

# Fair Classification with Noisy Protected Attributes: A Framework with Provable Guarantees

L. Elisa Celis  
Yale University

Lingxiao Huang  
Huawei

Vijay Keswani  
Yale University

Nisheeth K. Vishnoi  
Yale University

October 15, 2020

## Abstract

Due to the deployment of classification algorithms in a multitude of applications directly and indirectly affecting people and society, developing methods that are fair with respect to protected attributes such as gender or race is crucial. However, protected attributes in datasets may be inaccurate due to noise in the data collection or if the protected attributes are imputed either in whole or in part. Such inaccuracies can prevent existing fair classification algorithms from achieving their claimed fairness guarantees. Motivated by this, recent works have studied the fair classification problem in which a binary protected attribute is “noisy” (the protected type is flipped with a known fixed probability) by either suggesting optimization using tighter statistical or equalized odds constraints to counter the noise [39] or by identifying conditions under which prior equalized odds post-processing algorithms can handle noisy attributes [3].

We extend the study of noise-tolerant fair classification to a very general setting. Our main contribution is an optimization framework for learning a fair classifier in the presence of noisy perturbations in the protected attributes that can be employed with linear and linear-fractional class of fairness constraints, comes with probabilistic guarantees on accuracy and fairness, and can handle multiple, non-binary protected attributes. The technical novelty of our approach lies in the fact that we capture the range of alteration in the likelihood of a classifier prediction for different protected attribute values due to noise, and use it to appropriately modify the given fairness constraints. These constraints allow the optimal fair classifier to be feasible for our modified framework as well, leading to accuracy guarantees, and ensure that classifiers which considerably violate the desired fairness guarantees do not satisfy the modified constraints, leading to fairness guarantees. Empirically, we show that our framework can be used to attain either statistical rate or false positive rate fairness guarantees with a minimal loss in accuracy, even when the noise corruption is large in two real-world datasets. Prior existing noisy fair classification approaches [3, 39], on the other hand, either do not always achieve the desired fairness levels or suffer a larger loss in accuracy for guaranteeing high fairness compared to our framework.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Other related work</b>	<b>5</b>
<b>3</b>	<b>The model</b>	<b>6</b>
<b>4</b>	<b>Framework and Theoretical results</b>	<b>7</b>
4.1	Our optimization framework . . . . .	7
4.2	Performance of Program <b>DenoisedFair</b> . . . . .	8
4.3	Estimation errors . . . . .	9
4.4	Proof of Theorem 4.3 . . . . .	9
<b>5</b>	<b>Extension to general <math>p \geq 2</math> and multiple fairness constraints</b>	<b>11</b>
<b>6</b>	<b>Proof of Lemmas 4.9, 4.10</b>	<b>15</b>
6.1	Proof of Lemma 4.9 - Relation between Program TargetFair and DenoisedFair . . . . .	15
6.2	Proof of Lemma 4.10 - Bad classifiers are not feasible for Program DenoisedFair . . . . .	17
<b>7</b>	<b>Empirical results</b>	<b>19</b>
<b>8</b>	<b>Conclusion, limitations and future work</b>	<b>21</b>
<b>A</b>	<b>Analysis of the influences of estimation errors for <math>\eta_0</math> and <math>\eta_1</math></b>	<b>26</b>
<b>B</b>	<b>Comparison with theoretical guarantees of [3]</b>	<b>27</b>
<b>C</b>	<b>Discussion of initial attempts</b>	<b>27</b>
C.1	Randomized labeling . . . . .	27
C.2	Replacing $S$ by $\hat{S}$ in Program TargetFair . . . . .	28
<b>D</b>	<b>Other empirical details and results</b>	<b>34</b>
D.1	Implementation of our denoised algorithm. . . . .	34
D.2	Other results . . . . .	36
D.2.1	Variation of noise parameter . . . . .	36
D.2.2	Error in noise parameter estimation . . . . .	38

# 1 Introduction

Fair classification has been a topic of intense study in machine learning due to the growing importance of addressing social biases in automated prediction. Consequently, a host of fair classification algorithms have been proposed that learn from data; see [6]. Most of these fair classification algorithms crucially assume that one has access to the protected attributes (e.g., race, gender) for training and/or deployment. Data collection, however, is a complex process and may contain recording and reporting errors, unintentional or otherwise [50]. Cleaning the data also requires making difficult and political decisions along the way, yet is often necessary especially when it comes to questions of race, gender, or identity [46]. Further, information about protected attributes may be missing entirely [17], something that has also been recently brought into the public eye when attempting to measure COVID19 health disparities [4]. In such cases, protected attributes can be predicted from other data, however, we know that this process is itself contains errors and biases [45, 10]. All of the above scenarios cause a significant problem for fair classification as most approaches implicitly assume perfect protected attribute data and may not achieve the same performance on fairness metrics as they would if the data was perfect. Thus, for fair classification techniques to be effective in the above-mentioned cases, they must take protected attribute errors into consideration.

Recent approaches towards fair classification in the presence of errors or noise in protected attributes have either (a) formulated a modified constrained optimization problem to account for the errors in the attributes [39], or (b) analyzed the efficacy of existing fair classification approaches in the noisy setting [3].

Lamy et al. [39] study the setting where the noise in the binary protected attribute follows a mutually contaminated model [52]; the setting of “flipping noises” where a (binary) protected type  $Z = z$  may be flipped to  $\hat{Z} = 1 - z$  with some known fixed probability  $\eta_z$  is an important example of this mutually contaminated model [42]. They formulated an optimization problem that minimizes a standard loss function subject to statistical rate (SR) or equalized odds constraints [18] and, to counter the noise in the protected attribute, the fairness constraints are modified to be *tighter* than their uncorrupted counterparts. However, there is a tradeoff between fairness and accuracy, and tighter fairness constraints can lead to an increased prediction error [43]. While [39] provide a fairness guarantee on the learned classifier, they do not discuss the impact of the tighter constraints on the prediction error and, hence, do not have a guarantee on the accuracy of the classifier.

Awasthi et al. [3] study the performance of the equalized odds post-processing method of Hardt et al. [29] in the setting of noisy binary protected attribute. The noise in their model manifests itself in the form of incorrect estimates of the joint ( $\Pr[f, Y, \hat{Z}]$ ) and conditional ( $\Pr[f | Y, \hat{Z}]$ ) probabilities of classifier predictions  $f$  given class label  $Y$  and noisy protected attribute  $\hat{Z}$ ; once again “flipping noises” can cause such corruption. Their primary contribution is the characterization of the conditions on this noise in training data samples and predictions under which the bias of a classifier learned using the method of [29] is reduced even when using the noisy protected attribute; they further show that, under these conditions, the loss in accuracy can also be bounded. Awasthi et al. [3] exhibit the robustness of the post-processing method, but there are drawbacks that limit its applicability in the noisy setting. Firstly, the fairness guarantee of [3] assures that the post-processed classifier is relatively more fair than the original, but it is not apparent if it can be used to achieve any level of user-desired fairness. Secondly, the protected attributes of only the training samples are assumed to be corrupted, and that test/future samples have uncorrupted protected attributes; this assumption rules out the real-world settings where train, test, and future data arise from the same corrupted source, for example, erroneous protected attribute prediction models [45].

Furthermore, [3, 39] primarily work with SR and/or equalized odds metrics for binary protected attributes, and it is unclear how to extend their results to the important class of linear-fractional fairness metrics [13] (e.g., false discovery rate which is employed when there are large costs associated with positive classification) and to non-binary protected attributes.

Table 1: Comparison of our paper with prior work with respect to types of protected attributes, fairness constraints, and theoretical guarantees. Types of fairness constraints are defined in Defn 5.1. “SR/FP/FN/TP/TN/ACC” represents statistical/false positive/false negative/true positive/true negative/accuracy rates respectively, “EO” represents equalized odds and “FD/FO/PP/NP” represents false discovery/false omission/positive predictive/negative predictive rates respectively.  $\checkmark$  indicates that the paper satisfies that property,  $\star$  indicates that the method in the paper can be used to satisfy the property, but is not explicitly discussed, and  $\bullet$  indicates that the property is satisfied under certain conditions. Existing works [39, 3] consider a binary protected attribute together with linear fairness constraints. [39] also do not provide accuracy guarantees, while [3] provide accuracy guarantees under certain specific conditions, but cannot handle noise in test samples. In contrast, our algorithm can handle both both linear and linear-fractional fairness constraints, and provides both accuracy and fairness guarantees.

	Protected attributes			Fairness constraints (Definition 5.1)											Theoretical guarantees	
	mul- tiple	non- binary	noise in test	Linear							Linear-fractional				acc- uracy	fair- ness
[39]	$\star$		$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\star$	$\checkmark$						$\checkmark$
[3]					$\checkmark$	$\star$	$\checkmark$	$\star$		$\checkmark$					$\bullet$	$\checkmark$
Ours	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$

**Our contributions.** We present a general optimization framework for learning a fair classifier that can handle:

- **flipping noises** (Definition 3.1,5.3) in the train, test, and future samples, wherein each protected type is switched to another with some probability (for example, a binary protected attribute type  $Z = 0$  ( $Z = 1$ ) may be observed with type  $\hat{Z} = 1$  ( $\hat{Z} = 0$ ) with a known fixed probability  $\eta_0$  ( $\eta_1$ )),
- **multiple, non-binary** protected attributes,
- **multiple** fairness metrics, including statistical parity, equalized odds, false discovery parity, and the general class of “linear-fractional” fairness constraints [13, Table 1].

For the above settings, our framework can learn a near-optimal fair classifier on the underlying dataset with high probability and comes with provable guarantees on both accuracy and fairness.

We start with the problem of noisy fair classification with respect to SR fairness metric and introduce *denoised* constraints to achieve the desired SR guarantees while taking into account the noise in the protected attribute (Program DenoisedFair). The desired SR is governed using an input parameter  $\tau \in [0, 1]$ . An optimizer  $f^\Delta$  of our program is provably approximately optimal and fair on the underlying dataset (Theorem 4.3) with high probability under certain mild assumptions that an optimizer  $f^*$  of the underlying program (Program TargetFair) has a non-trivial lower bound for positive classification on both  $Z = 0$  and  $Z = 1$  (Assumption 1). The technical novelty of the constraints in our program is that they capture the range of alteration in the probability of any classifier prediction for different protected attribute types due to flipping noises and, consequently, allow us to provide guarantees on  $f^\Delta$  (Theorem 4.3). The guarantee on accuracy uses the fact that an optimal fair classifier  $f^*$  for the underlying uncorrupted dataset is *likely* to be feasible for Program DenoisedFair as well, which ensures that the empirical risk of  $f^\Delta$  is less than  $f^*$  (Lemma 4.9). The guarantee on fairness of  $f^\Delta$  is attained by arguing that classifiers which considerably violate the desired fairness guarantee are infeasible for Program DenoisedFair with high probability (Lemma 4.10).

Subsequently, we extend our framework to handle multiple, non-binary protected attributes, arbitrary flipping noises, and multiple linear or linear-fractional class of fairness constraints (Program Gen-DenoisedFair), with provable guarantees (Theorem 5.9).

The difference between the constrained program of [39] and our approach is that [39] down-scale the “fairness tolerance” parameter in the constraints to adjust for the noise (the scaling is computed as a pre-processing step), while our framework adapts the fairness metric over the noisy attribute in the constraints so that it reflects the true metric in the uncorrupted setting. Due to this difference, unlike [39], our approach can handle linear-fractional metrics (which measure the performance disparity across the protected types conditioned on the classifier prediction) and non-binary attributes. For linear-fractional metrics, the scaling parameter in [39] cannot be computed in the pre-processing step since the scaling depends on the conditional event, which is a function of the classifier prediction in this case. Our approach, instead, estimates the altered

form of the linear-fractional fairness metrics in the noisy setting and uses this to form the constraints for these metrics. Since we show how to alter the general class of fairness metrics considered in [13], our framework can handle multiple, non-binary protected attributes as well; it is unclear whether the scaling method of [39] can be employed for noise in non-binary protected attributes, and [3] do not provide extensions of their conditions under which post-processing [29] reduces bias even for non-binary protected attributes.

We implement our framework using logistic loss function [22] and examine it on **Adult** and **COMPAS** datasets (Section 7). We consider sex and race as the protected attribute and generate noisy datasets with respect to these attributes with varying flipping noise parameters. We use SR and false positive rate fairness metrics, and compare against natural baselines and existing noise-tolerant fair classification algorithms **LZMV** [39] and **AKM** [3]. The empirical results show that, for all combinations of dataset and protected attribute, our framework attains better fairness than an unconstrained classifier, with a minimal loss in accuracy. The fairness-accuracy tradeoff of our framework is also better than the baselines, **LZMV** and **AKM**, in most cases, which either do not always achieve high fairness levels or suffer a larger loss in accuracy for achieving high fairness levels compared to our framework. For instance, for the **Adult** dataset with sex as the protected attribute, the unconstrained classifier has SR 0.31 and accuracy 0.80; our framework in this setting attains SR close to 0.89, with an accuracy of 0.77; the baseline **AKM** achieves accuracy similar to ours (0.77), but low SR (0.66), while **LZMV** attains high SR ( $\geq 0.90$ ) at a much lower accuracy ( $\leq 0.67$ ) than ours. By varying  $\tau$ , we also present the impact of the desired fairness guarantees on the accuracy of our framework.

## 2 Other related work

**Fair classification.** A large body of works have focused on formulating fair classification problems as constrained optimization problems, e.g., constrained to statistical parity [57, 58, 44, 24, 13], or equalized odds [29, 56, 44, 13], and developing algorithms for it. Another class of algorithms for fair classification first learn an unconstrained optimal classifier and then shift the decision boundary according to the fairness requirement, e.g., [21, 29, 25, 49, 55, 20]. Interested readers can see a summary and comparisons of existing fair classification algorithms in [23, 5]. In contrast to this work, the assumption in all of these approaches is that the algorithm is given perfect information about the protected class.

**Data correction.** There has been significant effort to suitably encode and/or correct datasets to remove potential biases and inaccuracies. Cleaning raw data is a significant step in the pipeline, and efforts to correct for missing or inaccurately coded attributes have been studied in-depth for protected attributes, e.g., in the context of the census [46]. An alternate approach considers changing the composition of the dataset itself to correct for known biases in representation, and popular methods include re-labeling/re-weighting approach of [12, 36, 37], the repair methods of [26, 53], or optimization based methods such as [19, 14]. In either case, the correction process, while important, can be imperfect and our work can help by starting with these improved yet imperfect datasets in order to build fair classifiers.

**Unknown protected attributes.** A related setting to ours is when the information of some protected attributes is unknown. [27, 16, 35] considered this setting of unknown protected attributes and designed algorithms to improve fairness or assess disparity. In contrast, our approach aims to derive necessary information from the observed protected attributes to design alternate fairness constraints using the noisy attribute.

**Classifiers robust to the choice of datasets.** Recent studies have also pointed to the brittleness of fair classification algorithms under the noisy setting. For instance, [23] observed that fair classification algorithms may not be stable with respect to variations in the training dataset. [30] proved that empirical risk minimization amplifies representation disparity over time. Towards this, certain variance reduction or stability techniques have been introduced; see e.g., [34] who investigate how to achieve a fair classifier that is also stable with respect to variation in datasets. However, their approach cannot be used to learn a classifier that is provably fair over the underlying dataset.

**Noise in labels.** Recently, there have been some works [9, 8] that study fair classification when the label in the input dataset is noisy. The main difference of [9, 8] from our work is that they consider noisy labels instead of noisy protected attributes, which makes our denoised algorithms very different since the accuracy of protected attributes mainly relates to the fairness of the classifier but the accuracy of labels primarily affect to the empirical loss.

### 3 The model

Let  $\mathcal{D} = \mathcal{X} \times [p] \times \{0, 1\}$  denote the underlying domain. Each sample  $(X, Z, Y)$  drawn from  $\mathcal{D}$  contains a protected attribute  $Z \in [p]^1$ , a class label  $Y \in \{0, 1\}$  that we want to predict, and non-protected features  $X \in \mathcal{X}$ . We will assume that  $X$  is a  $d$ -dimensional vector, for a given  $d \in \mathbb{N}$ , i.e.,  $\mathcal{X} \subseteq \mathbb{R}^d$ . Let  $S = \{s_i = (x_i, z_i, y_i) \in \mathcal{D}\}_{i \in [N]}$  be the (underlying, uncorrupted) dataset. Let  $\mathcal{F} \subseteq \{0, 1\}^{\mathcal{X}}$  denote a family of all possible classifiers. Given a loss function  $L : \mathcal{F} \times \mathcal{D} \rightarrow \mathbb{R}$  and parameter  $\tau \in [0, 1]$ , the goal is to learn a classifier that minimizes:

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i \in [N]} L(f, s_i) \quad \text{s.t.} \quad \Omega(f, S) \geq \tau. \quad (\text{TargetFair})$$

Here  $\Omega : \mathcal{F} \times \mathcal{D}^* \rightarrow \mathbb{R}$  corresponds to a specific fairness metric, e.g., statistical rate [57, 44, 24, 13], or false positive/negative rate [29, 56, 44, 13], and  $\tau$  represents the desired fairness guarantee. For instance, let  $D$  denote the empirical distribution over  $S$ ; then the statistical rate of a classifier  $f$  on  $S$  defined as

$$\gamma(f, S) := \frac{\min_{l \in [p]} \Pr_D[f = 1 \mid Z = l]}{\max_{l \in [p]} \Pr_D[f = 1 \mid Z = l]}. \quad (1)$$

A sample fairness constraint could be  $\gamma(f, S) \geq 0.8$  (inspired by the 80% disparate impact rule [7]). Since  $\gamma(f, S) \geq 0.8$  is non-convex, often in the literature, one considers a convex function  $\Omega(f, S)$  as an estimate of  $\gamma(f, S)$ , e.g.,  $\Omega(f, S)$  is formulated as a covariance-type function in [57], and as the weighted sum of the logs of the empirical estimate of favorable bias in [24].

If  $S$  is observed, we can directly solve Program TargetFair. However, as discussed earlier, the protected attributes in  $S$  may be imperfect and we may only observe a noisy dataset  $\hat{S}$  instead of  $S$ . The noise model on the protected attributes considered in this paper (see also [39, 3, 54, 48, 9]) is presented below. For simplicity, we first consider the case of a binary protected attribute ( $p = 2$ ), and later generalize it to non-binary protected attributes (Section 5).

**Definition 3.1 (Flipping noises for a binary protected attribute)** Suppose  $p = 2$ , i.e.,  $Z \in \{0, 1\}$ . Let  $\eta_0, \eta_1 \in (0, 0.5)$  be noise parameters. For each  $i \in [N]$ , we assume that the  $i$ th noisy sample  $\hat{s}_i = (x_i, \hat{z}_i, y_i)$  is realized as follows:

- If  $z_i = 0$ , then  $\hat{z}_i = 0$  with probability  $1 - \eta_0$  and  $\hat{z}_i = 1$  with probability  $\eta_0$ .
- If  $z_i = 1$ , then  $\hat{z}_i = 0$  with probability  $\eta_1$  and  $\hat{z}_i = 1$  with probability  $1 - \eta_1$ .

As  $\eta_0$  or  $\eta_1$  increase, the observed dataset  $\hat{S}$  becomes more noisy. Specifically, if  $\eta_0 = \eta_1 = 0.5$ ,  $\hat{Z} = 1$  holds with probability 0.5, we can not learn any information of  $Z$  from  $\hat{Z}$ . Due to noises  $\eta_0, \eta_1$ , directly applying the same fairness constraints on  $\hat{S}$  may introduce bias on  $S$  and, hence, modifications to the constraints are necessary; see Appendix C for discussion.

**Remark 3.2 (Limitations of Definition 3.1)** In practice, we may not know  $\eta_0$  and  $\eta_1$  explicitly, and can only estimate them by, say, finding a small appropriate sample of the data for which ground truth is known (or can be found), and computing estimates for  $\eta_0$  and  $\eta_1$  accordingly. In the following sections, we assume that  $\eta_0$  and  $\eta_1$  are given. For settings in which the estimates of  $\eta_0$  and  $\eta_1$  may not be accurate, we analyze the influences of the estimation errors in Section 4.3.

Generated noises can also be more complicated. For instance, the noise parameter may also depend on other non-protected features of the individuals. For this paper, however, we consider the simple setting of Definition 3.1 and focus on providing a provable algorithm.

<sup>1</sup>The domain  $\mathcal{D}$  can be generalized to include multiple protected attributes  $Z_1, \dots, Z_m$  where  $Z_i \in [p_i]$ . We first discuss a single protected attribute for simplicity and generalize the model and results to multiple protected attributes in Section 5.

We also make the following assumption on an optimal classifier  $f^*$  of Program TargetFair.

