2] The code can be found on GitHub: https://github.com/predictive-analytics-lab/fair-gpytorch.

Figure 2: Accuracy and fairness (demographic parity) for various target choices. (a): Adult dataset using race as the sensitive attribute; (b): Adult dataset using gender. Center of the box is the mean; height and width of the box encode half of standard derivation of accuracy and disparate impact.

regression. Both are used in two ways: only using s for training and not for predictions (the default case), and using s during training and for predictions (indicated by "use s"). We also train baseline models that do not take fairness into account and do not use s as input.

The fair GP models and the baseline GP model are all based on variational inference and use the same settings. The batch size is set to 10100 so that there are 2 batches per epoch on the Adult dataset. The number of inducing inputs is 500 on the ProPublica dataset and 2500 on the Adult dataset which corresponds to approximately 1/8 of the number of training points for each dataset. We use a squared-exponential (SE) kernel with automatic relevance determination (ARD) and the probit function as the likelihood function. We optimize the hyperparameters and the variational parameters with the Adam method [21] with the default parameters ($\beta_1 = 0.9$, $\beta_2 = 0.999$). We use the full covariance matrix for the Gaussian variational distribution. The logistic regression is trained with Adam and uses L2 regularization. For the regularization coefficient, we conducted a hyperparameter search over 10 folds of the data. For each fold, we picked the hyperparameter which achieved the best fairness among those 5 with the best accuracy scores. We then averaged over the 10 hyperparameter values chosen in this way and then used this average for all runs to obtain our final results. For the Adult dataset, the regularization parameter is 0.0024 for ProPublica and 0.00035 for Adult.

In addition to the GP and LR baselines, we compare our proposed model with the following methods: Support Vector Machine (*SVM*), *Kamiran & Calders* [19] ("reweighing" method), *Agarwal et al.* [1] (using logistic regression as the classifier) and several methods given by Zafar et al. [34,33], which include maximizing accuracy under demographic parity fairness constraints (*ZafarFairness*), maximizing demographic parity fairness under accuracy constraints (*ZafarAccuracy*), and removing disparate mistreatment by constraining the false negative rate (*ZafarEqOpp*). We make use of the fairness comparison by Friedler et al. [12] where
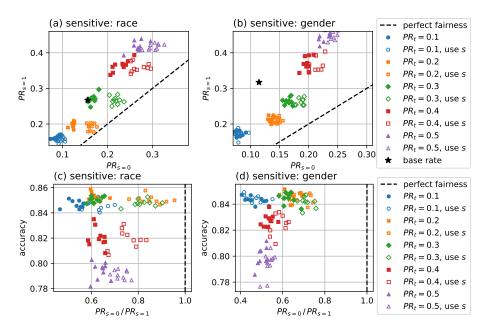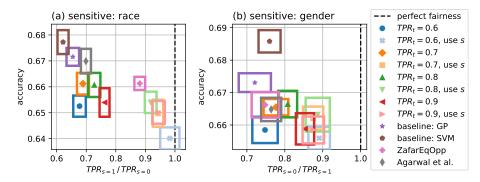
Figure 3: Predictions with different target acceptance rates (demographic parity) on Adult dataset. **(a)** and **(b)**: $PR_{s=0}$ vs $PR_{s=1}$. **(c)** and **(d)**: $PR_{s=0}/PR_{s=1}$ vs accuracy. Left column: using race as the sensitive attribute; Right column: using gender. The *base rate* indicates the positive rates of the training data.

every method is evaluated over 10 repeats that each have different splits of the training and test set.