**Assumption 1 (Lower bound for the positive predictions of  $f^*$ )** *There exists a constant  $\lambda \in (0, 0.5)$  such that*

$$\min \left\{ \Pr_D[f^* = 1, Z = 0], \Pr_D[f^* = 1, Z = 1] \right\} \geq \lambda.$$

For instance, if there are 20% of samples with  $Z = 0$  and  $\Pr_D[f^* = 1 \mid Z = i] \geq 0.5$  ( $i \in \{0, 1\}$ ), we have  $\lambda \geq 0.1$ . In practice, exact  $\lambda$  is unknown but we can set  $\lambda$  according to the context, e.g.,  $\lambda$  can be set higher if  $\min \{\Pr_D[Y = 1, Z = 0], \Pr_D[Y = 1, Z = 1]\}$  is large. Making this assumption is not strictly necessary, i.e., we can simply set  $\lambda = 0$ , but the scale of  $\lambda$  affects the performance of our approaches; see Remark 4.4.<sup>2</sup> Given the above setup, the formal problem of fair classification can be stated as follows.

**Problem 1 (Fair classification with noisy protected attributes)** *Given a binary protected attribute ( $p = 2$ ), a fairness constraint of the form  $\gamma(f, S) \geq \tau$ , a noisy dataset  $\hat{S}$  drawn from the underlying dataset  $S$  with flipping noise parameters  $\eta_0, \eta_1 \in (0, 0.5)$ , and  $\lambda \in (0, 0.5)$  for which Assumption 1 holds, the goal is to learn an (approximately) optimal fair classifier  $f \in \mathcal{F}$  of Program TargetFair.*

## 4 Framework and Theoretical results

In this section, we tackle Problem 2 for statistical rate constraints and state our optimization framework (Program TargetFair) for this setting as an example. Extension of the problem, assumptions, and framework to multiple, non-binary protected attributes and other kinds of fairness constraints is provided in Section 5.

The main difficulty in solving Problem 1 is to satisfy the fairness constraints when  $S$  is unknown. Our key idea is to design new constraints over  $\hat{S}$  that estimate the underlying fairness constraints of Program TargetFair.

### 4.1 Our optimization framework

Let  $\hat{D}$  denote the empirical distribution over  $\hat{S}$  and let  $\pi_{ij} := \Pr_{D, \hat{D}}[\hat{Z} = i \mid Z = j]$  for  $i, j \in \{0, 1\}$ ,  $\mu_i := \Pr_D[Z = i]$  and  $\hat{\mu}_i := \Pr_{\hat{D}}[\hat{Z} = i]$  for  $i \in \{0, 1\}$ . If  $D$  and  $\hat{D}$  are clear from the context, we denote  $\Pr_{D, \hat{D}}[\cdot]$  by  $\Pr[\cdot]$ . By Definition 3.1, we can estimate probabilities  $\pi_{ij}$  using the following observations:  $\mathbb{E}_{\hat{S}}[\pi_{10}] = \eta_0$  and  $\mathbb{E}_{\hat{S}}[\pi_{01}] = \eta_1$ , which helps us design the following denoised fairness constraints.

**Definition 4.1 (Denoised fairness constraints)** *Given a noisy dataset  $\hat{S}$  and a classifier  $f \in \{0, 1\}^{\mathcal{X}}$ , let*

$$\Gamma_0(f) := \frac{(1 - \eta_1) \Pr[f = 1, \hat{Z} = 0] - \eta_1 \Pr[f = 1, \hat{Z} = 1]}{(1 - \eta_1)\hat{\mu}_0 - \eta_1\hat{\mu}_1}$$

and

$$\Gamma_1(f) := \frac{(1 - \eta_0) \Pr[f = 1, \hat{Z} = 1] - \eta_0 \Pr[f = 1, \hat{Z} = 0]}{(1 - \eta_0)\hat{\mu}_1 - \eta_0\hat{\mu}_0}.$$

We define the denoised statistical rate to be  $\gamma^\Delta(f, \hat{S}) := \min \left\{ \frac{\Gamma_0(f)}{\Gamma_1(f)}, \frac{\Gamma_1(f)}{\Gamma_0(f)} \right\}$ , and define our denoised fairness constraints to be

$$\begin{cases} (1 - \eta_1) \Pr[f = 1, \hat{Z} = 0] - \eta_1 \Pr[f = 1, \hat{Z} = 1] \geq (1 - \eta_0 - \eta_1)\lambda - \delta, \\ (1 - \eta_0) \Pr[f = 1, \hat{Z} = 1] - \eta_0 \Pr[f = 1, \hat{Z} = 0] \geq (1 - \eta_0 - \eta_1)\lambda - \delta, \\ \gamma^\Delta(f, \hat{S}) \geq \tau - \delta, \end{cases} \quad (2)$$

<sup>2</sup>This assumption is for classification with statistical rate fairness metric and generalization of this assumption for broader fairness metrics is presented in Section 5.

where  $\delta \in (0, 1)$  is a fixed constant and  $\tau \in [0, 1]$  is the desired lower bound on statistical rate. Our denoised program is as follows:

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i \in [N]} L(f, \hat{s}_i) \quad \text{s.t.} \quad \text{Constraints (2)}. \quad (\text{DenoisedFair})$$

The  $\delta$  in Constraint (2) is used as a relaxation parameter depending on the context. Intuitively,  $\Gamma_i(f)$  is designed to estimate  $\Pr[f = 1 \mid Z = i]$  for  $i \in \{0, 1\}$ : its numerator approximates  $(1 - \eta_0 - \eta_1) \Pr[f = 1, Z = i]$  and its denominator approximates  $(1 - \eta_0 - \eta_1) \mu_i$ . For the denominator, since  $\Pr[\hat{Z} = 1 \mid Z = 0] \approx \eta_0$ , we can represent  $\mu_i$  ( $i \in \{0, 1\}$ ) by a linear combination of  $\hat{\mu}_0$  and  $\hat{\mu}_1$ . Similar intuition is behind the representation of the numerator.

Due to how  $\Gamma_0$  and  $\Gamma_1$  are chosen, the first two constraints are designed to estimate the constraint  $\min\{\Pr_D[f = 1, Z = 0], \Pr_D[f = 1, Z = 1]\} \geq \lambda$ , so as to satisfy Assumption 1, and the last constraint is designed to estimate  $\gamma(f, S) \geq \tau$  by the definition of  $\gamma$ .

## 4.2 Performance of Program DenoisedFair

Our main theorem shows that solving Program DenoisedFair leads to a classifier which does not increase the empirical risk (compared to the optimal fair classifier) and only slightly violates the fairness constraint. To state our result, we need the following definition that measures the complexity of  $\mathcal{F}$ .

**Definition 4.2 (VC-dimension of  $(S, \mathcal{F})$  [28])** Given a subset  $A \subseteq [N]$ , we define

$$\mathcal{F}_A := \{f \in \mathcal{F} : f(s_i) = 1 \mid i \in A\}$$

to be the collection of subsets of  $A$  that may be shattered by some  $f \in \mathcal{F}$ . The VC-dimension of  $(S, \mathcal{F})$  is the largest integer  $t$  such that there exists a subset  $A \subseteq [N]$  with  $|A| = t$  and  $|\mathcal{F}_A| = 2^t$ .

Suppose  $\mathcal{X} \subseteq \mathbb{R}^d$  for some integer  $d \geq 1$ . If  $\mathcal{F} = \{0, 1\}^{\mathcal{X}}$ , we observe that the VC-dimension is  $t = N$ . Several commonly used families  $\mathcal{F}$  have VC-dimension  $O(d)$ , including linear threshold functions [28], kernel SVM and gap tolerant classifiers [11]. Using this definition, the main theorem in this paper is as follows.

**Theorem 4.3 (Performance of Program DenoisedFair)** Suppose the VC-dimension of  $(S, \mathcal{F})$  is  $t$ . Given any parameters  $\eta_0, \eta_1, \lambda \in (0, 0.5)$  and  $\delta \in (0, 1)$ , let  $f^\Delta \in \mathcal{F}$  denote an optimal solution of Program DenoisedFair. With probability at least

$$1 - O\left(e^{-\frac{(1-\eta_0-\eta_1)^2 \lambda^2 \delta^2 n}{5000}} + t \ln\left(\frac{50}{(1-\eta_0-\eta_1)\lambda\delta}\right)\right),$$

we have

- $\frac{1}{N} \sum_{i \in [N]} L(f^\Delta, s_i) \leq \frac{1}{N} \sum_{i \in [N]} L(f^*, s_i);$
- $\gamma(f^\Delta, S) \geq \tau - 3\delta.$

**Remark 4.4 (Analysis of parameters that affect the performance)** Observe that the success probability depends on  $1 - \eta_0 - \eta_1$ ,  $\delta$ ,  $\lambda$  and the VC-dimension  $t$  of  $(S, \mathcal{F})$ . If  $1 - \eta_0 - \eta_1$  or  $\delta$  is close to 0, i.e., the protected attributes are very noisy or there is no relaxation for  $\gamma(f, S) \geq \tau$  respectively, the success probability guarantee naturally tends to be 0. Next we discuss the remaining parameters  $\lambda$  and  $t$ .

**Discussion on  $\lambda$ .** Intuitively, the success probability guarantee tends to 0 when  $\lambda$  is close to 0. For instance, suppose there is only one sample  $s_1$  with  $Z = 0$  for which  $f^*(s_1) = 1$ , i.e.,  $\Pr_D[f^* = 1, Z = 0] = 1/N$  and, therefore,  $\lambda = 1/N$ . To approximate  $f^*$ , we may need to label  $f(s_1) = 1$ . However, due to the flipping noises, it is likely that we can not find out the specific sample  $s_1$  to label  $f(s_1) = 1$ , unless we let the classifier prediction be  $f = 1$  for all samples, which leads to a large empirical risk (see discussion in Section C.1). In other words, the task is tougher for smaller values of  $\lambda$ .



**Discussion on  $t$ .** The success probability also depends on  $t$  which captures the complexity of  $\mathcal{F}$ . Suppose  $\mathcal{X} \subseteq \mathbb{R}^d$  for some integer  $d \geq 1$ . The worst case is  $\mathcal{F} = \{0, 1\}^{\mathcal{X}}$  with  $t = N$ , which takes the success probability guarantee to 0. On the other hand, if the VC-dimension does not depend on  $N$ , e.g., only depends on  $d \ll N$ , the failure probability is exponentially small on  $N$ . For instance, if  $\mathcal{F}$  is the collection of all linear threshold functions, i.e., each classifier  $f \in \mathcal{F}$  has the form  $f(s_i) = \mathbf{I}[\langle x_i, \theta \rangle \geq r]$  for some vector  $\theta \in \mathbb{R}^d$  and threshold  $r \in \mathbb{R}$ . We have  $t \leq d + 1$  for an arbitrary dataset  $S$  [28].

### 4.3 Estimation errors

In practice, we can use prior work on noise parameter estimation [42, 41, 47] to obtain estimates of  $\eta_0$  and  $\eta_1$ , say  $\eta'_0$  and  $\eta'_1$  respectively. The scale of estimation errors also affects the performance of our denoised program. Define  $\zeta := \max\{|\eta_0 - \eta'_0|, |\eta_1 - \eta'_1|\}$  to be the additive estimation error. This factor  $\zeta$  can be used to measure the fairness loss in Theorem 4.3 due to estimation errors. Concretely, there exists some constant  $\alpha > 0$  such that  $\gamma(f^\Delta, \cdot) \geq \tau - 3\delta - \zeta\alpha$  holds. Compared to Theorem 4.3, the estimation errors introduce an additive  $\zeta\alpha$  error term for the fairness guarantee of our denoised program. The discussion on the value of  $\alpha$  can be found in Appendix A.

### 4.4 Proof of Theorem 4.3

The main idea of the proof is to verify a)  $f^*$  is a feasible solution of Program DenoisedFair, and b) any “unfair” classifier  $f \in \mathcal{F}$  violates Constraint (2). If both conditions hold, the empirical risk of  $f^\Delta$  is guaranteed to be at most that of  $f^*$  and  $f^\Delta$  must be fair over  $S$  (Theorem 4.3). The primary difficulty is that there can be a large number of unfair classifiers and we need to show the probability that all these classifiers violate Constraint (2) is close to 1. We first define the collection of classifiers that are expected to violate Constraint (2).

**Definition 4.5 (Bad classifiers)** Given a family  $\mathcal{F} \subseteq \{0, 1\}^{\mathcal{X}}$ , we call  $f \in \mathcal{F}$  a bad classifier if  $f$  belongs to at least one of the following sub-families:

- $\mathcal{G}_0 := \{f \in \mathcal{F} : \min\{\Pr[f = 1, Z = 0], \Pr[f = 1, Z = 1]\} < \frac{\lambda}{2}\};$
- Let  $T = \lceil 232 \log \log \frac{2(\tau-3\delta)}{\lambda} \rceil$ . For  $i \in [T]$ , define  $\mathcal{G}_i := \left\{f \in \mathcal{F} \setminus \mathcal{G}_0 : \gamma(f, S) \in \left[\frac{\tau-3\delta}{1.01^{2^i+1}-1}, \frac{\tau-3\delta}{1.01^{2^i-1}-1}\right]\right\}.$

Intuitively, if  $f \in \mathcal{G}_0$ , then it is likely that  $f$  violates the first or the second of Constraint (2); if  $f \in \mathcal{G}_i$  for some  $i \in [T]$ , it is likely that  $\gamma^\Delta(f, \hat{S}) < \tau - \delta$ . Thus, any bad classifier is likely to violate Constraint (2) (Lemma 4.9). We still need to lower bound the total violating probability for all bad classifiers. Towards this, we need the following definition.

**Definition 4.6 ( $\varepsilon$ -nets)** Given a family  $\mathcal{F} \subseteq \{0, 1\}^{\mathcal{X}}$  of classifiers and  $\varepsilon \in (0, 1)$ , we say  $F \subseteq \mathcal{F}$  is an  $\varepsilon$ -net of  $\mathcal{F}$  if for any  $f, f' \in F$ ,  $\Pr_D[f \neq f'] \geq \varepsilon$ ; and for any  $f \in \mathcal{F}$ , there exists  $f' \in F$  such that  $\Pr_D[f \neq f'] \leq \varepsilon$ . We denote  $M_\varepsilon(\mathcal{F})$  as the smallest size of an  $\varepsilon$ -net of  $\mathcal{F}$ .

For instance, it follows from basic coding theory [40] that  $M_\varepsilon(\{0, 1\}^{\mathcal{X}}) = \Omega(2^{N-O(\varepsilon N \log N)})$ . The size of an  $\varepsilon$ -net is usually depends exponentially on the VC-dimension.

**Theorem 4.7 (Relation between VC-dimension and  $\varepsilon$ -nets [31])** Suppose the VC-dimension of  $(S, \mathcal{F})$  is  $t$ . For any  $\varepsilon \in (0, 1)$ ,  $M_\varepsilon(\mathcal{F}) = O(\varepsilon^{-t})$ .

Next, we define the capacity of bad classifiers based on  $\varepsilon$ -nets.

**Definition 4.8 (Capacity of bad classifiers)** Let  $\varepsilon_0 = \frac{(1-\eta_0-\eta_1)\lambda-2\delta}{5}$ . Let  $\varepsilon_i = \frac{1.01^{2^i-1}\delta}{5}$  for  $i \in [T]$  where  $T = \lceil 232 \log \log \frac{2(\tau-3\delta)}{\lambda} \rceil$ . Given a family  $\mathcal{F} \subseteq \{0, 1\}^{\mathcal{X}}$ , we denote the capacity of bad classifiers by

$$\Phi(\mathcal{F}) := 2e^{-2\varepsilon_0^2 n} M_{\varepsilon_0}(\mathcal{G}_0) + 4 \sum_{i \in [T]} e^{-\frac{\varepsilon_i^2 (1-\eta_0-\eta_1)^2 \lambda^2 n}{200}} M_{\varepsilon_i (1-\eta_0-\eta_1)\lambda/10}(\mathcal{G}_i).$$

We show that  $\Phi(\mathcal{F})$  is an upper bound for the probability that there exists a bad classifier that is feasible for Program DenoisedFair (Lemma 4.10). Roughly, the factor  $2e^{-2\varepsilon_0^2 n}$  is an upper bound of the probability that a bad classifier  $f \in \mathcal{G}_0$  violates Constraint (2), and the factor  $4e^{-\varepsilon_i^2 \lambda^2 \delta^2 n}$  is an upper bound of the probability that a bad classifier  $f \in \mathcal{G}_i$  violates Constraint (2). We prove that if all bad classifiers in the nets of  $\mathcal{G}_i$  ( $0 \leq i \leq T$ ) are not feasible for Program DenoisedFair, then all bad classifiers should violate Constraint (2). Note that the scale of  $\Phi(\mathcal{F})$  depends on the size of  $\varepsilon$ -nets of  $\mathcal{F}$ , which can be upper bounded by Theorem 4.7 and leads to the success probability of Theorem 4.3.

**Lemma 4.9 (Relation between Program TargetFair and DenoisedFair)** *Let  $f \in \mathcal{F}$  be an arbitrary classifier and  $\varepsilon \in (0, 0.5)$ . With probability at least  $1 - 2e^{-2\varepsilon^2 n}$ ,*

$$\begin{aligned} (1 - \eta_1) \Pr[f = 1, \widehat{Z} = 0] - \eta_1 \Pr[f = 1, \widehat{Z} = 1] &\in (1 - \eta_0 - \eta_1) \Pr[f = 1, Z = 0] \pm \varepsilon, \\ (1 - \eta_0) \Pr[f = 1, \widehat{Z} = 1] - \eta_0 \Pr[f = 1, \widehat{Z} = 0] &\in (1 - \eta_0 - \eta_1) \Pr[f = 1, Z = 1] \pm \varepsilon. \end{aligned}$$

Moreover, if  $\min_{i \in \{0,1\}} \Pr[f = 1, Z = i] \geq \frac{\lambda}{2}$ , then with probability at least  $1 - 4e^{-\frac{\varepsilon^2(1-\eta_0-\eta_1)^2 \lambda^2 n}{200}}$ ,

$$\gamma^\Delta(f, \widehat{S}) \in (1 \pm \varepsilon) \gamma(f, S).$$

The first part of this lemma shows how to estimate  $\Pr[f = 1, Z = i]$  ( $i \in \{0, 1\}$ ) in terms of  $\Pr[f = 1, \widehat{Z} = 0]$  and  $\Pr[f = 1, \widehat{Z} = 1]$ , which motivates the first two constraints of Program DenoisedFair. The second part of the lemma motivates the last constraint of Program DenoisedFair. By Assumption 1, we have

$$\min\{\Pr[f^* = 1, Z = 0], \Pr[f^* = 1, Z = 1]\} \geq \lambda.$$

Hence, by the above lemma,  $f^*$  is likely to be feasible for Program DenoisedFair. Consequently,  $f^\Delta$  has empirical loss at most that of  $f^*$ . We provide a proof sketch here and the complete proof is presented in Section 6.1.

**Proof sketch:** The first part of Lemma 4.9 follows from the fact that for  $i \in \{0, 1\}$ ,

$$\Pr[f = 1, \widehat{Z} = i] = \Pr[\widehat{Z} = i | f = 1, Z = i] \cdot \Pr[f = 1, Z = i] + \Pr[\widehat{Z} = i | f = 1, Z = 1 - i] \cdot \Pr[f = 1, Z = 1 - i].$$

Then by the additive form of Chernoff bound [32], we have that for  $i \in \{0, 1\}$

$$\Pr[\widehat{Z} = i | f = 1, Z = 1 - i] \in \eta_{1-i} \pm \frac{\varepsilon}{2 \Pr[f = 1, Z = 1 - i]}$$

with probability at least  $1 - 2e^{-2\varepsilon^2 n}$ , which implies the first part.

For the second part, let  $\varepsilon' = \frac{\varepsilon(1-\eta_0-\eta_1)\lambda}{20}$  and assume that the first part holds. This implies that

$$(1 - \eta_1) \Pr[f = 1, \widehat{Z} = 0] - \eta_1 \Pr[f = 1, \widehat{Z} = 1] \geq .45(1 - \eta_0 - \eta_1)\lambda,$$

i.e.,  $\varepsilon'$  is negligible compared to  $(1 - \eta_1) \Pr[f = 1, \widehat{Z} = 0] - \eta_1 \Pr[f = 1, \widehat{Z} = 1]$ . By a similar argument, we also have that  $\varepsilon'$  is negligible compared to  $(1 - \eta_1)\widehat{\mu}_0 - \eta_1\widehat{\mu}_1$ . These properties ensure that  $\Gamma_0(f)$  estimates  $\Pr[f = 1 | Z = 0]$ : its numerator approximates  $(1 - \eta_0 - \eta_1) \Pr[f = 1, Z = 0]$  and its denominator approximates  $(1 - \eta_0 - \eta_1)\mu_0$ , which leads to the second part of Lemma 4.9.  $\square$

For the fairness performance, we need the following lemma that lower bounds the total probability that all bad classifiers violate Constraint (2), by the capacity of bad classifiers (Definition 4.8). Once again, we provide a proof sketch here, and the complete proof can be found in Section 6.2.