Results for Demographic Parity on Adult dataset. Following Zafar et al. [34] we evaluate demographic parity on the Adult dataset. Table 1 shows the accuracy and fairness for several algorithms. In the table, and in the following, we use $PR_{s=i}$ to denote the observed rate of positive predictions per demographic group $\mathbb{P}(\hat{y} = 1|s = i)$. Thus, $PR_{s=0}/PR_{s=1}$ is a measure for demographic parity where a completely fair model would attain a value of 1.0. This measure for demographic parity is also called "disparate impact" (see e.g. [11,33]). As the results in Table 1 show, both FairGP variants are clearly fairer than the baseline GP. We use the mean (PR_t^{avg}) for the target acceptance rate. In Fig. 2, we investigate which choice of target $(PR_t^{avg}, PR_t^{min} \text{ or } PR_t^{max})$ gives the best result. The Fig.2(a) shows results from Adult dataset with *race* as sensitive attribute where we have $PR_t^{min} = 0.156$, $PR_t^{max} = 0.267$ and $PR_t^{avg} = 0.211$. PR_t^{avg} performs equal (for *race* as sensitive attribute) or better (for *gender*) compared with the two other possibilities. The variant that uses s for training and prediction ("use s") performs significantly better here. *ZafarAccuracy* can achieve good fairness results at the cost of accuracy. The results of FairGP are characterized by high fairness and high accuracy. FairLR achieves similar results to FairGP, but with generally slightly lower accuracy but better fairness. We used

Figure 4: Accuracy and fairness (with respect to *equality of opportunity*) for various methods on ProPublica dataset. **(a)**: using race as the sensitive attribute; **(b)**: using gender. A completely fair model would achieve a value of 1.0 in the x-axis. See Fig. 5(a) and (b) on how these choices of TPR setting translate to $TPR_{s=0}$ vs $TPR_{s=1}$.

the two step procedure of Donini et al. [10] to verify that we cannot achieve the same fairness result of FairLR with just parameter search on LR.

Fig. 3(a) and (b) show runs of FairGP where we explicitly set a target acceptance rate, $PR_t := \mathbb{P}(\bar{y} = 1)$, instead of taking the mean PR_t^{avg}. A perfect targeting mechanism would produce a diagonal. The data points are not exactly on the diagonal but they show that setting the target rate has the expected effect on the observed acceptance rate. This tuning of the target rate is the unique aspect of the approach. This would be very difficult to achieve with existing fairness methods; a new constraint would have to be added. Fig. 3(c) and (d) show the same data as Fig. 3(a) and (b) but with different axes. It can be seen from from this Fig. 3(a) and (b) that the target acceptance rate can be used to *control* the trade-off between accuracy and fairness. In this specific case, changing the target rate barely affects fairness and it only affects the accuracy because target acceptance rates that are different from the base acceptance rate necessarily lead to "missclassifications".

Results for Equality of Opportunity on ProPublica dataset. For equality of opportunity, we again follow Zafar et al. [33] and evaluate the algorithm on the ProPublica dataset. As we did for demographic parity, we define a measure of equality of opportunity via the ratio of the true positive rates (TPRs) within the demographic groups. We use $TPR_{s=i}$ to denote the observed TPR in group i: $\mathbb{P}(\hat{y} = 1 | y = 1, s = i)$, and $TNR_{s=i}$ for the observed true negative rate (TNR) in the same manner. The measure is then given by $TPR_{s=0}/TPR_{s=1}$. A perfectly fair algorithm would achieve 1.0 on the measure. In order to get the target TNR (TNR_t), we run the baseline GP with and without s 10 times at first, and obtain the means of $TNR_{s=0}$ and $TNR_{s=1}$ for both with and without s as input. As described in Section 3.2, we take the minimum of the TNRs as the target TNR. We tried other target TNRs as well with similar results.

In order to demonstrate the *tuning* aspect of our proposed framework, we set different target TPRs: $0.6, 0.7, 0.8, 0.9, 1.0$. The target value of 0.6 corresponds
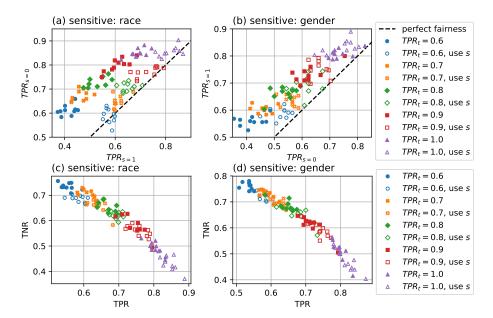
Figure 5: Predictions with different target true positive rates (TPR_t; equality of opportunity) on ProPublica dataset. **(a)** and **(b)**: $TPR_{s=0}$ vs $TPR_{s=1}$. **(c)** and **(d)**: TPR vs TNR. Left column: using race as the sensitive attribute; Right column: using gender.

approximately to the maximum of the two TPRs. Therefore, this value would be the default value according to the procedure described in Section 3.2. The results of 10 runs are shown in Fig. 4. Here, for the race attribute, we see that *FairGP* with $TPR_t = 0.6$ using s for predictions performs best in terms of the TPR ratio. For the gender attribute, most of our *FairGP* models with and without s as input (except $TPR_t = 1$) achieve better fairness than *ZafarEqOpp* and the other baselines, with little to no accuracy drop. Furthermore, although our *FairGP* cannot always outperform *ZafarEqOpp*, we can achieve significantly higher TPRs for each group ($TPR_{s=0}$, $TPR_{s=1}$) and the whole test dataset (TPR), which is discussed next based on the visualization in Fig. 5(a) and (b).

It can be seen that higher target TPRs lead to higher $TPR_{s=0}$, $TPR_{s=1}$, and TPR. Fig. 5(a) and (b) show the actual TPRs for the two groups ($s = 0$ and $s = 1$) setting several TPR targets. As a comparison, *ZafarEqOpp* achieves $TPR_{s=0} = 0.574$ and $TPR_{s=1} = 0.505$ with *race* as sensitive attribute and $TPR_{s=0} = 0.421$ and $TPR_{s=1} = 0.563$ with *gender*. By setting the target TPRs higher, we can easily improve our actual TPRs for each group. Fig. 5(c) and (d) show the same data with different axes (TPR and TNR). A clear trend is that setting higher target TPRs will lead to the higher actual TPRs and lower actual TNRs. Thus, the target TPR does *control* the trade-off between accuracy and fairness. Fig. 5(c) and (d) also illustrate the trade-off between TPR and TNR of our algorithm, which was discussed in Section 3.2.

6   Discussion and conclusion

We have developed a machine learning framework which allows us to set a *target rate* for a variety of fairness notions, including demographic parity, equalized odds, and equality of opportunity. For example, we can set the target true positive rate for equality of opportunity to be 0.6 for different groups. This capability is unique to our approach and can be used as an intuitive mechanism to control the trade-off between fairness and accuracy. In contrast to other methods which rely on unintuitive parameters, such as covariance thresholds, to enforce fairness, our method enables more control over tuning fairness, by introducing control parameters that are understandable to the general public: true positive rates, false positive rates, and positive rates. Our framework is general and will be applicable for sensitive variables with binary and multi-level values. The current work focuses on a single binary sensitive variable. For future work, we plan to extend the framework to allow the target labels to directly depend on input features. We can then model more complex biases in the labels and the inputs.

References

1. Agarwal, A., Beygelzimer, A., Dudik, M., Langford, J., Wallach, H.: A reductions approach to fair classification. In: ICML. vol. 80, pp. 60–69 (2018)
2. Angwin, J., Larson, J., Mattu, S., Kirchner, L.: Machine bias. ProPublica **23** (2016)
3. Barocas, S., Selbst, A.D.: Big data's disparate impact. California Law Review **104**, 671–732 (2016)
4. Bonilla, E.V., Krauth, K., Dezfouli, A.: Generic inference in latent gaussian process models. arXiv:1609.00577 (2016)
5. Calders, T., Kamiran, F., Pechenizkiy, M.: Building classifiers with independency constraints. In: ICDMW. pp. 13–18. IEEE (2009)
6. Chouldechova, A.: Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. Big data **5**(2), 153–163 (2017)
7. Cotter, A., Jiang, H., Wang, S., Narayan, T., Gupta, M.R., You, S., Sridharan, K.: Optimization with non-differentiable constraints with applications to fairness, recall, churn, and other goals. arXiv:1809.04198 (2018)
8. Dezfouli, A., Bonilla, E.V.: Scalable inference for Gaussian process models with black-box likelihoods. In: NIPS. pp. 1414–1422 (2015)
9. Dheeru, D., Karra Taniskidou, E.: UCI machine learning repository (2017), http://archive.ics.uci.edu/ml
10. Donini, M., Oneto, L., Ben-David, S., Shawe-Taylor, J.S., Pontil, M.: Empirical risk minimization under fairness constraints. In: NeurIPS. pp. 2796–2806 (2018)
11. Feldman, M., Friedler, S.A., Moeller, J., Scheidegger, C., Venkatasubramanian, S.: Certifying and removing disparate impact. In: SIGKDD. pp. 259–268. ACM (2015)