**Lemma 4.10 (Bad classifiers are not feasible for Program 2)** *Assuming  $\delta \in (0, 0.1\lambda)$ , then with probability at least  $1 - \Phi(\mathcal{F})$ , any bad classifier violates Constraint (2). Suppose the VC-dimension of  $(S, \mathcal{F})$  is  $t$ ; then with probability at least  $1 - O\left(e^{-\frac{(1-\eta_0-\eta_1)^2 \lambda^2 \delta^2 n}{5000} + t \ln(\frac{50}{(1-\eta_0-\eta_1)\lambda\delta})}\right)$ , any bad classifier violates Constraint (2).*

**Proof sketch:** We discuss the cases of  $\mathcal{G}_0$  and  $\mathcal{G}_i$  ( $i \in [T]$ ) separately. For  $\mathcal{G}_0$ , let  $G_0$  be an  $\varepsilon_0$ -net of  $\mathcal{G}_0$  of size  $M_{\varepsilon_0}(\mathcal{G}_0)$ . By Lemma 4.9, we can prove that with probability at least  $1 - 2e^{-2\varepsilon_0^2 n M_{\varepsilon_0}(\mathcal{G}_0)}$ , all classifiers  $g \in G_0$  violate either the first or the second constraint in (2) by at least an additive  $\frac{(1-\eta_0-\eta_1)\lambda}{2}$  term. Conditioned on this event, we can verify that all classifiers  $f \in \mathcal{G}_0$  violate at least one of the first two constraints of (2) since there must exist a classifier  $g \in G_0$  such that  $\Pr[f \neq g] \leq \varepsilon_0$ .

For  $\mathcal{G}_i$  ( $i \in [T]$ ), the argument is similar: first construct an  $\varepsilon_i$ -net  $G_i$  of  $\mathcal{G}_i$ , then show all classifiers in  $G_i$  are not feasible with probability at least  $1 - 4e^{-\frac{\varepsilon_i^2(1-\eta_0-\eta_1)^2\lambda^2 n}{200}}$  by Lemma 4.9, and finally extend to all classifiers in  $\mathcal{G}_i$ . The only difference is that each  $f \in \mathcal{G}_i$  violates the third constraint of (2), say  $\gamma^\Delta(f, \hat{S}) < \tau - \delta$  holds with high probability. The lemma is a direct corollary by the union bound.  $\square$   
Now we are ready to prove the main theorem.

**Proof:** [Proof of Theorem 4.3] We first upper bound the probability that  $\gamma^\Delta(f^\Delta, \hat{S}) \geq \tau - 3\delta$ . Let  $\mathcal{F}_b = \{f \in \mathcal{F} : \gamma(f, S) < \tau - 3\delta\}$ . If all classifiers in  $\mathcal{F}_b$  violate Constraint (2), we have that  $\gamma^\Delta(f^\Delta, \hat{S}) \geq \tau - 3\delta$ . Note that if  $\min_{i \in \{0,1\}} \Pr[f = 1, Z = i] \geq \frac{\lambda}{2}$ , then  $\gamma(f, S) \geq \frac{\lambda}{2}$  holds by definition. Also,  $\frac{\lambda - 3\delta}{1.01^{2T+1} - 1} \leq \frac{\lambda}{2}$ . Thus, we conclude that  $\mathcal{F}_b \subseteq \cup_{i=0}^T \mathcal{G}_i$ . Then if all bad classifiers violate Constraint (2), we have  $\gamma^\Delta(f^\Delta, \hat{S}) \geq \tau - 3\delta$ . By Lemma 4.10,  $\gamma^\Delta(f^\Delta, \hat{S}) \geq \tau - 3\delta$  holds with probability at least  $1 - O\left(e^{-\frac{(1-\eta_0-\eta_1)^2\lambda^2\delta^2 n}{5000} + t \ln\left(\frac{50}{(1-\eta_0-\eta_1)\lambda\delta}\right)}\right)$ .

Next, we upper bound the probability that  $f^*$  is feasible for Program DenoisedFair, which implies  $\frac{1}{N} \sum_{i \in [N]} L(f^\Delta, s_i) \leq \frac{1}{N} \sum_{i \in [N]} L(f^*, s_i)$ . Letting  $\varepsilon = \delta$  in Lemma 4.9, we have that with probability at least  $1 - 2e^{-2\delta^2 n} - 4e^{-\frac{(1-\eta_0-\eta_1)^2\lambda^2\delta^2 n}{200}}$ ,

$$\begin{cases} (1-\eta) \Pr[f^* = 1, \hat{Z} = 0] - \eta \Pr[f^* = 1, \hat{Z} = 1] \geq (1-\eta_0-\eta_1) \Pr[f^* = 1, Z = 0] - \delta, \\ (1-\eta) \Pr[f^* = 1, \hat{Z} = 1] - \eta \Pr[f^* = 1, \hat{Z} = 0] \geq (1-\eta_0-\eta_1) \Pr[f^* = 1, Z = 1] - \delta, \\ \gamma^\Delta(f^*, \hat{S}) \geq (1-\delta)\gamma(f, S) \geq \gamma(f, S) - \delta. \end{cases}$$

It implies that  $f^*$  is feasible for Program DenoisedFair with probability at least  $1 - 2e^{-2\delta^2 n} - 4e^{-\frac{(1-\eta_0-\eta_1)^2\lambda^2\delta^2 n}{200}}$ . This completes the proof.  $\square$

## 5 Extension to general $p \geq 2$ and multiple fairness constraints

In this section, we show how to solve Problem 1 for multiple, non-binary protected attributes and multiple fairness constraints. We consider a general class of fairness metrics defined in [13], based on the following definition.

**Definition 5.1 (Linear-fractional/Linear group performance functions [13])** *Given a classifier  $f \in \mathcal{F}$  and  $i \in [p]$ , we call  $q_i(f)$  the group performance of  $Z = i$  if  $q_i(f) = \Pr[\xi(f) \mid \xi'(f), Z = i]$  for some events  $\xi(f), \xi'(f)$  that might depend on the choice of  $f$ . Define a group performance function  $q : \mathcal{F} \rightarrow [0, 1]^p$  for any classifier  $f \in \mathcal{F}$  as  $q(f) = (q_1(f), \dots, q_p(f))$ . Denote  $\mathcal{Q}_{\text{linf}}$  to be the collection of all group performance functions. If  $\xi'$  does not depend on the choice of  $f$ ,  $q$  is said to be **linear**. Denote  $\mathcal{Q}_{\text{lin}} \subseteq \mathcal{Q}_{\text{linf}}$  to be the collection of linear group performance functions.*

At a high level, a classifier  $f$  is considered to be fair w.r.t. to  $q$  if  $q_1(f) \approx \dots \approx q_p(f)$ . Definition 5.1 is general and contains many fairness metrics. For instance, if  $\xi := (f = 1)$  and  $\xi' := (Y = 0)$ , we have  $q_i(f) = \Pr[f = 1 \mid Y = 0, Z = i]$  which is linear and called the false positive rate. If  $\xi := (Y = 0)$  and  $\xi' := (f = 1)$ , we have  $q_i(f) = \Pr[Y = 0 \mid f = 1, Z = i]$  which is linear-fractional and called the false discovery rate. See [13, Table 1] for more examples. Given a group performance function  $q$ , we define  $\Omega_q$  to be

$$\Omega_q(f, S) := \min_{i \in [p]} q_i(f) / \max_{i \in [p]} q_i(f).$$

**Remark 5.2** The fairness metric considered in [3], i.e., equalized odds, can also be captured using the above definition; equalized odds simply requires equal false positive and true positive rates across the protected types. The fairness metrics used in [39], on the other hand, are somewhat different; they work with statistical parity and equalized odds for binary protected attributes, however, while we define disparity  $\Omega_q$  as the ratio between the minimum and maximum  $q_i$ , [39] define the disparity using the additive difference of  $q_i$  across the protected types. It is not apparent how to extend their method for improving additive metrics to linear-fractional fairness metrics as they counter the noise by scaling the tolerance of their constraints, and it is unclear how to compute these scaling parameters prior to the optimization step when the group performance function  $q$  is conditioned on the classifier prediction. On the other hand, our method can handle additive metrics by using the difference of altered  $q_i$  across the noisy protected attribute to form fairness constraints.

Next, we extend the flipping noises to general  $p \geq 2$ .

**Definition 5.3 (Flipping noises in general)** Let  $H \in [0, 1]^{p \times p}$  be a matrix satisfying that  $\sum_{j \in [p]} H_{ij} = 1$  for any  $i \in [p]$ . For each  $i \in [N]$ , we assume that the protected attribute of the  $i$ -th sample  $z_i$  is observed as  $\hat{z}_i = j$  with probability  $H_{z_i j}$ , for any  $j \in [p]$ .

Note that  $H$  can be non-symmetric, i.e., it is possible that  $H_{ij} \neq H_{ji}$  for  $i \neq j$ . Definition 3.1 is also a special case of Definition 5.3 by letting  $H = \begin{bmatrix} 1 - \eta_0 & \eta_0 \\ \eta_1 & 1 - \eta_1 \end{bmatrix}$ .

In the binary setting, we assumed that  $\eta_0, \eta_1 \in (0, 0.5)$ , i.e., the probability that a protected attribute is not flipped is strictly greater than the probability that it is flipped. As stated earlier, when the noise parameter is high, we cannot learn any information about  $Z$  from  $\hat{Z}$ . Similar argument holds for the case of non-binary protected attribute, and so the sum of non-diagonal entries in each row is assumed to be strictly less than the diagonal entry, implying that probability of not flipping is greater than the probability of flipping for every protected attribute type. A useful property of such a *diagonally-dominant* matrix is that it is always non-singular [33].

With the above definitions, we are ready to propose the extension of Problem 1 to general  $p \geq 2$  and multiple protected attributes.

**Problem 2 (Fair classification with noisy protected attributes)** Given  $m$  protected attributes,  $k$  group performance functions  $q^{(1)}, \dots, q^{(k)}$  where each one is based on some protected attribute, a threshold vector  $\tau \in [0, 1]^k$  and a noisy dataset  $\hat{S}$  with noise matrix  $H$ , the goal is to learn an (approximate) optimal fair classifier  $f \in \mathcal{F}$  of the following program:

$$\begin{aligned} \min_{f \in \mathcal{F}} \quad & \frac{1}{N} \sum_{i \in [N]} L(f, s_i) \quad \text{s.t.} \\ & \Omega_{q^{(i)}}(f, S) \geq \tau_i, \quad \forall i \in [k]. \end{aligned} \tag{Gen-TargetFair}$$

We slightly abuse the notation by letting  $f^*$  also denote an optimal fair classifier of Program Gen-TargetFair. We will now design a denoised program for Problem 2. Note that we only need to show how to design denoised fairness constraints for an arbitrary group performance function  $q$ , and it can be naturally extended to multiple fairness constraints. Thus, we consider the case that  $k = 1$  in the following, i.e., Program Gen-TargetFair for a given function  $q$ . Accordingly, Assumption 1 changes to the following.

**Assumption 2 (Lower bound for events of  $f^*$ )** Suppose there exists constant  $\lambda \in (0, 0.5)$  such that  $\min_{i \in [p]} \Pr [\xi(f^*), \xi'(f^*), Z = i] \geq \lambda$ .

By definition, we know that for any  $i \in [p]$ ,

$$q_i(f) = \frac{\Pr [\xi(f), \xi'(f), Z = i]}{\Pr [\xi'(f), Z = i]}.$$

As in Program DenoisedFair, the main idea is to represent  $\Pr[\xi(f), \xi'(f), Z = i]$  or  $\Pr[\xi'(f), Z = i]$  by a linear combination of  $\{\Pr[\xi(f), \xi'(f), Z = j]\}_{j \in [p]}$  or  $\{\Pr[\xi'(f), Z = j]\}_{j \in [p]}$  respectively. For  $\Pr[\xi'(f), Z = i]$ , we only need to replace  $f = 1$  in the argument of statistical rate by  $\xi'(f)$ , and replace  $f = 1$  by  $(\xi(f), \xi'(f))$  in  $\Pr[\xi(f), \xi'(f), Z = i]$ .

Next, we show how to compute  $\Pr[\xi'(f), Z = i]$  (the argument for  $\Pr[\xi(f), \xi'(f), Z = i]$  is similar) Recall that  $\pi_{ij} := \Pr[\widehat{Z} = i \mid Z = j]$  for  $i, j \in [p]$ ,  $\mu_i := \Pr[Z = i]$  and  $\widehat{\mu}_i := \Pr[\widehat{Z} = i]$  for  $i \in [p]$ . Similar to Eq (5), we have for each  $i \in [p]$

$$\Pr[\xi'(f), \widehat{Z} = i] = \sum_{j \in [p]} \Pr[\widehat{Z} = i \mid \xi'(f), Z = j] \Pr[\xi'(f), Z = j].$$

By Definition 5.3 and a similar argument as in the proof of Lemma 4.9, we have the following lemma.

**Lemma 5.4 (Relation between  $\Pr[\xi'(f), \widehat{Z} = i]$  and  $\Pr[\xi'(f), Z = j]$ )** *Let  $\varepsilon \in (0, 1)$  be a fixed constant. With probability at least  $1 - 2pe^{-2\varepsilon^2 n}$ , we have for each  $i \in [p]$ ,*

$$\Pr[\xi'(f), \widehat{Z} = i] \in \sum_{j \in [p]} H_{ji} \cdot \Pr[\xi'(f), Z = j] \pm \varepsilon.$$

We define

$$w(f) := (\Pr[\xi'(f), Z = 1], \dots, \Pr[\xi'(f), Z = p]), \text{ and} \\ \widehat{w}(f) := (\Pr[\xi'(f), \widehat{Z} = 1], \dots, \Pr[\xi'(f), \widehat{Z} = p]).$$

Since  $H$  is non-singular,  $(H^\top)^{-1}$  exists. Let  $M := \max_{i \in [p]} \|(H^\top)^{-1}\|_1$  denote the maximum  $\ell_1$ -norm of a row of  $(H^\top)^{-1}$ . By Lemma 5.4, we directly obtain the following lemma.

**Lemma 5.5 (Approximation of  $\Pr[\xi'(f), Z = i]$ )** *With probability at least  $1 - 2pe^{-2\varepsilon^2 n}$ , for each  $i \in [p]$ ,*

$$w(f)_i \in (H^\top)_i^{-1} \widehat{w}(f) \pm \varepsilon \|(H^\top)_i^{-1}\|_1 \in (H^\top)_i^{-1} \widehat{w}(f) \pm \varepsilon M.$$

Thus, we use  $(H^\top)_i^{-1} \widehat{w}(f)$  to estimate  $\Pr[\xi'(f), Z = i]$ , and to estimate constraint  $\min_{i \in [p]} \Pr[\xi(f), \xi'(f), Z = i] \geq \lambda$ , we construct the following constraint:

$$(H^\top)^{-1} \widehat{w}(f) \geq (\lambda - \varepsilon M) \mathbf{1}. \quad (3)$$

Similarly, we define

$$u(f) := (\Pr[\xi(f), \xi'(f), Z = 1], \dots, \Pr[\xi(f), \xi'(f), Z = p]), \text{ and} \\ \widehat{u}(f) := (\Pr[\xi(f), \xi'(f), \widehat{Z} = 1], \dots, \Pr[\xi(f), \xi'(f), \widehat{Z} = p]).$$

Once again, we use  $(H^\top)_i^{-1} \widehat{u}(f)$  to estimate  $\Pr[\xi(f), \xi'(f), Z = i]$  and to estimate constraint  $\min_{i \in [p]} \Pr[\xi(f), \xi(f), \xi'(f), Z = i] \geq \lambda$ , we construct the following constraint:

$$(H^\top)^{-1} \widehat{u}(f) \geq (\lambda - \varepsilon M) \mathbf{1}. \quad (4)$$

Note that  $\widehat{u}(f) \leq \widehat{w}(f)$  by definition, and Inequality (4) is a sufficient condition for Inequality (3).

**Remark 5.6** *The first two constraints of Program DenoisedFair are a special case of Constraint (3). By Definition 3.1, we have that*

$$H = \begin{bmatrix} 1 - \eta_0 & \eta_0 \\ \eta_1 & 1 - \eta_1 \end{bmatrix} \text{ and } (H^\top)^{-1} = \begin{bmatrix} \frac{1 - \eta_1}{1 - \eta_0 - \eta_1} & -\frac{\eta_1}{1 - \eta_0 - \eta_1} \\ -\frac{\eta_0}{1 - \eta_0 - \eta_1} & \frac{1 - \eta_0}{1 - \eta_0 - \eta_1} \end{bmatrix}.$$

Then  $M = \frac{1}{1 - \eta_0 - \eta_1}$  and we can verify that the first two constraints of Program DenoisedFair are equivalent to Constraint (3) when  $\xi'(f) = (f = 1)$ .

Given  $\delta \in (0, 1)$ , we can now define the general denoised fair program as follows.

$$\begin{aligned} \min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i \in [N]} L(f, \hat{s}_i) \quad s.t. \\ (H^\top)^{-1} \hat{u}(f) \geq (\lambda - \delta) \mathbf{1}, \\ \min_{i \in [p]} \frac{(H^\top)^{-1} \hat{u}(f)}{(H^\top)^{-1} \hat{w}(f)} \geq (\tau - \delta) \cdot \max_{i \in [p]} \frac{(H^\top)^{-1} \hat{u}(f)}{(H^\top)^{-1} \hat{w}(f)}. \end{aligned} \quad (\text{Gen-DenoisedFair})$$

To provide the performance guarantees on the solution of the above program, once again we define the following general notions of bad classifiers and the corresponding capacity.

**Definition 5.7 (Bad classifiers in general)** *Given a family  $\mathcal{F} \subseteq \{0, 1\}^{\mathcal{X}}$ , we call  $f \in \mathcal{F}$  a bad classifier if  $f$  belongs to at least one of the following sub-families:*

- $\mathcal{G}_0 := \{f \in \mathcal{F} : \min_{i \in [p]} \Pr[\xi(f), \xi'(f), Z = i] < \frac{\lambda}{2}\};$
- Let  $T = \lceil 232 \log \log \frac{2(\tau-3\delta)}{\lambda} \rceil$ . For  $i \in [T]$ , define

$$\mathcal{G}_i := \left\{ f \in \mathcal{F} \setminus \mathcal{G}_0 : \Omega_q(f, S) \in \left[ \frac{\tau - 3\delta}{1.01^{2^{i+1}-1}}, \frac{\tau - 3\delta}{1.01^{2^i-1}} \right] \right\}.$$

Note that Definition 4.5 is a special case of the above definition by letting  $p = 2$ ,  $M = 10$ ,  $\xi(f) = (f = 1)$  and  $\xi'(f) = \emptyset$ . We next propose the following definition of capacity of bad classifiers.

**Definition 5.8 (Capacity of bad classifiers in general)** *Let  $\varepsilon_0 = \frac{\lambda-2\delta}{5M}$ . Let  $\varepsilon_i = \frac{1.01^{2^{i-1}}\delta}{5}$  for  $i \in [T]$  where  $T = \lceil 232 \log \log \frac{2(\tau-3\delta)}{\lambda} \rceil$ . Given a family  $\mathcal{F} \subseteq \{0, 1\}^{\mathcal{X}}$ , we denote the capacity of bad classifiers by*

$$\Phi(\mathcal{F}) := 2pe^{-2\varepsilon_0^2 n} M_{\varepsilon_0}(\mathcal{G}_0) + 4p \sum_{i \in [T]} e^{-\frac{\varepsilon_i^2 \lambda^2 n}{200M^2}} \cdot M_{\varepsilon_i \lambda / 10M}(\mathcal{G}_i).$$

By a similar argument as in Lemma 4.10, we can prove that  $\Phi(\mathcal{F})$  is an upper bound of the probability that there exists a bad classifier feasible for Program Gen-DenoisedFair. Consequently, we obtain the following theorem as an extension of Theorem 4.3.

**Theorem 5.9 (Performance of Program Gen-DenoisedFair)** *Suppose the VC-dimension of  $(S, \mathcal{F})$  is  $t \geq 1$ . Given any non-singular matrix  $H \in [0, 1]^{p \times p}$  with  $\sum_{j \in [p]} H_{ij} = 1$  for each  $i \in [p]$ ,  $\lambda \in (0, 0.5)$  and  $\delta \in (0, 0.1\lambda)$ , let  $f^\Delta \in \mathcal{F}$  denote an optimal fair classifier of Program Gen-DenoisedFair. With probability at least  $1 - \Phi(\mathcal{F}) - 4pe^{-\frac{\lambda^2 \delta^2 n}{200M^2}}$ , the following properties hold*

- $\frac{1}{N} \sum_{i \in [N]} L(f^\Delta, s_i) \leq \frac{1}{N} \sum_{i \in [N]} L(f^*, s_i);$
- $\Omega_q(f^\Delta, S) \geq \tau - 3\delta.$

*Specifically, if the VC-dimension of  $(S, \mathcal{F})$  is  $t$  and  $\delta \in (0, 1)$ , the success probability is at least  $1 - O(pe^{-\frac{\lambda^2 \delta^2 n}{5000M^2} + t \ln(50M/\lambda\delta)})$ .*

**Proof:** The proof is almost the same as in Theorem 4.3: we just need to replace  $\frac{1}{1-\eta_0-\eta_1}$  by  $M$  everywhere.