12. Friedler, S.A., Scheidegger, C., Venkatasubramanian, S., Choudhary, S., Hamilton, E.P., Roth, D.: A comparative study of fairness-enhancing interventions in machine learning. arXiv:1802.04422 (2018)
13. Gardner, J.R., Pleiss, G., Bindel, D., Weinberger, K.Q., Wilson, A.G.: GPyTorch: Blackbox matrix-matrix gaussian process inference with GPU acceleration. In: NeurIPS. pp. 7587–7597 (2018)
14. Goh, G., Cotter, A., Gupta, M., Friedlander, M.P.: Satisfying real-world goals with dataset constraints. In: NIPS. pp. 2415–2423 (2016)
15. Hardt, M., Price, E., Srebro, N., et al.: Equality of opportunity in supervised learning. In: NIPS. pp. 3315–3323 (2016)
16. Hensman, J., Matthews, A., Ghahramani, Z.: Scalable variational Gaussian process classification. In: AISTATS (2015)
17. Hernández-Lobato, J.M., Li, Y., Rowland, M., Bui, T.D., Hernández-Lobato, D., Turner, R.E.: Black-box alpha divergence minimization. In: ICML (2016)
18. Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. Machine Learning **37**(2), 183–233 (1999)
19. Kamiran, F., Calders, T.: Data preprocessing techniques for classification without discrimination. Knowledge and Information Systems **33**, 1–33 (Oct 2012)
20. Kamishima, T., Akaho, S., Asoh, H., Sakuma, J.: Fairness-aware classifier with prejudice remover regularizer. In: ECML PKDD. pp. 35–50. Springer (2012)
21. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
22. Kleinberg, J., Mullainathan, S., Raghavan, M.: Inherent trade-offs in the fair determination of risk scores. arXiv:1609.05807 (2016)
23. Krauth, K., Bonilla, E.V., Cutajar, K., Filippone, M.: Autogp: Exploring the capabilities and limitations of gaussian process models. arXiv:1610.05392 (2016)
24. Lanckriet, G.R., Sriperumbudur, B.K.: On the convergence of the concave-convex procedure. In: NIPS. pp. 1759–1767 (2009)
25. Louizos, C., Swersky, K., Li, Y., Welling, M., Zemel, R.: The variational fair autoencoder. In: ICLR (2016)
26. Lum, K., Johndrow, J.: A statistical framework for fair predictive algorithms. arXiv:1610.08077 (2016)
27. Madras, D., Creager, E., Pitassi, T., Zemel, R.S.: Learning adversarially fair and transferable representations. In: ICML. pp. 3381–3390 (2018)
28. Quadrianto, N., Sharmanska, V.: Recycling privileged learning and distribution matching for fairness. In: NIPS. pp. 677–688 (2017)
29. Quadrianto, N., Sharmanska, V., Thomas, O.: Discovering fair representations in the data domain. In: CVPR (2019)
30. Rasmussen, C.E., Williams, C.K.: Gaussian process for machine learning. MIT press (2006)
31. Wilson, A.G., Hu, Z., Salakhutdinov, R.R., Xing, E.P.: Stochastic variational deep kernel learning. In: NIPS. pp. 2586–2594 (2016)
32. Woodworth, B., Gunasekar, S., Ohannessian, M.I., Srebro, N.: Learning non-discriminatory predictors. In: COLT. vol. 65, pp. 1920–1953. PMLR, Amsterdam, Netherlands (Jul 2017)
33. Zafar, M.B., Valera, I., Gomez Rodriguez, M., Gummadi, K.P.: Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In: WWW. pp. 1171–1180 (2017)
34. Zafar, M.B., Valera, I., Rogriguez, M.G., Gummadi, K.P.: Fairness constraints: Mechanisms for fair classification. In: AISTATS. pp. 962–970 (2017)
35. Zemel, R., Wu, Y., Swersky, K., Pitassi, T., Dwork, C.: Learning fair representations. In: ICML. pp. 325–333 (2013)