Note that the term  $4pe^{-\frac{\lambda^2 \delta^2 n}{200M^2}}$  is an upper bound of the probability that  $f^*$  is not feasible for Program Gen-DenoisedFair. The idea comes from Lemma 5.5 by letting  $\varepsilon = \frac{\lambda\delta}{20M}$  such that for each  $i \in [p]$ ,

$$w(f^*)_i \in (1 \pm \frac{\delta}{10})(H^\top)^{-1} \hat{w}(f^*) \text{ and}$$

$$u(f^*)_i \in (1 \pm \frac{\delta}{10})(H^\top)_i^{-1} \hat{u}(f^*).$$

Consequently,  $\frac{1}{N} \sum_{i \in [N]} L(f^\Delta, s_i) \leq \frac{1}{N} \sum_{i \in [N]} L(f^*, s_i)$ . Since  $\Phi(\mathcal{F})$  is an upper bound of the probability that there exists a bad classifier feasible for Program Gen-DenoisedFair, we complete the proof.  $\square$   
For multiple fairness constraints, the success probability of Theorem 5.9 changes to be

$$1 - O(kpe^{-\frac{\lambda^2 \delta^2 n}{5000M^2} + t \ln(50M/\lambda\delta)}).$$

## 6 Proof of Lemmas 4.9, 4.10

### 6.1 Proof of Lemma 4.9 - Relation between Program TargetFair and Denoised-Fair

**Proof:** We first present the following simple observation.

**Observation 6.1** 1)  $\mu_0 + \mu_1 = 1$ ,  $\hat{\mu}_0 + \hat{\mu}_1 = 1$ , and  $\pi_{0,i} + \pi_{1,i} = 1$  holds for  $i \in \{0, 1\}$ ; 2) For any  $i, j \in \{0, 1\}$ ,  $\Pr[Z = i \mid \hat{Z} = j] = \frac{\pi_{ji} \mu_i}{\hat{\mu}_j}$ ; 3) For any  $i \in \{0, 1\}$ ,  $\hat{\mu}_i = \pi_{i,i} \mu_i + \pi_{i,1-i} \mu_{1-i}$ .

Similar to Equation 36, we have

$$\begin{aligned} \Pr[f = 1, \hat{Z} = 0] &= \Pr[\hat{Z} = 0 \mid f = 1, Z = 0] \cdot \Pr[f = 1, Z = 0] \\ &\quad + \Pr[\hat{Z} = 0 \mid f = 1, Z = 1] \cdot \Pr[f = 1, Z = 1]. \end{aligned} \quad (5)$$

Similar to the proof of Lemma C.5, by the Chernoff bound (additive form) [32], both

$$\Pr[\hat{Z} = 1 \mid f = 1, Z = 0] \in \eta_0 \pm \frac{\varepsilon}{2 \Pr[f = 1, Z = 0]}, \quad (6)$$

and

$$\Pr[\hat{Z} = 0 \mid f = 1, Z = 1] \in \eta_1 \pm \frac{\varepsilon}{2 \Pr[f = 1, Z = 1]}, \quad (7)$$

hold with probability at least

$$1 - 2e^{-\frac{\varepsilon^2 n}{12\eta \Pr[f=1, Z=0]}} - 2e^{-\frac{\varepsilon^2 n}{12\eta \Pr[f=1, Z=1]}} \stackrel{\eta \leq 0.4}{\geq} 1 - 2e^{-2\varepsilon^2 n}.$$

Consequently, we have

$$\begin{aligned} \Pr[f = 1, \hat{Z} = 0] &= \Pr[\hat{Z} = 0 \mid f = 1, Z = 0] \cdot \Pr[f = 1, Z = 0] \\ &\quad + \Pr[\hat{Z} = 0 \mid f = 1, Z = 1] \cdot \Pr[f = 1, Z = 1] \text{ (Eq. 5)} \\ &\in \left(1 - \eta_0 \pm \frac{\varepsilon}{2 \Pr[f = 1, Z = 0]}\right) \cdot \Pr[f = 1, Z = 0] + \left(\eta_1 \pm \frac{\varepsilon}{2 \Pr[f = 1, Z = 1]}\right) \cdot \Pr[f = 1, Z = 1] \quad (8) \\ &\text{(Ineqs. 6 and 7)} \\ &\in (1 - \eta_0) \Pr[f = 1, Z = 0] + \eta_1 \Pr[f = 1, Z = 1] \pm \varepsilon, \end{aligned}$$

and similarly,

$$\begin{aligned} \Pr[f = 1, \hat{Z} = 1] &\in \eta_0 \Pr[f = 1, Z = 0] \\ &\quad + (1 - \eta_1) \Pr[f = 1, Z = 1] \pm \varepsilon. \end{aligned} \quad (9)$$

By the above two inequalities, we conclude that

$$(1 - \eta_1) \Pr[f = 1, \hat{Z} = 0] - \eta_1 \Pr[f = 1, \hat{Z} = 1]$$

$$\begin{aligned}
&\in (1 - \eta_1)((1 - \eta_0) \Pr[f = 1, Z = 0] \\
&\quad + \eta_1 \Pr[f = 1, Z = 1] \pm \varepsilon) - \eta_1(\eta_0 \Pr[f = 1, Z = 0] + (1 - \eta_1) \Pr[f = 1, Z = 1] \pm \varepsilon) \quad (\text{Ineqs. 8 and 9}) \\
&\in (1 - \eta_0 - \eta_1) \Pr[f = 1, Z = 0] \pm \varepsilon.
\end{aligned}$$

Similarly, we have

$$(1 - \eta_0) \Pr[f = 1, \widehat{Z} = 1] - \eta_0 \Pr[f = 1, \widehat{Z} = 0] \in (1 - \eta_0 - \eta_1) \Pr[f = 1, Z = 1] \pm \varepsilon.$$

This completes the proof of the first relation.

Next, we focus on the second relation. By assumption,  $\min\{\Pr[f = 1, Z = 0], \Pr[f = 1, Z = 1]\} \geq \frac{\lambda}{2}$ . Let  $\varepsilon' = \frac{\varepsilon(1-\eta_0-\eta_1)\lambda}{20}$ . By a similar argument as for the first relation, we have the following claim.

**Claim 6.2** *With probability at least  $1 - 4e^{-2(\varepsilon')^2 n}$ , we have*

$$\left\{ \begin{array}{l} (1 - \eta_1) \Pr[f = 1, \widehat{Z} = 0] - \eta_1 \Pr[f = 1, \widehat{Z} = 1] \\ \in (1 - \eta_0 - \eta_1) \Pr[f = 1, Z = 0] \pm \varepsilon', \\ (1 - \eta_0) \Pr[f = 1, \widehat{Z} = 1] - \eta_0 \Pr[f = 1, \widehat{Z} = 0] \\ \in (1 - \eta_0 - \eta_1) \Pr[f = 1, Z = 1] \pm \varepsilon', \\ (1 - \eta_1)\widehat{\mu}_0 - \eta_1\widehat{\mu}_1 \in (1 - \eta_0 - \eta_1)\mu_0 \pm \varepsilon', \\ (1 - \eta_0)\widehat{\mu}_1 - \eta_0\widehat{\mu}_0 \in (1 - \eta_0 - \eta_1)\mu_1 \pm \varepsilon'. \end{array} \right.$$

Now we assume Claim 6.2 holds whose success probability is at least  $1 - 4e^{-\frac{\varepsilon^2(1-\eta_0-\eta_1)^2\lambda^2 n}{200}}$  since  $\varepsilon' = \frac{\varepsilon(1-\eta_0-\eta_1)\lambda}{20}$ . Consequently, we have

$$\begin{aligned}
(1 - \eta_1) \Pr[f = 1, \widehat{Z} = 0] - \eta_1 \Pr[f = 1, \widehat{Z} = 1] &\geq (1 - \eta_0 - \eta_1) \Pr[f = 1, Z = 0] - \varepsilon' \quad (\text{Claim 6.2}) \\
&\geq \frac{(1 - \eta_0 - \eta_1)\lambda}{2} - \varepsilon' \quad (\text{by assumption}) \\
&\geq 0.45 \cdot (1 - \eta_0 - \eta_1)\lambda. (\eta \leq 0.4, \varepsilon' = \frac{\varepsilon(1 - \eta_0 - \eta_1)\lambda}{20})
\end{aligned} \tag{10}$$

Similarly, we can also argue that

$$(1 - \eta_1)\widehat{\mu}_0 - \eta_1\widehat{\mu}_1 \geq 0.45 \cdot (1 - \eta_0 - \eta_1)\lambda. \tag{11}$$

Then we have

$$\begin{aligned}
\Pr[f = 1 \mid Z = 0] &= \frac{\Pr[f = 1, Z = 0]}{\mu_0} \\
&\in \frac{(1 - \eta_1) \Pr[f = 1, \widehat{Z} = 0] - \eta_1 \Pr[f = 1, \widehat{Z} = 1] \pm \varepsilon'}{(1 - \eta_1)\widehat{\mu}_0 - \eta_1\widehat{\mu}_1 \pm \varepsilon'} \\
&\quad (\text{Claim 6.2}) \\
&\in \frac{\left((1 - \eta_1) \Pr[f = 1, \widehat{Z} = 0] - \eta_1 \Pr[f = 1, \widehat{Z} = 1]\right)}{\left(1 \pm \frac{\varepsilon'}{0.45 \cdot (1 - \eta_0 - \eta_1)\lambda}\right) ((1 - \eta_1)\widehat{\mu}_0 - \eta_1\widehat{\mu}_1)} \times \left(1 \pm \frac{\varepsilon'}{0.45 \cdot (1 - \eta_0 - \eta_1)\lambda}\right) \quad (\text{Ineq. 10}) \\
&\in \left(1 \pm \frac{\varepsilon}{9}\right)^2 \cdot \Gamma_0(f). \quad (\text{Defns. of } \Gamma_0(f) \text{ and } \varepsilon')
\end{aligned}$$

Similarly, we can also prove that

$$\Pr[f = 1 \mid Z = 1] \in \left(1 \pm \frac{\varepsilon}{9}\right)^2 \cdot \Gamma_1(f).$$



By the above two inequalities, we have that with probability at least  $1 - 4e^{-\frac{\varepsilon^2(1-\eta_0-\eta_1)^2\lambda^2n}{200}}$ ,

$$\begin{aligned}\gamma^\Delta(f, S) &= \min \left\{ \frac{\Gamma_0(f)}{\Gamma_1(f)}, \frac{\Gamma_1(f)}{\Gamma_0(f)} \right\} \\ &\in (1 \pm \varepsilon) \times \min \left\{ \frac{\Pr[f = 1 \mid Z = 0]}{\Pr[f = 1 \mid Z = 1]}, \frac{\Pr[f = 1 \mid Z = 1]}{\Pr[f = 1 \mid Z = 0]} \right\} \\ &\in (1 \pm \varepsilon) \cdot \gamma(f, S).\end{aligned}$$

Combining with Claim 6.2, we complete the proof of the second relation.  $\square$

## 6.2 Proof of Lemma 4.10 - Bad classifiers are not feasible for Program Denoised-Fair

**Proof:** Suppose  $\delta \in (0, 0.1\lambda)$ . We discuss  $\mathcal{G}_0$  and  $\mathcal{G}_i$  ( $i \in [T]$ ) separately.

**Bad classifiers in  $\mathcal{G}_0$ .** Let  $G_0$  be an  $\varepsilon_0$ -net of  $\mathcal{G}_0$  of size  $M_{\varepsilon_0}(\mathcal{G}_0)$ . Consider an arbitrary classifier  $g \in G_0$ . By Lemma 4.9, with probability at least  $1 - 2e^{-2\varepsilon_0^2n}$ , we have

$$\begin{aligned}(1 - \eta_1) \Pr[g = 1, \widehat{Z} = 0] - \eta_1 \Pr[g = 1, \widehat{Z} = 1] &\leq (1 - \eta_0 - \eta_1) \Pr[g = 1, Z = 0] + \varepsilon_0 \\ &< \frac{(1 - \eta_0 - \eta_1)\lambda}{2} + \varepsilon_0, \quad (\text{Defn. of } \mathcal{G}_0)\end{aligned}\tag{12}$$

and

$$(1 - \eta_0) \Pr[g = 1, \widehat{Z} = 1] - \eta_0 \Pr[g = 1, \widehat{Z} = 0] < \frac{(1 - \eta_0 - \eta_1)\lambda}{2} + \varepsilon_0.\tag{13}$$

By the union bound, all classifiers  $g \in G_0$  satisfy Inequalities 12 and 13 with probability at least  $1 - 2e^{-2\varepsilon_0^2n}M_{\varepsilon_0}(\mathcal{G}_0)$ . Suppose this event happens. We consider an arbitrary classifier  $f \in \mathcal{G}_0$ . W.l.o.g., we assume  $\Pr[f = 1, Z = 0] < \frac{\lambda}{2}$ . By Definition 4.6, there must exist a classifier  $g \in G_0$  such that  $\Pr[f \neq g] \leq \varepsilon_0$ . Then we have

$$\begin{aligned}(1 - \eta_1) \Pr[f = 1, \widehat{Z} = 0] - \eta_1 \Pr[f = 1, \widehat{Z} = 1] &\leq (1 - \eta_1)(\Pr[g = 1, \widehat{Z} = 0] + \varepsilon_0) \\ &\quad - \eta_1(\Pr[g = 1, \widehat{Z} = 1] - \varepsilon_0) \quad (\Pr[f \neq g] \leq \varepsilon_0) \\ &\leq \frac{(1 - \eta_0 - \eta_1)\lambda}{2} + 2\varepsilon_0 \quad (\text{Ineq. 12}) \\ &\leq \frac{(1 - \eta_0 - \eta_1)\lambda}{2} + \frac{(1 - \eta_0 - \eta_1)\lambda - 2\delta}{2} \quad (\text{Defn. of } \varepsilon_0) \\ &= (1 - \eta_0 - \eta_1)\lambda - \delta,\end{aligned}$$

Thus, we conclude that all classifiers  $f \in \mathcal{G}_0$  violate Constraint 2 with probability at least  $1 - 2e^{-2\varepsilon_0^2n}M_{\varepsilon_0}(\mathcal{G}_0)$ .

**Bad classifiers in  $\mathcal{G}_i$  for  $i \in [T]$ .** We can assume that  $\tau - 3\delta \geq \lambda/2$ . Otherwise, all  $\mathcal{G}_i$  for  $i \in [T]$  are empty, and hence, we complete the proof. Consider an arbitrary  $i \in [T]$  and let  $G_i$  be an  $\varepsilon_i$ -net of  $\mathcal{G}_i$  of size  $M_{\varepsilon_i(1-\eta_0-\eta_1)\lambda/10}(\mathcal{G}_i)$ . Consider an arbitrary classifier  $g \in G_i$ . By the proof of Lemma 4.9, with probability at least  $1 - 4e^{-\frac{\varepsilon_i^2(1-\eta_0-\eta_1)^2\lambda^2n}{200}}$ , we have

$$\begin{cases} (1 - \eta_1) \Pr[g = 1, \widehat{Z} = 0] - \eta_1 \Pr[g = 1, \widehat{Z} = 1] \in (1 - \eta_0 - \eta_1) \Pr[g = 1, Z = 0] \pm \frac{\varepsilon_i(1-\eta_0-\eta_1)\lambda}{20}, \\ (1 - \eta_0) \Pr[g = 1, \widehat{Z} = 1] - \eta_0 \Pr[g = 1, \widehat{Z} = 0] \in (1 - \eta_0 - \eta_1) \Pr[g = 1, Z = 1] \pm \frac{\varepsilon_i(1-\eta_0-\eta_1)\lambda}{20}, \\ \gamma^\Delta(f, \widehat{S}) \in (1 \pm \varepsilon_i) \cdot \gamma(f, S). \end{cases}\tag{14}$$

Moreover, we have

$$\gamma^\Delta(g, \widehat{S}) \leq (1 + \varepsilon_i) \cdot \gamma(g, S) < (1 + \varepsilon_i) \cdot \frac{\tau - 3\delta}{1.01^{2^i - 1}}. \quad (\text{Defn. of } \mathcal{G}_i) \quad (15)$$

By the union bound, all classifiers  $g \in G_i$  satisfy Inequality 15 with probability at least

$$1 - 4e^{-\frac{\varepsilon_i^2(1-\eta_0-\eta_1)^2\lambda^2n}{200}} M_{\varepsilon_i(1-\eta_0-\eta_1)\lambda/10}(\mathcal{G}_i).$$

Suppose this event happens. We consider an arbitrary classifier  $f \in \mathcal{G}_i$ . By Definition 4.6, there must exist a classifier  $g \in G_i$  such that  $\Pr[f \neq g] \leq \varepsilon_i(1 - \eta_0 - \eta_1)\lambda/10$ . By Inequality 14 and a similar argument as that for Inequality 10, we have

$$(1 - \eta_1) \Pr[g = 1, \widehat{Z} = 0] - \eta_1 \Pr[g = 1, \widehat{Z} = 1] \geq 0.45 \cdot (1 - \eta_0 - \eta_1)\lambda. \quad (16)$$

$$\begin{aligned} \Gamma_0(f) &= \frac{(1 - \eta_1) \Pr[f = 1, \widehat{Z} = 0] - \eta_1 \Pr[f = 1, \widehat{Z} = 1]}{(1 - \eta)\widehat{\mu}_0 - \eta\widehat{\mu}_1} \\ &\in \frac{(1 - \eta_1) \left( \Pr[g = 1, \widehat{Z} = 0] \pm \frac{\varepsilon_i(1-\eta_0-\eta_1)\lambda}{10} \right)}{(1 - \eta)\widehat{\mu}_0 - \eta\widehat{\mu}_1} \\ &\quad - \frac{\eta_1 \left( \Pr[g = 1, \widehat{Z} = 1] \pm \frac{\varepsilon_i(1-\eta_0-\eta_1)\lambda}{10} \right)}{(1 - \eta)\widehat{\mu}_0 - \eta\widehat{\mu}_1} (\Pr[f \neq g] \leq \varepsilon_i(1 - \eta_0 - \eta_1)\lambda/10) \\ &\in \frac{(1 - \eta_1) \Pr[g = 1, \widehat{Z} = 0] - \eta_1 \Pr[g = 1, \widehat{Z} = 1]}{(1 - \eta)\widehat{\mu}_0 - \eta\widehat{\mu}_1} \pm \frac{\frac{\varepsilon_i(1-\eta_1-\eta_1)\lambda}{5}}{(1 - \eta)\widehat{\mu}_0 - \eta\widehat{\mu}_1} \quad (\eta \leq 0.4) \\ &\in \frac{(1 - \eta) \Pr[g = 1, \widehat{Z} = 0] - \eta \Pr[g = 1, \widehat{Z} = 1]}{(1 - \eta)\widehat{\mu}_0 - \eta\widehat{\mu}_1} \times (1 \pm 0.45\varepsilon_i) \quad (\text{Ineq. 16}) \\ &\in (1 \pm 0.45\varepsilon_i) \cdot \Gamma_0(g). \end{aligned} \quad (17)$$

Similarly, we can also prove

$$\Gamma_1(f) \in (1 \pm 0.45\varepsilon_i) \cdot \Gamma_1(g). \quad (18)$$

Thus, we conclude that

$$\begin{aligned} \gamma^\Delta(f, \widehat{S}) &= \min \left\{ \frac{\Gamma_0(f)}{\Gamma_1(f)}, \frac{\Gamma_1(f)}{\Gamma_0(f)} \right\} \\ &\leq \frac{1 + 0.45\varepsilon_i}{1 - 0.45\varepsilon_i} \cdot \min \left\{ \frac{\Gamma_0(g)}{\Gamma_1(g)}, \frac{\Gamma_1(g)}{\Gamma_0(g)} \right\} \\ &\quad (\text{Ineqs. 17 and 18}) \\ &< \frac{1 + 0.45\varepsilon_i}{1 - 0.45\varepsilon_i} \cdot (1 + \varepsilon_i) \cdot \frac{\tau - 3\delta}{1.01^{2^i - 1}} \\ &\quad (\text{Ineq. 15}) \\ &\leq \frac{1 + 0.45\varepsilon_1}{1 - 0.45\varepsilon_1} \cdot (1 + \varepsilon_1) \cdot (\tau - 3\delta) \quad (\delta < 0.1\lambda) \\ &\leq \tau - \delta. \quad (\varepsilon_1 = \frac{1.01\delta}{5}) \end{aligned}$$

It implies that all classifiers  $f \in \mathcal{G}_i$  violate Constraint 2 with probability at least

$$1 - 4e^{-\frac{\varepsilon_i^2(1-\eta_0-\eta_1)^2\lambda^2n}{200}} M_{\varepsilon_i(1-\eta_0-\eta_1)\lambda/10}(\mathcal{G}_i).$$

Table 2: The performance of all algorithms over test datasets, including the average and standard error (in brackets) of accuracy, statistical rate and false positive rate. For **DenoisedLR-SR** and **DenoisedLR-FPR**, we report the performances with parameter  $\tau = 0.7, 0.9$ , and for **LZMV**, we report the performances with parameter  $\varepsilon_L = 0.01, 0.04, 0.10$ . The full accuracy-fairness tradeoffs when varying  $\tau$  can be found in Figure 1 for statistical rate and Appendix D for false positive rate. For each dataset and protected attribute, the metrics of the method that achieves the largest sum of mean accuracy and mean statistical rate (one way to measure fairness-accuracy tradeoff) has also been colored in green, and the method that achieves the largest sum of mean accuracy and mean false positive rate has been colored in yellow. Our method **DenoisedLR** achieves the best tradeoff, as measured in this manner, in 7 out of 8 settings; the only outlier is **Adult** (race) with false positive rate metric, where the performance of **DenoisedLR-FPR** is within one standard deviation of **AKM** which has the best tradeoff.

	Adult						COMPAS					
	acc	sex SR	FPR	acc	race SR	FPR	acc	sex SR	FPR	acc	race SR	FPR
<b>Unconstrained</b>	<b>.80</b>	.31	.45	<b>.80</b>	.68	.81	<b>.66</b>	.60	.62	<b>.66</b>	.53	.55
	(.0.0)	(.01)	(.03)	(0.0)	(.02)	(.09)	(.01)	(.05)	(.10)	(.01)	(.04)	(.05)
<b>FairLR-SR</b>	.76	.68	.68	.76	.69	.71	.58	.74	.75	.57	.74	.73
	(.01)	(.24)	(.21)	(.01)	(.27)	(.26)	(.04)	(.15)	(.13)	(.04)	(.13)	(.12)
<b>FairLR-FPR</b>	.76	.82	.78	.76	.83	.84	.59	.63	.67	.55	.72	.73
	(.01)	(.21)	(.25)	(0.0)	(.29)	(.29)	(.04)	(.24)	(.17)	(.03)	(.13)	(.14)
<b>LZMV</b> ( $\varepsilon_L = .01$ )	.35	<b>.99</b>	<b>.99</b>	.37	<b>.98</b>	<b>.99</b>	.54	.63	.67	.54	.53	.54
	(.01)	(0.0)	(0.0)	(.05)	(0.0)	(0.0)	(.01)	(.04)	(.09)	(.01)	(.02)	(.04)
<b>LZMV</b> ( $\varepsilon_L = .04$ )	.67	.85	<b>.99</b>	.77	.79	.85	.54	.61	.64	.56	.53	.54
	(.04)	(.06)	(.01)	(.03)	(.10)	(.09)	(.01)	(.04)	(.08)	(.01)	(.02)	(.03)
<b>LZMV</b> ( $\varepsilon_L = .10$ )	.78	.69	.79	.80	.70	.82	.61	.61	.64	.58	.54	.55
	(.02)	(.09)	(.11)	(0.0)	(.01)	(.08)	(.02)	(.05)	(.07)	(.01)	(.04)	(.03)
<b>AKM</b>	.77	.66	.89	<b>.80</b>	.72	<b>.90</b>	.65	.67	.72	.64	.69	.77
	(0.0)	(.05)	(.04)	(0.0)	(.02)	(.08)	(.01)	(.06)	(.09)	(.01)	(.05)	(.06)
<b>DenoisedLR-SR</b> ( $\tau = .7$ )	.77	.74	.87	.79	.80	.90	.63	.75	.77	.60	.72	.72
	(.01)	(.14)	(.17)	(.01)	(.12)	(.10)	(.02)	(.07)	(.11)	(.03)	(.08)	(.09)
<b>DenoisedLR-SR</b> ( $\tau = .9$ )	<b>.76</b>	<b>.85</b>	.80	<b>.76</b>	<b>.88</b>	.90	<b>.58</b>	<b>.85</b>	<b>.86</b>	<b>.55</b>	<b>.83</b>	.80
	(.01)	(.15)	(.12)	(.01)	(.18)	(.19)	(.04)	(.09)	(.11)	(.03)	(.12)	(.11)
<b>DenoisedLR-FPR</b> ( $\tau = .7$ )	.77	.73	.85	.78	.77	.88	.61	.80	.82	.61	.72	.76
	(.02)	(.14)	(.17)	(.02)	(.11)	(.11)	(.03)	(.06)	(.08)	(.05)	(.12)	(.11)
<b>DenoisedLR-FPR</b> ( $\tau = .9$ )	<b>.77</b>	.77	<b>.91</b>	.77	.80	.88	<b>.61</b>	.79	<b>.83</b>	<b>.57</b>	<b>.83</b>	<b>.86</b>
	(.02)	(.12)	(.11)	(.02)	(.15)	(.14)	(.04)	(.07)	(.10)	(.04)	(.14)	(.08)

By the union bound, we complete the proof of Lemma 4.10 for  $\delta \in (0, 0.1\lambda)$ .

For general  $\delta \in (0, 1)$ , each bad classifier violates Constraint 2 with probability at most  $4e^{-\frac{\varepsilon_1^2(1-\eta_0-\eta_1)^2\lambda^2n}{200}}$  by the above argument. By Definition 4.5,  $|M_{\varepsilon_0}(\mathcal{G}_0)| + \sum_{i \in [T]} |M_{\varepsilon_i(1-\eta_0-\eta_1)\lambda/10}(\mathcal{G}_i)| \leq |M_{\varepsilon_1(1-\eta_0-\eta_1)\lambda/10}(\mathcal{F})|$ . Then by the definition of  $\Phi(\mathcal{F})$  and Theorem 4.7, the probability that there exists a bad classifier violating Constraint 2 is at most  $\Phi(\mathcal{F}) = O\left(e^{-\frac{(1-\eta_0-\eta_1)^2\lambda^2\delta^2n}{5000} + t \ln\left(\frac{50}{(1-\eta_0-\eta_1)\lambda\delta}\right)}\right)$ . This completes the proof of Lemma 4.10.  $\square$

## 7 Empirical results

We implement our denoised program and compare the performance with baseline algorithms on real-world datasets.

**Datasets.** We perform simulations on the **Adult** [2] and **COMPAS** [1] datasets, as pre-processed in [5]. The **Adult** dataset consists of rows corresponding to 48,842 individuals, with 18 binary features and a label indicating whether the income is greater than 50k USD or not. The **COMPAS** dataset consists of rows corresponding to 5378 individuals, with 10 binary features and a label indicating whether the individual reoffends. We take sex and race (coded as binary in the preprocessed datasets) to be the protected attributes.

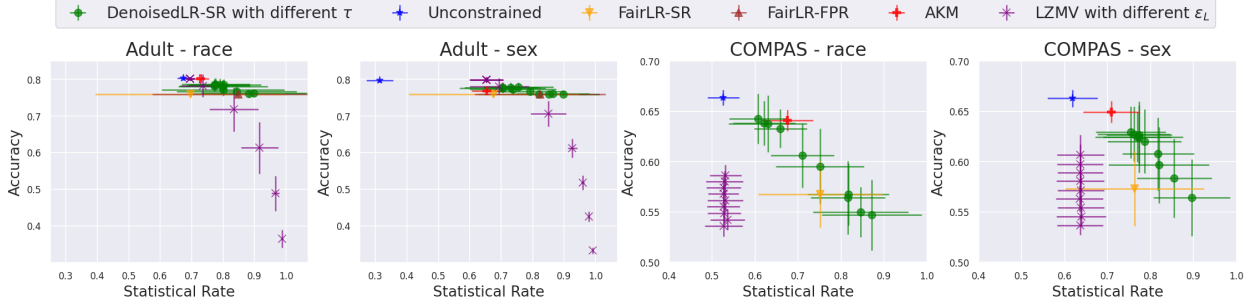


Figure 1: Performance of **DenoisedLR-SR** and baselines with respect to statistical rate and accuracy for different combinations of dataset and protected attribute. For **DenoisedLR-SR**, the performance for different  $\tau$  is presented, while for **LZMV** the input parameter  $\varepsilon_L$  is varied. The plots show that for all settings **DenoisedLR-SR** can attain a high statistical rate, often with minimal loss in accuracy.

**Baseline and metrics.** We implement our program with denoised constraints with respect to the statistical rate and false positive rate metrics; we refer to our algorithm with statistical rate constraints as **DenoisedLR-SR** and with false positive rate constraints as **DenoisedLR-FPR**. For the simulations, the probability terms in (2) are replaced by their empirical counterparts; the details of the constraints are presented in Section D. We compare against state-of-the-art noise-tolerant fair classification algorithms: **LZMV** [39] and **AKM** [3]. Similar to our approach, the algorithm **LZMV** also takes as input a parameter to control the fairness of the final classifier; for statistical parity, this parameter represents the desired absolute difference between the likelihood of positive class label across the two protected groups and **LZMV** is, therefore, the primary baseline for comparison with respect to statistical rate. We call the parameter in **LZMV**  $\varepsilon_L$  and present the results of [39] for different  $\varepsilon_L$  values. **AKM** is the primary baseline for comparison with respect to false positive rate metric. As discussed earlier, the algorithm **AKM** is the post-processing algorithm of Hardt et al. [29], and [3] claim that this post-processing algorithm can ensure equalized odds for certain classes of noisy datasets as well.<sup>3</sup> Additionally, we implement the algorithm **FairLR** which minimizes the logistic loss with fairness constraints over the given noisy dataset as described in Section C.2. When the fairness metric is statistical rate, we will refer to this program as **FairLR-SR**, and when the fairness metric is false positive rate, we will refer to it as **FairLR-FPR**. Finally, we also learn an unconstrained optimal classifier as a baseline.

**Implementation details.** We first shuffle and partition the dataset into a train and test partition (70-30 split). Given the training dataset  $S$ , we generate a noisy dataset  $\hat{S}$  with, for illustrative purposes,  $\eta_0 = 0.3$  and  $\eta_1 = 0.1$  (i.e., the minority group is more likely to contain errors, as would be expected in various applications [46]). We consider other choices of  $\eta_0, \eta_1$  and impact of error in estimates of  $\eta_0, \eta_1$  in Appendix D. We train each algorithm on  $\hat{S}$  and vary the fairness constraints (e.g., the choice of  $\tau \in [0.5, 0.95]$  in **DenoisedLR**), learn the corresponding fair classifier, and report its accuracy (acc) and fairness metric (either statistical rate or false positive rate)  $\gamma$  over the noisy version of the test dataset. We perform 50 repetitions, and report the mean and standard error of fairness and accuracy metrics across the repetitions. For COMPAS dataset we use  $\lambda = 0.1$  as a large fraction (47%) of training samples have class label 1, while for Adult dataset we use  $\lambda = 0$  as the fraction of positive class labels is small (24%).

**Results.** Table 2 summarizes the fairness and accuracy achieved by **DenoisedLR-SR**, **DenoisedLR-FPR** and baseline algorithms over the **Adult** and **COMPAS** test datasets. The first observation to note is the low statistical rate and false positive rate of the unconstrained classifier, emphasizing the necessity of noise-tolerant fairness interventions. Secondly, our approach **DenoisedLR-SR**, **DenoisedLR-FPR** can achieve higher fairness than the baselines in most cases. The extent of this improvement varies with the strength of the constraint  $\tau$ , but comes with a natural tradeoff with accuracy.

<sup>3</sup>Equalized odds fairness metric aims for parity w.r.t false positive and true positive rates. For clarity of presentation, we present the empirical analysis with respect to false positive rate only. However, true positive rate constraints can also be incorporated.

With respect to statistical rate, **DenoisedLR-SR** can attain a higher statistical rate than **FairLR-SR** for  $\tau = 0.9$ , and performs similarly with respect to accuracy. The statistical rate-accuracy tradeoff of **DenoisedLR-SR** is also better than **LZMV** and **AKM**; in particular, high statistical rate for Adult dataset using **LZMV** (i.e.,  $\geq 0.8$ ) is achieved only with a relatively larger loss in accuracy (for example, with  $\varepsilon_L = 0.01$ ), whereas for **DenoisedLR-SR**, the loss in accuracy when using  $\tau = 0.9$  is relatively small ( $\sim 0.03$ ) while the statistical rate is still high ( $\sim 0.85$ ). For COMPAS dataset **LZMV** cannot achieve high statistical rate for any  $\varepsilon_L$ , while **DenoisedLR-SR** returns classifier that approximately satisfies the desired statistical rate guarantees. The tradeoff between statistical rate and accuracy for all methods is also presented in Figure 1. The plot further shows that, for Adult dataset, **DenoisedLR-SR** can achieve higher statistical rate than **LZMV** at a much smaller loss in accuracy, while for COMPAS, **LZMV** is not suitable in any setting.

With respect to false positive rate, once again **DenoisedLR-FPR** achieves a larger false positive rate than **FairLR-FPR** and the unconstrained classifier. The primary baseline for comparison with false positive rate is **AKM**; due to noise in protected attribute during test phase, **AKM** cannot achieve high false positive rate in every setting. The false positive rate of **AKM** is high for the Adult dataset ( $\sim 0.90$ ) but quite low for the COMPAS dataset. In comparison **DenoisedLR-FPR** for Adult dataset with  $\tau = 0.9$  can achieve similar high false positive rate as **AKM** at a small loss of accuracy (the best false positive rate and accuracy of both methods are within a standard deviation of each other). On the other hand, for COMPAS dataset, **AKM** has lower false positive rate than **DenoisedLR-FPR** for both protected attributes. Baseline **LZMV** attains high false positive rate too for the Adult dataset, but the loss in accuracy is larger compared to **DenoisedLR-FPR**. The tradeoff between false positive rate and accuracy for all methods is also presented in Figure 3.

Evaluation with respect to both metrics shows that our framework can attain close to the user-desired fairness metric values (as defined using  $\tau$ ); comparison with baselines further shows that, unlike **AKM**, our approach can always return classifiers with high fairness metrics values, and unlike **LZMV**, the loss in accuracy to achieve high fairness values is relatively small.

## 8 Conclusion, limitations and future work

In this paper, we study fair classification with noisy protected attributes. We consider flipping noises and propose a unified framework that constructs an approximate optimal fair classifier over the underlying dataset for multiple, non-binary protected attributes and multiple fairness constraints. Empirically, our denoised algorithm can achieve the high fairness values at a small cost to accuracy.

Our work leaves several interesting future directions. One is to consider other noise models that are not independent, e.g., settings where the noise follows a general mutually contaminated model [52] or settings where the noise on the protected type also depends on other features, such as, when imputing the protected attributes. While our framework can still be employed in these settings (e.g., given group prediction error rates), methods that take into account the protected attribute prediction model could potentially further improve the performance. There exist several works that also design fair classifiers with noisy labels [9, 8] and another direction is to consider joint noises over both protected attributes and labels. Our model is also related to the setting in which each protected attribute follows a known distribution; whether our methods can be adapted to this setting can be investigated as part of future work.

In terms of broader impact, our recent news cycle has once again pointed to the fact that systemic biases pervade our world to the detriment of marginalized people, ranging from disparities in policing to widespread health disparities brought to light yet again with COVID19. This has been shown time and again by compelling research within and outside of computer science. Algorithms are yet one more system and one that again perpetuates harm when left unchecked. Thus, developing techniques for fair classification is crucial if we hope to effect broader societal change.

This work helps broaden the class of settings where fair classification techniques can be applied by working even when the information about protected attributes is noisy. Data collection and cleaning are complex (and sometimes political) processes and may contain recording and reporting errors unintentional or otherwise [50, 46]. Further, information about protected attributes may be missing entirely [17] or could be predicted by other algorithms thus containing additional errors and biases [45, 10]. A setting in which

protected attributes are known (near) perfectly are the exception rather than the norm, and this paper shows that one must be cautious when applying existing fairness methodology to noisy settings, and provides an alternate path forward. Our approach takes protected attribute errors into consideration and is likely to expand the scope of application for fair classification by removing the clean data assumption.

Additionally, our framework can be applied to a wide class of fairness metrics, and hence may be suitable in many domains. However, it is not apriori clear which protected attributes or which fairness metrics should be used in any given setting, and the answers will be very context-dependent; the effectiveness of our framework towards mitigating bias will depend crucially on whether the appropriate choice of features and parameters are selected. An ideal implementation of our framework would involve an active dialogue between the users and designers, a careful assessment of impact both pre and post-deployment. This would in particular benefit from regular public audits, as well as ways to obtain and incorporate community feedback from stakeholders [51, 15].

## Acknowledgements

This research was supported in part by a J.P. Morgan Faculty Award, an AWS MLRA grant, and NSF CCF-1908347 grant.

## References

- [1] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. <https://github.com/propublica/compas-analysis>, 2016.
- [2] A. Asuncion and D.J. Newman. UCI machine learning repository. [archive.ics.uci.edu/ml/index.php](http://archive.ics.uci.edu/ml/index.php), 2007. University of California, Irvine, School of Information and Computer Sciences.
- [3] Pranjal Awasthi, Matthaus Kleindessner, and Jamie Morgenstern. Equalized odds postprocessing under imperfect group information. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020*, 2020.
- [4] Tony Barboza and Joseph Serna. As coronavirus deaths surge, missing racial data worry L.A. county officials. <https://www.latimes.com/california/story/2020-04-06/missing-racial-data-coronavirus-deaths-worries-los-angeles-county-officials>, 2020. Los Angeles Times.
- [5] Rachel K. E. Bellamy, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, Seema Nagar, Karthikeyan Natesan Ramamurthy, John Richards, Diptikalyan Saha, Prasanna Sattigeri, Moninder Singh, Kush R. Varshney, and Yunfeng Zhang. AI Fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias, October 2018.
- [6] Rachel KE Bellamy, Kuntal Dey, Michael Hind, Samuel C Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, Jacquelyn Martino, Sameep Mehta, Aleksandra Mojsilovic, et al. AI fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias. *arXiv preprint arXiv:1810.01943*, 2018.
- [7] Dan Biddle. *Adverse impact and test validation: A practitioner’s guide to valid and defensible employment testing*. Gower Publishing, Ltd., 2006.
- [8] Arpita Biswas and Suvam Mukherjee. Ensuring fairness under prior probability shifts. *ArXiv*, abs/2005.03474, 2020.
- [9] Avrim Blum and Kevin Stangl. Recovering from biased data: Can fairness constraints improve accuracy? *Symposium on the foundations of responsible computing, (FORC) 2020*, 2020.
- [10] Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency, FAT 2018, 23-24 February 2018, New York, NY, USA*, pages 77–91, 2018.

- [11] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.
- [12] Toon Calders, Faisal Kamiran, and Mykola Pechenizkiy. Building classifiers with independency constraints. *2009 IEEE International Conference on Data Mining Workshops*, pages 13–18, 2009.
- [13] L. Elisa Celis, Lingxiao Huang, Vijay Keswani, and Nisheeth K. Vishnoi. Classification with fairness constraints: A meta-algorithm with provable guarantees. In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT\* 2019, Atlanta, GA, USA, January 29-31, 2019*, pages 319–328, 2019.
- [14] L Elisa Celis, Vijay Keswani, and Nisheeth K Vishnoi. Data preprocessing to mitigate bias: A maximum entropy based approach.
- [15] Stevie Chancellor, Shion Guha, Jofish Kaye, Jen King, Niloufar Salehi, Sarita Schoenebeck, and Elizabeth Stowell. The relationships between data, power, and justice in cscw research. In *Conference Companion Publication of the 2019 on Computer Supported Cooperative Work and Social Computing*, pages 102–105, 2019.
- [16] Jiahao Chen, Nathan Kallus, Xiaojie Mao, Geoffry Svacha, and Madeleine Udell. Fairness under unawareness: Assessing disparity when protected class is unobserved. In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT\* 2019, Atlanta, GA, USA, January 29-31, 2019*, pages 339–348, 2019.
- [17] Ethnicity Data, Michele Ver Ploeg, and Edward Perrin. Eliminating health disparities: Measurement and data needs. *Washington (DC): National Academies Press (US)*, 2004.
- [18] Michele Donini, Luca Oneto, Shai Ben-David, John S Shawe-Taylor, and Massimiliano Pontil. Empirical risk minimization under fairness constraints. In *Advances in Neural Information Processing Systems*, pages 2791–2801, 2018.
- [19] Flávio du Pin Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R. Varshney. Optimized pre-processing for discrimination prevention. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 3992–4001, 2017.
- [20] Cynthia Dwork, Nicole Immorlica, Adam Tauman Kalai, and Mark D. M. Leiserson. Decoupled classifiers for group-fair and efficient machine learning. In *Fairness, Accountability, and Transparency in Machine Learning*, pages 119–133, 2018.
- [21] Benjamin Fish, Jeremy Kun, and Ádám D Lelkes. A confidence-based approach for balancing fairness and accuracy. In *Proceedings of the 2016 SIAM International Conference on Data Mining, 2016*, pages 144–152. SIAM, 2016.
- [22] David A Freedman. *Statistical models: theory and practice*. cambridge university press, 2009.
- [23] Sorelle A Friedler, Carlos Scheidegger, Suresh Venkatasubramanian, Sonam Choudhary, Evan P Hamilton, and Derek Roth. A comparative study of fairness-enhancing interventions in machine learning. In *Fairness, Accountability, and Transparency in Machine Learning*, 2019.
- [24] Naman Goel, Mohammad Yaghini, and Boi Faltings. Non-discriminatory machine learning through convex fairness criteria. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [25] Gabriel Goh, Andrew Cotter, Maya R. Gupta, and Michael P. Friedlander. Satisfying real-world goals with dataset constraints. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems*, pages 2415–2423, 2016.

- [26] Paula Gordaliza, Eustasio del Barrio, Fabrice Gamboa, and Jean-Michel Loubes. Obtaining fairness using optimal transport theory. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 2357–2365, 2019.
- [27] Maya R. Gupta, Andrew Cotter, Mahdi Milani Fard, and Serena Wang. Proxy fairness. *CoRR*, abs/1806.11212, 2018.
- [28] Sarel Har-peled. *Geometric Approximation Algorithms*. American Mathematical Society, USA, 2011.
- [29] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing*, pages 3315–3323, 2016.
- [30] Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *ICML 2018: Thirty-fifth International Conference on Machine Learning*, pages 1929–1938, 2018.
- [31] David Haussler. Sphere packing numbers for subsets of the Boolean  $n$ -cube with bounded Vapnik-Chervonenkis dimension. *Journal of Combinatorial Theory, Series A*, 69(2):217–232, 1995.
- [32] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, pages 409–426. Springer, 1994.
- [33] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.
- [34] Lingxiao Huang and Nisheeth K. Vishnoi. Stable and fair classification. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 2879–2890, 2019.
- [35] Nathan Kallus, Xiaojie Mao, and Angela Zhou. Assessing algorithmic fairness with unobserved protected class using data combination. In *FAT\* ’20: Conference on Fairness, Accountability, and Transparency, Barcelona, Spain, January 27-30, 2020*, page 110, 2020.
- [36] Faisal Kamiran and Toon Calders. Classifying without discriminating. In *Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference on*, pages 1–6. IEEE, 2009.
- [37] Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, 2012.
- [38] D. Kraft. *A Software Package for Sequential Quadratic Programming*. Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt Köln: Forschungsbericht. Wiss. Berichtswesen d. DFVLR, 1988.
- [39] Alexandre Louis Lamy and Ziyuan Zhong. Noise-tolerant fair classification. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 294–305, 2019.
- [40] J. H. Van Lint. *Introduction to Coding Theory*. Springer-Verlag, Berlin, Heidelberg, 3rd edition, 1998.
- [41] Tongliang Liu and Dacheng Tao. Classification with noisy labels by importance reweighting. *IEEE Transactions on pattern analysis and machine intelligence*, 38(3):447–461, 2015.
- [42] Aditya Menon, Brendan Van Rooyen, Cheng Soon Ong, and Bob Williamson. Learning from corrupted binary labels via class-probability estimation. In *International Conference on Machine Learning*, pages 125–134, 2015.
- [43] Aditya Krishna Menon and Robert C Williamson. The cost of fairness in binary classification. In *Conference on Fairness, Accountability and Transparency*, pages 107–118, 2018.
- [44] Aditya Krishna Menon and Robert C. Williamson. The cost of fairness in binary classification. In *Conference on Fairness, Accountability and Transparency, FAT*, pages 107–118, 2018.



- [45] Vidya Muthukumar, Tejaswini Pedapati, Nalini K. Ratha, Prasanna Sattigeri, Chai-Wah Wu, Brian Kingsbury, Abhishek Kumar, Samuel Thomas, Aleksandra Mojsilovic, and Kush R. Varshney. Understanding unequal gender classification accuracy from face images. *CoRR*, abs/1812.00099, 2018.
- [46] Melissa Nobles. Shades of citizenship: Race and the census in modern politics. *Bibliovault OAI Repository, the University of Chicago Press*, 2000.
- [47] Curtis G Northcutt, Tailin Wu, and Isaac L Chuang. Learning with confident examples: Rank pruning for robust classification with noisy labels. *arXiv preprint arXiv:1705.01936*, 2017.
- [48] Ryan O’Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.
- [49] Geoff Pleiss, Manish Raghavan, Felix Wu, Jon M. Kleinberg, and Kilian Q. Weinberger. On fairness and calibration. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 5684–5693, 2017.
- [50] Jose A Saez, Mikel Galar, Julian Luengo, and Francisco Herrera. Tackling the problem of classification with noisy data using multiple classifier systems: Analysis of the performance and robustness. *Information Sciences*, 247:1–20, 2013.
- [51] Hannah Sassaman, Jennifer Lee, Jenessa Irvine, and Shankar Narayan. Creating community-based tech policy: case studies, lessons learned, and what technologists and communities can do together. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 685–685, 2020.
- [52] Clayton Scott, Gilles Blanchard, and Gregory Handy. Classification with asymmetric label noise: Consistency and maximal denoising. In *Conference on Learning Theory*, pages 489–511, 2013.
- [53] Hao Wang, Berk Ustun, and Flávio P. Calmon. Repairing without retraining: Avoiding disparate impact with counterfactual distributions. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 6618–6627, 2019.
- [54] Serena Wang, Wenshuo Guo, Harikrishna Narasimhan, Andrew Cotter, Maya R. Gupta, and Michael I. Jordan. Robust optimization for fairness with noisy protected groups. *CoRR*, abs/2002.09343, 2020.
- [55] Blake E. Woodworth, Suriya Gunasekar, Mesrob I. Ohannessian, and Nathan Srebro. Learning non-discriminatory predictors. In *Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017*, pages 1920–1953, 2017.
- [56] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017*, pages 1171–1180, 2017.
- [57] Muhammad Bilal Zafar, Isabel Valera, Manuel Gomez-Rodriguez, and Krishna P. Gummadi. Fairness constraints: Mechanisms for fair classification. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, AISTATS 2017, 20-22 April 2017, Fort Lauderdale, FL, USA*, pages 962–970, 2017.
- [58] Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pages 335–340, 2018.

## A Analysis of the influences of estimation errors for $\eta_0$ and $\eta_1$

Recall that we assume  $\eta_0$  and  $\eta_1$  are given in Theorem 4.3. However, we may only have estimations for  $\eta_0$  and  $\eta_1$  in practice, say  $\eta'_0$  and  $\eta'_1$  respectively. Define  $\zeta := \max\{|\eta_0 - \eta'_0|, |\eta_1 - \eta'_1|\}$  to be the additive estimation error. We want to understand the influences of  $\zeta$  to the performance of our denoised program.

Since  $\eta_0$  and  $\eta_1$  are unknown now, we can not directly compute  $\Gamma_0(f)$  and  $\Gamma_1(f)$  in Definition 4.1. Instead, we can compute

$$\begin{aligned}\Gamma'_0(f) &:= \frac{(1 - \eta'_1) \Pr[f = 1, \widehat{Z} = 0] - \eta'_1 \Pr[f = 1, \widehat{Z} = 1]}{(1 - \eta'_1)\widehat{\mu}_0 - \eta'_1\widehat{\mu}_1}, \\ \Gamma_1(f) &:= \frac{(1 - \eta'_0) \Pr[f = 1, \widehat{Z} = 1] - \eta'_0 \Pr[f = 1, \widehat{Z} = 0]}{(1 - \eta'_0)\widehat{\mu}_1 - \eta'_0\widehat{\mu}_0}.\end{aligned}$$

Then we have

$$\begin{aligned}\Gamma'_0(f) &= \frac{(1 - \eta'_1) \Pr[f = 1, \widehat{Z} = 0] - \eta'_1 \Pr[f = 1, \widehat{Z} = 1]}{(1 - \eta'_1)\widehat{\mu}_0 - \eta'_1\widehat{\mu}_1} \\ &= \frac{(1 - \eta_1) \Pr[f = 1, \widehat{Z} = 0] - \eta_1 \Pr[f = 1, \widehat{Z} = 1]}{(1 - \eta_1)\widehat{\mu}_0 - \eta_1\widehat{\mu}_1 + (\eta_1 - \eta'_1)} + \frac{(\eta_1 - \eta'_1) \Pr[f = 1]}{(1 - \eta_1)\widehat{\mu}_0 - \eta_1\widehat{\mu}_1 + (\eta_1 - \eta'_1)} \\ &\in \frac{(1 - \eta_1) \Pr[f = 1, \widehat{Z} = 0] - \eta_1 \Pr[f = 1, \widehat{Z} = 1]}{(1 - \eta_1)\widehat{\mu}_0 - \eta_1\widehat{\mu}_1} \pm \frac{\zeta \cdot \Pr[f = 1]}{(1 - \eta_1)\widehat{\mu}_0 - \eta_1\widehat{\mu}_1} \quad (\text{Defn. of } \zeta) \\ &\in \Gamma_0(f) \pm \frac{\zeta \cdot \Pr[f = 1]}{(1 - \eta_1)\widehat{\mu}_0 - \eta_1\widehat{\mu}_1}. \quad (\text{Defn. of } \Gamma_0(f))\end{aligned} \tag{19}$$

Symmetrically, we have

$$\Gamma'_1(f) \in \Gamma_1(f) \pm \frac{\zeta \cdot \Pr[f = 1]}{(1 - \eta_0)\widehat{\mu}_1 - \eta_0\widehat{\mu}_0}. \tag{20}$$

By a similar argument, we can also prove that

$$\frac{1}{\Gamma'_0(f)} \in \frac{1}{\Gamma_0(f)} \pm \frac{\zeta}{(1 - \eta_1) \Pr[f = 1, \widehat{Z} = 0] - \eta_1 \Pr[f = 1, \widehat{Z} = 1]}. \tag{21}$$

and

$$\frac{1}{\Gamma'_1(f)} \in \frac{1}{\Gamma_1(f)} \pm \frac{\zeta}{(1 - \eta_0) \Pr[f = 1, \widehat{Z} = 1] - \eta_0 \Pr[f = 1, \widehat{Z} = 0]}. \tag{22}$$

Then by the denoised constraint on  $\eta'_0$  and  $\eta'_1$ , i.e.,

$$\min \left\{ \frac{\Gamma'_1(f)}{\Gamma'_0(f)}, \frac{\Gamma'_0(f)}{\Gamma'_1(f)} \right\} \geq \tau - \delta, \tag{23}$$

we conclude that

$$\begin{aligned}\frac{\Gamma_1(f)}{\Gamma_0(f)} &\geq \left( \Gamma'_1(f) - \frac{\zeta \cdot \Pr[f = 1]}{(1 - \eta_0)\widehat{\mu}_1 - \eta_0\widehat{\mu}_0} \right) \times \\ &\quad \left( \frac{1}{\Gamma'_0(f)} - \frac{\zeta}{(1 - \eta_1) \Pr[f = 1, \widehat{Z} = 0] - \eta_1 \Pr[f = 1, \widehat{Z} = 1]} \right) (\text{Ineqs. 20 and 21}) \\ &\geq \frac{\Gamma'_1(f)}{\Gamma'_0(f)} - \zeta \left( \frac{\Pr[f = 1]}{\Gamma'_0(f) ((1 - \eta_0)\widehat{\mu}_1 - \eta_0\widehat{\mu}_0)} + \frac{\Gamma'_1(f)}{(1 - \eta_1) \Pr[f = 1, \widehat{Z} = 0] - \eta_1 \Pr[f = 1, \widehat{Z} = 1]} \right)\end{aligned}$$

$$\geq \frac{\Gamma'_1(f)}{\Gamma'_0(f)} - \delta - \zeta \alpha_1, \quad (\text{Ineqs. 23})$$

where  $\alpha_1 = \frac{\Pr[f=1]}{\Gamma'_0(f)((1-\eta_0)\hat{\mu}_1 - \eta_0\hat{\mu}_0)} + \frac{\Gamma'_1(f)}{(1-\eta_1)\Pr[f=1, \hat{Z}=0] - \eta_1\Pr[f=1, \hat{Z}=1]}$ . Similarly, by Inequalities 19 and 22, we have

$$\frac{\Gamma_0(f)}{\Gamma_1(f)} \geq \frac{\Gamma'_0(f)}{\Gamma'_1(f)} - \delta - \zeta \alpha_0,$$

where  $\alpha_0 = \frac{\Pr[f=1]}{\Gamma'_1(f)((1-\eta_1)\hat{\mu}_0 - \eta_1\hat{\mu}_1)} + \frac{\Gamma'_0(f)}{(1-\eta_0)\Pr[f=1, \hat{Z}=1] - \eta_0\Pr[f=1, \hat{Z}=0]}$ . Thus, we have

$$\gamma^\Delta(f, \hat{S}) \geq \tau - \delta - \zeta \cdot \max\{\alpha_0, \alpha_1\}.$$

The influence of the above inequality is that the fairness guarantee of Theorem 4.3 changes to be

$$\gamma(f^\Delta, S) \geq \tau - 3(\delta + \zeta \cdot \max\{\alpha_0, \alpha_1\}),$$

i.e., the estimation errors will weaken the fairness guarantee of our denoised program. Also, observe that the influence becomes smaller as  $\zeta$  goes to 0.

## B Comparison with theoretical guarantees of [3]

Theorem 4.3 can also be generalized to handle equalized odds, the primary fairness metric of [3]; see Theorem 5.9 in Section 5. Recall that [3] first computes an unconstrained optimal classifier  $\tilde{f}$  and then apply the post-processing algorithm in [29] to achieve a classifier  $\hat{f}$ . By [3, Theorem 1],  $\hat{f}$  must have a smaller bias than  $\tilde{f}$  for any fixed noise parameters  $\eta_0$  and  $\eta_1$  satisfying certain assumptions. There is no guarantee that  $\hat{f}$  achieves a comparable fairness guarantee as  $f^*$ ; as our denoised program.

More concretely, the post-processing approach of [3], given predictions from a base classifier, class labels, protected attribute values for training samples, formulate a linear program to solve for four variables: each variable  $p_{\hat{y},z}$  represents the probability that final prediction should be 1 given that the original prediction is  $\hat{y} \in \{0, 1\}$  and protected attribute value is  $z \in \{0, 1\}$ . In many settings, the output of this linear program is non-unique. For example, suppose that original prediction is random whenever original class label  $Y = 1$ , and is always 1 when  $Y = 0, Z = 0$  and always 0 when  $Y = 0, Z = 1$ , i.e., the false positives are high for  $Z = 0$  group. In this case, the optimal solution is non-unique and, for any  $c > 0$ ,  $p_{0,0}^* = c$ , is part of an optimal solution (as long as total probability is less than 1). While this is not an issue in the normal fair classification scenario, it is problematic when the protected attribute is noisy. The fairness guarantee of the post-processing algorithm (see proof of Thm 1 in [3]) depends on the values  $\{p_{\hat{y},z}\}$ . Concretely, it depends on  $\eta_0 \cdot p_{0,0}^* = \eta_0 \cdot c$  and so the bias guarantee depends on  $c$ . Therefore, in common settings where the solution can be non-unique, the bias guarantee can vary across the solutions and it is not clear if there is a principled way to select the solution that achieves the user-desired fairness guarantee.

## C Discussion of initial attempts

We first discuss two natural ideas including randomized labeling (Section C.1) and solving Program ConFair that only depends on  $\hat{S}$  (Section C.2). We also discuss their weakness on either the empirical loss or the fairness constraints.

### C.1 Randomized labeling

A simple idea is that for each sample  $s_i \in S$ , i.i.d. draw the label  $f(s_i)$  to be 0 with probability  $\alpha$  and to be 1 with probability  $1 - \alpha$  ( $\alpha \in [0, 1]$ ). This simple idea leads to a fair classifier by the following lemma.

**Lemma C.1 (A random classifier is fair)** *Let  $f \in \{0, 1\}^X$  be a classifier generated by randomized labeling. With probability at least  $1 - 2e^{-\frac{\alpha \lambda N}{1.2 \times 10^5}}$ ,  $\gamma(f, S) \geq 0.99$ .*

**Proof:** Let  $A = \{i \in [N] : z_i = 0\}$  be the collection of samples with  $z_i = 0$ . By Assumption 1, we know that  $|A| \geq \lambda N$ . For  $i \in A$ , let  $X_i$  be the random variable where  $X_i = f(s_i)$ . By randomized labeling, we know that  $\Pr[X_i = 1] = \alpha$ . Also,

$$\Pr[f = 1 \mid Z = 0] = \frac{\sum_{i \in A} X_i}{|A|}. \quad (24)$$

Since all  $X_i$  ( $i \in A$ ) are independent, we have

$$\Pr\left[\sum_{i \in A} X_i \in (1 \pm 0.005) \cdot \alpha|A|\right] \geq 1 - 2e^{-\frac{0.005^2 \alpha|A|}{3}} \quad (\text{Chernoff bound}) \geq 1 - 2e^{-\frac{\alpha \lambda N}{1.2 \times 10^5}}. \quad (|A| \geq \lambda N) \quad (25)$$

Thus, with probability at least  $1 - 2e^{-\frac{\alpha \lambda N}{1.2 \times 10^5}}$ ,

$$\begin{aligned} \Pr[f = 1 \mid Z = 0] &= \frac{\sum_{i \in A} X_i}{|A|} && (\text{Eq. 24}) \\ &\in (1 \pm 0.005) \cdot \frac{\alpha|A|}{|A|} && (\text{Ineq. 25}) \\ &\in (1 \pm 0.005)\alpha. \end{aligned}$$

Similarly, we have that with probability at least  $1 - 2e^{-\frac{\alpha \lambda N}{1.2 \times 10^5}}$ ,

$$\Pr[f = 1 \mid Z = 1] \in (1 \pm 0.005)\alpha.$$

By the definition of  $\gamma(f, S)$ , we complete the proof.  $\square$

However, there is no guarantee for the empirical risk of randomized labeling. For instance, consider the loss function  $L(f, s) := \mathbf{I}[f(s) = y]$  where  $\mathbf{I}[\cdot]$  is the indicator function, and suppose there are  $\frac{N}{2}$  samples with  $y_i = 0$ . In this setting, the empirical risk of  $f^*$  may be close to 0, e.g.,  $f^* = Y$ . Meanwhile, the expected empirical risk of randomized labeling is

$$\frac{1}{N} \left( (1 - \alpha) \cdot \frac{N}{2} + \alpha \cdot \frac{N}{2} \right) = \frac{1}{2},$$

which is much larger than that of  $f^*$ .

## C.2 Replacing $S$ by $\hat{S}$ in Program TargetFair

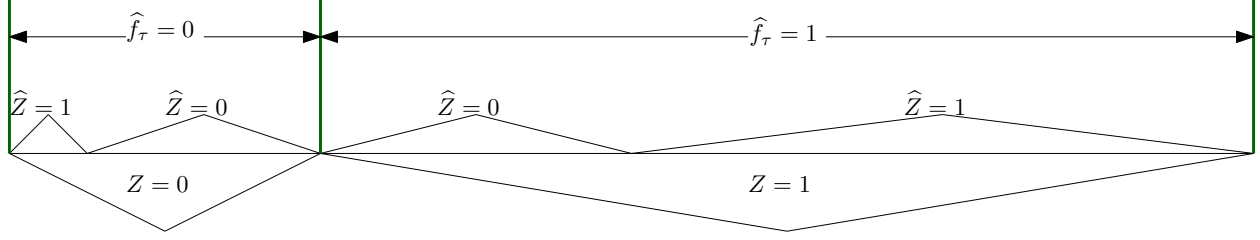
Another idea is to solve the following program which only depends on  $\hat{S}$ , i.e., simply replacing  $S$  by  $\hat{S}$  in Program TargetFair.

$$\begin{aligned} \min_{f \in \mathcal{F}} \frac{1}{N} \sum_{i \in [N]} L(f, \hat{s}_i) \quad & s.t. \\ \gamma(f, \hat{S}) &\geq \tau. \end{aligned} \quad (\text{ConFair})$$

**Remark C.2** Similar to Section 7, we can design an algorithm that solves Program ConFair by logistic regression.

$$\begin{aligned} \min_{\theta \in \mathbb{R}^d} -\frac{1}{N} \sum_{i \in [N]} (y_i \log f_\theta(s_i) + (1 - y_i) \log(1 - f_\theta(s_i))) \quad & s.t. \\ \hat{\mu}_1 \cdot \sum_{i \in [N]: \hat{Z}=0} \mathbf{I}[\langle x_i, \theta \rangle \geq 0] &\geq \tau \hat{\mu}_0 \cdot \sum_{i \in [N]: \hat{Z}=1} \mathbf{I}[\langle x_i, \theta \rangle \geq 0], \\ \hat{\mu}_0 \cdot \sum_{i \in [N]: \hat{Z}=1} \mathbf{I}[\langle x_i, \theta \rangle \geq 0] &\geq \tau \hat{\mu}_1 \cdot \sum_{i \in [N]: \hat{Z}=0} \mathbf{I}[\langle x_i, \theta \rangle \geq 0]. \end{aligned} \quad (\text{FairLR})$$

Figure 2: An example showing that  $\gamma(f, S)$  and  $\gamma(f, \hat{S})$  can differ by a lot. The detailed explanation can be found in Example C.3.



Let  $\hat{f}^*$  denote an optimal solution of Program ConFair. Ideally, we want to use  $\hat{f}^*$  to estimate  $f^*$ . Since  $Z$  is not used for prediction, we have that for any  $f \in \mathcal{F}$ ,

$$\sum_{i \in [N]} L(f, s_i) = \sum_{i \in [N]} L(f, \hat{s}_i).$$

Then if  $\hat{f}^*$  satisfies  $\gamma(\hat{f}^*, S) \geq \tau$ , we conclude that  $\hat{f}^*$  is also an optimal solution of Program TargetFair. However, due to the flipping noises,  $\hat{f}^*$  may be far from  $f^*$  (Example C.3). More concretely, it is possible that  $\gamma(\hat{f}^*, S) \ll \tau$  (Lemma C.4). Moreover, we discuss the range of  $\Omega(f^*, \hat{S})$  (Lemma C.5). We find that  $\Omega(f^*, \hat{S}) < \tau$  may hold which implies that  $f^*$  may not be feasible for Program ConFair. We first give an example showing that  $\hat{f}^*$  can perform very bad over  $S$  with respect to the fairness metric.

**Example C.3** Our example is shown in Figure 2. We assume that  $\mu_0 = 1/3$  and  $\mu_1 = 2/3$ . Let  $\eta = 1/3$  be the noise parameter and we assume  $\pi_{20} = \pi_{01} = 1/3$ . Consequently, we have that

$$\hat{\mu}_0 = 1/3 \times 2/3 + 2/3 \times 1/3 = 4/9.$$

Then we consider the following simple classifier  $f \in \{0, 1\}^{\mathcal{X}}$ :  $\hat{f}^* = Z$ . We directly have that  $\Pr[\hat{f}^* = 1 \mid Z = 0] = 0$  and  $\Pr[\hat{f}^* = 1 \mid Z = 1] = 1$ , which implies that  $\gamma(\hat{f}^*, S) = 0$ . We also have that

$$\begin{aligned} \Pr[\hat{f}^* = 1 \mid \hat{Z} = 0] &= \Pr[Z = 1 \mid \hat{Z} = 0] \quad (\hat{f}^* = Z) \\ &= \frac{\pi_{01} \cdot \mu_1}{\hat{\mu}_0} \quad (\text{Observation 6.1}) = 0.5, \end{aligned}$$

and

$$\begin{aligned} \Pr[\hat{f}^* = 1 \mid \hat{Z} = 1] &= \Pr[Z = 1 \mid \hat{Z} = 1] \quad (\hat{f}^* = Z) \\ &= \frac{\pi_{11} \cdot \mu_1}{\hat{\mu}_1} \quad (\text{Observation 6.1}) = 0.8, \end{aligned}$$

which implies that  $\gamma(\hat{f}^*, \hat{S}) = 0.625$ . Hence, there is a gap between  $\gamma(\hat{f}^*, S)$  and  $\gamma(\hat{f}^*, \hat{S})$ , say 0.625, in this example. Consequently,  $\hat{f}^*$  can be very unfair over  $S$ , and hence, is far from  $f^*$ .

Next, we give some theoretical results showing the weaknesses of Program ConFair.

**An upper bound for  $\gamma(f, S)$ .** More generally, given a classifier  $f \in \{0, 1\}^{\mathcal{X}}$ , we provide an upper bound for  $\gamma(f, S)$  that is represented by  $\gamma(f, \hat{S})$ ; see the following lemma.

**Lemma C.4 (An upper bound for  $\gamma(f, S)$ )** Suppose we have

1.  $\Pr[f = 1 \mid \widehat{Z} = 0] \leq \Pr[f = 1 \mid \widehat{Z} = 1];$
2.  $\Pr[f = 1, Z = 0 \mid \widehat{Z} = 0] \leq \alpha_0 \cdot \Pr[f = 1, Z = 1 \mid \widehat{Z} = 0]$  for some  $\alpha_0 \in [0, 1];$
3.  $\Pr[f = 1, Z = 0 \mid \widehat{Z} = 1] \leq \alpha_1 \cdot \Pr[f = 1, Z = 1 \mid \widehat{Z} = 1]$  for some  $\alpha_1 \in [0, 1].$

Let  $\beta_{ij} = \frac{\widehat{\mu}_i}{\mu_j}$  for  $i, j \in \{0, 1\}$ . The following inequality holds

$$\gamma(f, S) \leq \frac{\alpha_0(1+\alpha_1)\beta_{00}\gamma(f, \widehat{S}) + \alpha_1(1+\alpha_0)\beta_{10}}{(1+\alpha_1)\beta_{01}\gamma(f, \widehat{S}) + (1+\alpha_0)\beta_{11}} \leq \max\{\alpha_0, \alpha_1\} \cdot \frac{\mu_1}{\mu_0}.$$

The intuition of the first assumption is that the statistical rate for  $Z = 0$  is at most that for  $Z = 1$  over the noisy dataset  $\widehat{S}$ . The second and the third assumptions require the classifier  $f$  to be less positive when  $Z = 0$ . Intuitively,  $f$  is restricted to induce a smaller statistical rate for  $Z = 0$  over both  $S$  and  $\widehat{S}$ . Specifically, if  $\alpha_0 = \alpha_1 = 0$  as in Example C.3, we have  $\gamma(f, S) = 0$ . Even if  $\alpha_0 = \alpha_1 = 1$ , we have  $\gamma(f, S) \leq \frac{\mu_1}{\mu_0}$  which does not depend on  $\gamma(f, \widehat{S})$ .

**Proof:** [Proof of Lemma C.4] By the first assumption, we have

$$\gamma(f, \widehat{S}) = \frac{\Pr[f = 1 \mid \widehat{Z} = 0]}{\Pr[f = 1 \mid \widehat{Z} = 1]}. \quad (26)$$

By the second assumption, we have

$$\begin{aligned} \Pr[f = 1, Z = 1 \mid \widehat{Z} = 0] &= \frac{(1 + \alpha_0) \cdot \Pr[f = 1, Z = 1 \mid \widehat{Z} = 0]}{1 + \alpha_0} \\ &\geq \frac{\Pr[f = 1, Z = 1 \mid \widehat{Z} = 0]}{1 + \alpha_0} + \frac{\Pr[f = 1, Z = 0 \mid \widehat{Z} = 0]}{1 + \alpha_0} \\ &= \frac{1}{1 + \alpha_0} \cdot \Pr[f = 1 \mid \widehat{Z} = 0]. \end{aligned} \quad (27)$$

Similarly, we have the following

$$\Pr[f = 1, Z = 0 \mid \widehat{Z} = 0] \leq \frac{\alpha_0}{1 + \alpha_0} \Pr[f = 1 \mid \widehat{Z} = 0]. \quad (28)$$

Also, by the third assumption, we have

$$\Pr[f = 1, Z = 1 \mid \widehat{Z} = 1] \geq \frac{1}{1 + \alpha_1} \Pr[f = 1 \mid \widehat{Z} = 1], \quad (29)$$

and

$$\Pr[f = 1, Z = 0 \mid \widehat{Z} = 1] \leq \frac{\alpha_1}{1 + \alpha_1} \Pr[f = 1 \mid \widehat{Z} = 1]. \quad (30)$$

Then

$$\begin{aligned} \Pr[f = 1 \mid Z = 0] &= \Pr[f = 1, \widehat{Z} = 0 \mid Z = 0] + \Pr[f = 1, \widehat{Z} = 1 \mid Z = 0] \\ &= \Pr[f = 1, Z = 0 \mid \widehat{Z} = 0] \cdot \frac{\widehat{\mu}_0}{\mu_0} + \Pr[f = 1, Z = 0 \mid \widehat{Z} = 1] \cdot \frac{\widehat{\mu}_1}{\mu_0} \\ &= \Pr[f = 1, Z = 0 \mid \widehat{Z} = 0] \cdot \beta_{00} + \Pr[f = 1, Z = 0 \mid \widehat{Z} = 1] \cdot \beta_{10} \\ &\quad (\text{Defn. of } \beta_{00} \text{ and } \beta_{10}) \\ &\leq \frac{\alpha_0 \beta_{00}}{1 + \alpha_0} \cdot \Pr[f = 1 \mid \widehat{Z} = 0] + \frac{\alpha_1 \beta_{10}}{1 + \alpha_1} \cdot \Pr[f = 1 \mid \widehat{Z} = 1]. \\ &\quad (\text{Ineqs. 28 and 30}) \end{aligned} \quad (31)$$

By a similar argument, we have

$$\begin{aligned}
\Pr[f = 1 \mid Z = 1] &= \Pr[f = 1, Z = 1 \mid \widehat{Z} = 0] \cdot \beta_{01} + \Pr[f = 1, Z = 1 \mid \widehat{Z} = 1] \cdot \beta_{11} \\
&\quad (\text{Defn. of } \beta_{01} \text{ and } \beta_{11}) \\
&\geq \frac{\beta_{01}}{1 + \alpha_0} \cdot \Pr[f = 1 \mid \widehat{Z} = 0] + \frac{\beta_{11}}{1 + \alpha_1} \cdot \Pr[f = 1 \mid \widehat{Z} = 1] \\
&\quad (\text{Ineqs. 27 and 29})
\end{aligned} \tag{32}$$

Thus, we have

$$\begin{aligned}
\gamma(f, S) &\leq \frac{\Pr[f = 1 \mid Z = 0]}{\Pr[f = 1 \mid Z = 1]} \quad (\text{Defn. of } \gamma(f, S)) \\
&\leq \frac{\frac{\alpha_0 \beta_{00}}{1 + \alpha_0} \cdot \Pr[f = 1 \mid \widehat{Z} = 0] + \frac{\alpha_1 \beta_{10}}{1 + \alpha_1} \cdot \Pr[f = 1 \mid \widehat{Z} = 1]}{\frac{\beta_{01}}{1 + \alpha_0} \cdot \Pr[f = 1 \mid \widehat{Z} = 0] + \frac{\beta_{11}}{1 + \alpha_1} \cdot \Pr[f = 1 \mid \widehat{Z} = 1]} \\
&\quad (\text{Ineqs. 31 and 32}) \\
&= \frac{\alpha_0(1 + \alpha_1)\beta_{00} \cdot \gamma(f, \widehat{S}) + \alpha_1(1 + \alpha_0)\beta_{10}}{(1 + \alpha_1)\beta_{01} \cdot \gamma(f, \widehat{S}) + (1 + \alpha_0)\beta_{11}} \quad (\text{Eq. 26}) \\
&\leq \max \left\{ \alpha_0 \cdot \frac{\beta_{00}}{\beta_{01}}, \alpha_1 \cdot \frac{\beta_{10}}{\beta_{11}} \right\} \\
&= \max \{ \alpha_0, \alpha_1 \} \cdot \frac{\mu_1}{\mu_0}, \quad (\text{Defn. of } \beta_{ij})
\end{aligned}$$

which completes the proof.  $\square$

**$f^*$  may not be feasible in Program ConFair.** We consider a simple case that  $\eta_1 = \eta_2 = \eta$ . Without loss of generality, we assume that  $\Pr[f^* = 1 \mid Z = 0] \leq \Pr[f^* = 1 \mid Z = 1]$ , i.e., the statistical rate of  $Z = 0$  is smaller than that of  $Z = 1$  over  $S$ . Consequently, we have

$$\gamma(f^*, S) = \frac{\Pr[f^* = 1 \mid Z = 0]}{\Pr[f^* = 1 \mid Z = 1]}.$$

**Lemma C.5 (Range of  $\Omega(f^*, \widehat{S})$ )** *Let  $\varepsilon \in (0, 0.5)$  be a given constant and let*

$$\Gamma = \frac{\eta\mu_0 + (1 - \eta)(1 - \mu_0)}{(1 - \eta)\mu_0 + \eta(1 - \mu_0)} \cdot \frac{(1 - \eta)\mu_0\gamma(f^*, S) + \eta(1 - \mu_0)}{\eta\mu_0\gamma(f^*, S) + (1 - \eta)(1 - \mu_0)}.$$

*With probability at least  $1 - 4e^{-\frac{\varepsilon^2 \eta \lambda N}{192}}$ , the following holds*

$$\gamma(f^*, \widehat{S}) \in (1 \pm \varepsilon) \cdot \min \left\{ \Gamma, \frac{1}{\Gamma} \right\}.$$

For instance, if  $\mu_0 = 0.5$ ,  $\gamma(f^*, S) = 0.8 = \tau$  and  $\eta = 0.2$ , we have

$$\gamma(f^*, \widehat{S}) \approx 0.69 < \tau.$$

Then  $f^*$  is not a feasible solution of Program ConFair. Before proving the lemma, we give some intuitions.

**Discussion C.6** *By Definition 3.1, we have that for a given classifier  $f^* \in \mathcal{F}$ ,*

$$\Pr[\widehat{Z} = 1 \mid Z = 0] \approx \Pr[\widehat{Z} = 0 \mid Z = 1] \approx \eta \tag{33}$$

Moreover, the above property also holds when conditioned on a subset of samples with  $Z = 0$  or  $Z = 1$ . Specifically, for  $i \in \{0, 1\}$ ,

$$\begin{aligned} & \Pr \left[ \widehat{Z} = 1 \mid f^* = 1, Z = 0 \right] \\ & \approx \Pr \left[ \widehat{Z} = 0 \mid f^* = 1, Z = 1 \right] \approx \eta \end{aligned} \quad (34)$$

Another consequence of Property 33 is that for  $i \in \{0, 1\}$ ,

$$\begin{aligned} \widehat{\mu}_i &= \pi_{i,i}\mu_i + \pi_{i,1-i}\mu_{1-i} \quad (\text{Observation 6.1}) \\ &\approx (1 - \eta)\mu_i + \eta\mu_{1-i}. \quad (\text{Property 33}) \end{aligned} \quad (35)$$

Then we have

$$\begin{aligned} & \Pr \left[ f^* = 1 \mid \widehat{Z} = 0 \right] = \Pr \left[ f^* = 1, Z = 0 \mid \widehat{Z} = 0 \right] + \Pr \left[ f^* = 1, Z = 1 \mid \widehat{Z} = 0 \right] \\ = & \Pr \left[ Z = 0 \mid \widehat{Z} = 0 \right] \cdot \Pr \left[ f^* = 1 \mid Z = 0, \widehat{Z} = 0 \right] + \Pr \left[ Z = 1 \mid \widehat{Z} = 0 \right] \cdot \Pr \left[ f^* = 1 \mid Z = 1, \widehat{Z} = 0 \right] \\ = & \frac{\pi_{00}\mu_0}{\widehat{\mu}_0} \cdot \Pr \left[ f^* = 1 \mid Z = 0, \widehat{Z} = 0 \right] + \frac{\pi_{01}\mu_1}{\widehat{\mu}_0} \cdot \Pr \left[ f^* = 1 \mid Z = 1, \widehat{Z} = 0 \right] \\ & (\text{Observation 6.1}) \\ \approx & \frac{(1 - \eta)\mu_0}{(1 - \eta)\mu_0 + \eta\mu_1} \cdot \Pr \left[ f^* = 1 \mid Z = 0, \widehat{Z} = 0 \right] + \frac{\eta\mu_1}{(1 - \eta)\mu_0 + \eta\mu_1} \cdot \Pr \left[ f^* = 1 \mid Z = 1, \widehat{Z} = 0 \right] \\ & (\text{Properties 33 and 35}) \\ = & \frac{(1 - \eta)\mu_0}{(1 - \eta)\mu_0 + \eta(1 - \mu_0)} \times \frac{\Pr \left[ f^* = 1 \mid Z = 0 \right] \cdot \Pr \left[ \widehat{Z} = 0 \mid f^* = 1, Z = 0 \right]}{\Pr \left[ \widehat{Z} = 0 \mid Z = 0 \right]} \\ & + \frac{\eta\mu_1}{(1 - \eta)\mu_0 + \eta(1 - \mu_0)} \times \frac{\Pr \left[ f^* = 1 \mid Z = 1 \right] \cdot \Pr \left[ \widehat{Z} = 0 \mid f^* = 1, Z = 1 \right]}{\Pr \left[ \widehat{Z} = 0 \mid Z = 1 \right]} \\ \approx & \frac{(1 - \eta)\mu_0}{(1 - \eta)\mu_0 + \eta(1 - \mu_0)} \cdot \Pr \left[ f^* = 1 \mid Z = 0 \right] + \frac{\eta\mu_1}{(1 - \eta)\mu_0 + \eta(1 - \mu_0)} \cdot \Pr \left[ f^* = 1 \mid Z = 1 \right]. \\ & (\text{Properties 33 and 34}) \end{aligned}$$

Similarly, we can represent

$$\begin{aligned} & \Pr \left[ f^* = 1 \mid \widehat{Z} = 1 \right] \\ \approx & \frac{\eta\mu_0}{\eta\mu_0 + (1 - \eta)(1 - \mu_0)} \Pr \left[ f^* = 1 \mid Z = 0 \right] + \frac{(1 - \eta)\mu_1}{\eta\mu_0 + (1 - \eta)(1 - \mu_0)} \Pr \left[ f^* = 1 \mid Z = 1 \right]. \end{aligned}$$

Applying the approximate values of  $\Pr \left[ f^* = 1 \mid \widehat{Z} = 0 \right]$  and  $\Pr \left[ f^* = 1 \mid \widehat{Z} = 1 \right]$  to compute  $\gamma(f^*, S)$ , we have Lemma C.5.

**Proof:** [Proof of Lemma C.5] By definition, we have

$$\gamma(f^*, \widehat{S}) \leq \frac{\Pr \left[ f^* = 1 \mid \widehat{Z} = 0 \right]}{\Pr \left[ f^* = 1 \mid \widehat{Z} = 1 \right]}.$$



Thus, it suffices to provide an upper bound for  $\Pr[f^* = 1 \mid \widehat{Z} = 0]$  and a lower bound for  $\Pr[f^* = 1 \mid \widehat{Z} = 1]$ . Similar to Discussion C.6, we have

$$\begin{aligned}
\Pr[f^* = 1 \mid \widehat{Z} = 0] &= \frac{\Pr[Z = 0] \cdot \Pr[f^* = 1 \mid Z = 0]}{\Pr[\widehat{Z} = 0]} \times \Pr[\widehat{Z} = 0 \mid f^* = 1, Z = 0] \\
&\quad + \frac{\Pr[Z = 1] \cdot \Pr[f^* = 1 \mid Z = 1]}{\Pr[\widehat{Z} = 0]} \times \Pr[\widehat{Z} = 0 \mid f^* = 1, Z = 1] \\
&= \frac{\mu_0 \cdot \Pr[f^* = 1 \mid Z = 0]}{\pi_{00}\mu_0 + \pi_{01}(1 - \mu_0)} \times \Pr[\widehat{Z} = 0 \mid f^* = 1, Z = 0] \\
&\quad + \frac{\mu_1 \cdot \Pr[f^* = 1 \mid Z = 1]}{\pi_{00}\mu_0 + \pi_{01}(1 - \mu_0)} \times \Pr[\widehat{Z} = 0 \mid f^* = 1, Z = 1],
\end{aligned} \tag{36}$$

and

$$\begin{aligned}
\Pr[f^* = 1 \mid \widehat{Z} = 1] &= \frac{\Pr[Z = 0] \cdot \Pr[f^* = 1 \mid Z = 0]}{\Pr[\widehat{Z} = 1]} \times \Pr[\widehat{Z} = 1 \mid f^* = 1, Z = 0] \\
&\quad + \frac{\Pr[Z = 1] \cdot \Pr[f^* = 1 \mid Z = 1]}{\Pr[\widehat{Z} = 1]} \times \Pr[\widehat{Z} = 1 \mid f^* = 1, Z = 1] \\
&= \frac{\mu_0 \cdot \Pr[f^* = 1 \mid Z = 0]}{\pi_{11}(1 - \mu_0) + \pi_{20}\mu_0} \times \Pr[\widehat{Z} = 1 \mid f^* = 1, Z = 0] \\
&\quad + \frac{\mu_1 \cdot \Pr[f^* = 1 \mid Z = 1]}{\pi_{11}(1 - \mu_0) + \pi_{20}\mu_0} \times \Pr[\widehat{Z} = 1 \mid f^* = 1, Z = 1],
\end{aligned} \tag{37}$$

We then analyze the right side of the Equation 36. We take the term  $\Pr[\widehat{Z} = 0 \mid f^* = 1, Z = 1]$  as an example. Let  $A = \{i \in [n] : f^*(s_i) = 1, z_i = 0\}$ . By Assumption 1, we have  $|A| \geq \lambda N$ . For  $i \in A$ , let  $X_i$  be the random variable where  $X_i = 1 - \widehat{z}_i$ . By Definition 3.1, we know that  $\Pr[X_i = 1] = \eta$ . Also,

$$\Pr[\widehat{Z} = 0 \mid f^* = 1, Z = 1] = \frac{\sum_{i \in A} X_i}{|A|}. \tag{38}$$

Since all  $X_i$  ( $i \in A$ ) are independent, we have

$$\begin{aligned}
\Pr\left[\sum_{i \in A} X_i \in (1 \pm \frac{\varepsilon}{8}) \cdot \eta|A|\right] &\geq 1 - 2e^{-\frac{\varepsilon^2 \eta |A|}{192}} \quad (\text{Chernoff bound}) \\
&\geq 1 - 2e^{-\frac{\varepsilon^2 \eta \lambda N}{192}}. \quad (|A| \geq \lambda N)
\end{aligned} \tag{39}$$

Thus, with probability at least  $1 - 2e^{-\frac{\varepsilon^2 \eta \lambda N}{192}}$ ,

$$\begin{aligned}
\Pr[\widehat{Z} = 0 \mid f^* = 1, Z = 1] &= \frac{\sum_{i \in A} X_i}{|A|} \quad (\text{Eq. 38}) \\
&\in (1 \pm \frac{\varepsilon}{8}) \cdot \frac{\eta|A|}{|A|} \quad (\text{Ineq. 39}) \\
&\in (1 \pm \frac{\varepsilon}{8})\eta.
\end{aligned}$$

Consequently, we have

$$\Pr[\widehat{Z} = 1 \mid f^* = 1, Z = 1] = 1 - \Pr[\widehat{Z} = 0 \mid f^* = 1, Z = 1]$$

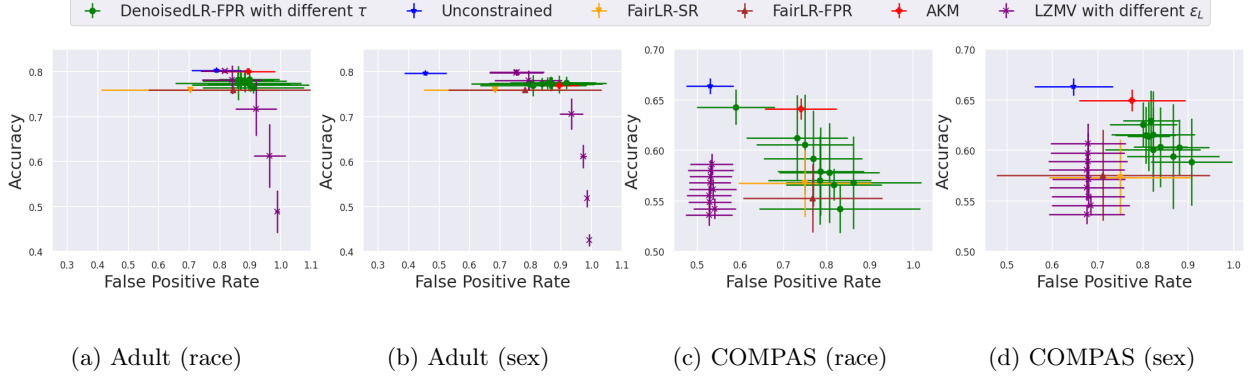


Figure 3: Performance of **DenoisedLR-FPR** and baselines with respect to false positive rate and accuracy for different combinations of dataset and protected attribute. For **DenoisedLR-FPR**, the performance for different  $\tau$  is plotted to present the entire fairness-accuracy tradeoff picture. Similarly, for **LZMV** the input parameter  $\varepsilon_L$  is varied. The plots shows that for all settings **FPR** can attain a high false positive rate, often with minimal loss in accuracy.

$$\begin{aligned} &\in 1 - (1 \pm \frac{\varepsilon}{8})\eta && (\text{Ineq. 40}) \\ &\in (1 \pm \frac{\varepsilon}{8})(1 - \eta) && (\eta < 0.5) \end{aligned}$$

Similarly, we can prove that with probability at least  $1 - 4e^{-\frac{\varepsilon^2 \eta \lambda N}{192}}$ ,

- $\pi_{01}, \pi_{20}, \Pr \left[ \widehat{Z} = 1 \mid f^* = 1, Z = 0 \right], \Pr \left[ \widehat{Z} = 0 \mid f^* = 1, Z = 1 \right] \in (1 \pm \frac{\varepsilon}{8})\eta;$
- $\pi_{00}, \pi_{11}, \Pr \left[ \widehat{Z} = 0 \mid f^* = 1, Z = 0 \right], \Pr \left[ \widehat{Z} = 1 \mid f^* = 1, Z = 1 \right] \in (1 \pm \frac{\varepsilon}{8})(1 - \eta).$

Applying these inequalities to Equations 36 and 37, we have that with probability at least  $1 - 4e^{-\frac{\varepsilon^2 \eta \lambda N}{192}}$ ,

$$\begin{aligned} \frac{\Pr \left[ f^* = 1 \mid \widehat{Z} = 0 \right]}{\Pr \left[ f^* = 1 \mid \widehat{Z} = 1 \right]} &\in (1 \pm \varepsilon) \cdot \frac{\eta \mu_0 + (1 - \eta)(1 - \mu_0)}{(1 - \eta)\mu_0 + \eta(1 - \mu_0)} \times \frac{(1 - \eta)\mu_0 \gamma(f^*, S) + \eta(1 - \mu_0)}{\eta \mu_0 \gamma(f^*, S) + (1 - \eta)(1 - \mu_0)} \\ &\in (1 \pm \varepsilon) \cdot \Gamma, \end{aligned}$$

and

$$\frac{\Pr \left[ f^* = 1 \mid \widehat{Z} = 1 \right]}{\Pr \left[ f^* = 1 \mid \widehat{Z} = 0 \right]} \in (1 \pm \varepsilon) \cdot \frac{1}{\Gamma}.$$

By the definition of  $\gamma(f^*, \widehat{S})$ , we complete the proof.  $\square$

## D Other empirical details and results

We state the exact empirical form of the constraints used for our simulations in this section and then present additional empirical results.

### D.1 Implementation of our denoised algorithm.

As a use case, we solve Program DenoisedFair for logistic regression. Let  $\mathcal{F}' = \{f'_\theta \mid \theta \in \mathbb{R}^d\}$  be the family of logistic regression classifiers where for each sample  $s = (x, z, y)$ ,  $f'_\theta(s) := \frac{1}{1 + e^{-\langle x, \theta \rangle}}$ . We learn a classifier  $f'_\theta \in \mathcal{F}'$  and then round each  $f'_\theta(\widehat{s}_i)$  to  $f_\theta(\widehat{s}_i) := \mathbf{I}[f(\widehat{s}_i) \geq 0.5]$ .

**Program DenoisedFair for statistical rate metric.** We first show how to implement the Program DenoisedFair for statistical rate constraints. We can represent  $\Pr[f_\theta = 1, \widehat{Z} = i]$  as

$$\Pr[f_\theta = 1, \widehat{Z} = i] = \frac{1}{N} \sum_{i \in [N]: \widehat{Z}=i} \mathbf{I}[\langle x_i, \theta \rangle \geq 0].$$

Recall that  $\widehat{\mu}_i := \Pr_{\widehat{D}}[\widehat{Z} = i]$  for  $i \in \{0, 1\}$ . Then let  $\mu'_0 := (1 - \eta_1)\widehat{\mu}_0 - \eta_1\widehat{\mu}_1$  and  $\mu'_1 := (1 - \eta_0)\widehat{\mu}_1 - \eta_0\widehat{\mu}_0$ . Constraint 2 can be written as

$$\left\{ \begin{array}{l} \frac{1-\eta_1}{N} \sum_{i \in [N]: \widehat{Z}=0} \mathbf{I}[\langle x_i, \theta \rangle \geq 0] \\ - \frac{\eta_1}{N} \sum_{i \in [N]: \widehat{Z}=1} \mathbf{I}[\langle x_i, \theta \rangle \geq 0] \geq (1 - \eta_0 - \eta_1)\lambda - \delta, \\ \frac{1-\eta_0}{N} \sum_{i \in [N]: \widehat{Z}=1} \mathbf{I}[\langle x_i, \theta \rangle \geq 0] \\ - \frac{\eta_0}{N} \sum_{i \in [N]: \widehat{Z}=0} \mathbf{I}[\langle x_i, \theta \rangle \geq 0] \geq (1 - \eta_0 - \eta_1)\lambda - \delta, \\ ((\tau - \delta)\eta_0\mu'_0 + (1 - \eta_1)\mu'_1) \sum_{i \in [N]: \widehat{Z}=0} \mathbf{I}[\langle x_i, \theta \rangle \geq 0] \\ \geq ((\tau - \delta)(1 - \eta_0)\mu'_0 + \eta_1\mu'_1) \sum_{i \in [N]: \widehat{Z}=1} \mathbf{I}[\langle x_i, \theta \rangle \geq 0], \\ ((\tau - \delta)\eta_1\mu'_1 + (1 - \eta_0)\mu'_0) \sum_{i \in [N]: \widehat{Z}=1} \mathbf{I}[\langle x_i, \theta \rangle \geq 0] \\ \geq ((\tau - \delta)(1 - \eta_1)\mu'_1 + \eta_0\mu'_0) \sum_{i \in [N]: \widehat{Z}=0} \mathbf{I}[\langle x_i, \theta \rangle \geq 0]. \end{array} \right. \quad (40)$$

Now we propose the following program that minimizes the logistic loss.

$$\begin{aligned} \min_{\theta \in \mathbb{R}^d} & -\frac{1}{N} \sum_{i \in [N]} (y_i \log f_\theta(s_i) + (1 - y_i) \log(1 - f_\theta(s_i))) \\ \text{s.t.} & \text{ 41.} \end{aligned} \quad (\text{DenoisedLR-SR})$$

**Program DenoisedFair for false positive rate metric.** Next, we show how to implement the Program DenoisedFair for false positive rate constraints. We can represent  $\Pr[f_\theta = 1, \widehat{Z} = i, Y = 0]$  as

$$\Pr[f_\theta = 1, \widehat{Z} = i, Y = 0] = \frac{1}{N} \sum_{i \in [N]: \widehat{Z}=i, Y=0} \mathbf{I}[\langle x_i, \theta \rangle \geq 0].$$

Once again  $\widehat{\mu}_i := \Pr_{\widehat{D}}[\widehat{Z} = i, Y = 0]$  for  $i \in \{0, 1\}$ . Then let  $\mu'_0 := (1 - \eta_1)\widehat{\mu}_0 - \eta_1\widehat{\mu}_1$  and  $\mu'_1 := (1 - \eta_0)\widehat{\mu}_1 - \eta_0\widehat{\mu}_0$ . Constraint 2 can be written as

$$\left\{ \begin{array}{l} \frac{1-\eta_1}{N} \sum_{i \in [N]: \widehat{Z}=0, Y=0} \mathbf{I}[\langle x_i, \theta \rangle \geq 0] \\ - \frac{\eta_1}{N} \sum_{i \in [N]: \widehat{Z}=1, Y=0} \mathbf{I}[\langle x_i, \theta \rangle \geq 0] \geq (1 - \eta_0 - \eta_1)\lambda - \delta, \\ \frac{1-\eta_0}{N} \sum_{i \in [N]: \widehat{Z}=1, Y=0} \mathbf{I}[\langle x_i, \theta \rangle \geq 0] \\ - \frac{\eta_0}{N} \sum_{i \in [N]: \widehat{Z}=0, Y=0} \mathbf{I}[\langle x_i, \theta \rangle \geq 0] \geq (1 - \eta_0 - \eta_1)\lambda - \delta, \\ ((\tau - \delta)\eta_0\mu'_0 + (1 - \eta_1)\mu'_1) \sum_{i \in [N]: \widehat{Z}=0, Y=0} \mathbf{I}[\langle x_i, \theta \rangle \geq 0] \\ \geq ((\tau - \delta)(1 - \eta_0)\mu'_0 + \eta_1\mu'_1) \sum_{i \in [N]: \widehat{Z}=1, Y=0} \mathbf{I}[\langle x_i, \theta \rangle \geq 0], \\ ((\tau - \delta)\eta_1\mu'_1 + (1 - \eta_0)\mu'_0) \sum_{i \in [N]: \widehat{Z}=1, Y=0} \mathbf{I}[\langle x_i, \theta \rangle \geq 0] \\ \geq ((\tau - \delta)(1 - \eta_1)\mu'_1 + \eta_0\mu'_0) \sum_{i \in [N]: \widehat{Z}=0, Y=0} \mathbf{I}[\langle x_i, \theta \rangle \geq 0]. \end{array} \right. \quad (41)$$

Now we propose the following program that minimizes the logistic loss.

$$\begin{aligned} \min_{\theta \in \mathbb{R}^d} & -\frac{1}{N} \sum_{i \in [N]} (y_i \log f_\theta(s_i) + (1 - y_i) \log(1 - f_\theta(s_i))) \\ \text{s.t.} & \text{ 41.} \end{aligned} \quad (\text{DenoisedLR-FPR})$$

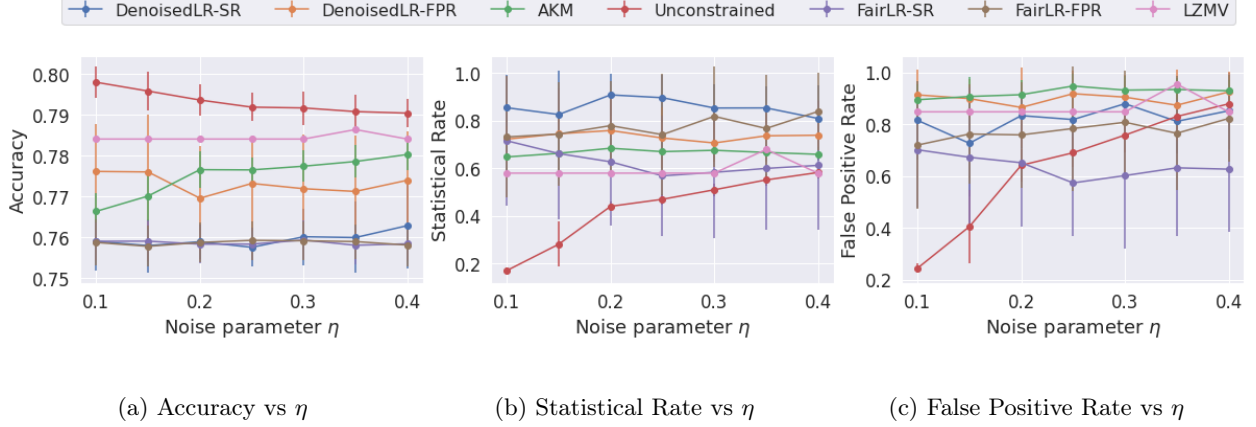


Figure 4: Performance of **DenoisedLR-SR**, **DenoisedLR-FPR** ( $\tau = 0.9$ ) and baselines with respect to statistical rate, false positive rate and accuracy for different noise parameters  $\eta$ . The dataset used is **Adult** and the protected attribute is sex.

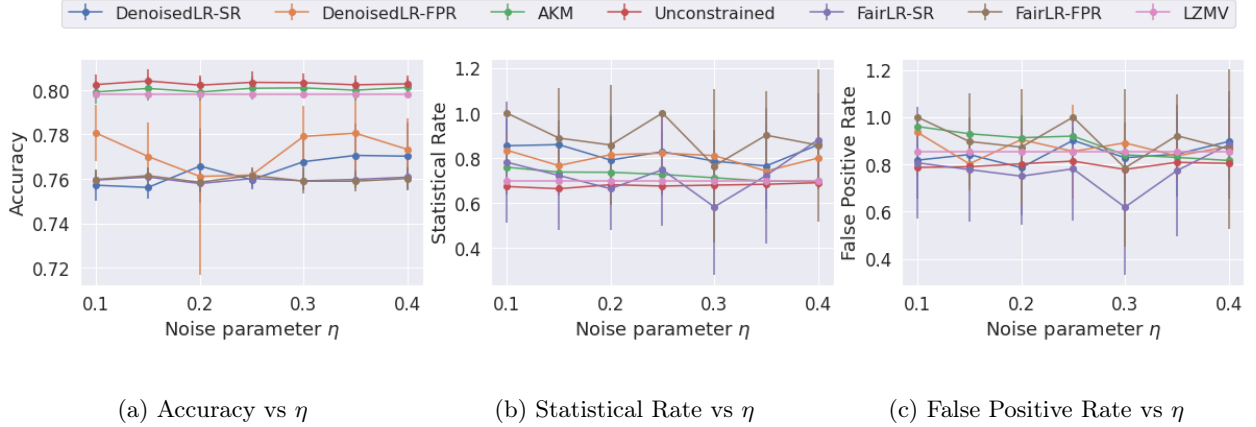


Figure 5: Performance of **DenoisedLR-SR**, **DenoisedLR-FPR** ( $\tau = 0.9$ ) and baselines with respect to statistical rate, false positive rate and accuracy for different noise parameters  $\eta$ . The dataset used is **Adult** and the protected attribute is race.

Sometimes, we may append a regularization term  $C \cdot \|\theta\|_2^2$  to the above loss function where  $C \geq 0$  is a given regularization parameter. We can apply some constrained optimization packages to solve this program, e.g., SLSQP [38].

## D.2 Other results

In this section, we present other empirical results to complement the arguments made in Section 7. First, we present the plot for comparison of all methods with respect to false positive rate, Figure 3.

### D.2.1 Variation of noise parameter

We also investigate the performances of algorithms w.r.t. varying  $\eta_0, \eta_1$ . We consider  $\eta_0 = \eta_1 = \eta \in \{0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4\}$ . Other settings are the same as in the main text. We select  $\tau = 0.9$  for **FairLR** and **DenoisedLR**. The performance on Adult dataset is presented in Figure 4 when sex is the protected attribute and in Figure 5 when race is the protected attribute. The performance on COMPAS dataset is presented in Figure 6 when sex is the protected attribute and in Figure 7 when race is the protected attribute.

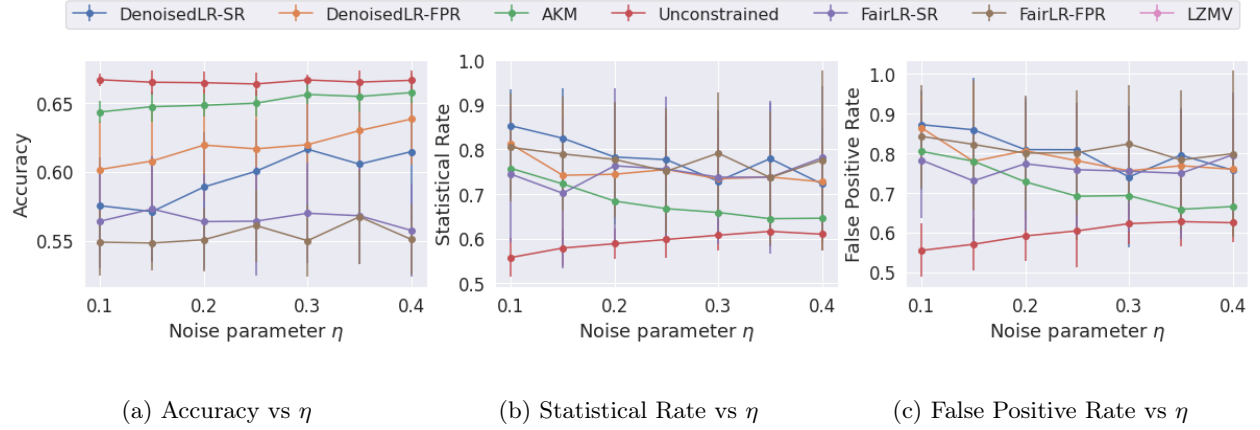


Figure 6: Performance of **DenoisedLR-SR**, **DenoisedLR-FPR** ( $\tau = 0.9$ ) and baselines with respect to statistical rate, false positive rate and accuracy for different noise parameters  $\eta$ . The dataset used is **COMPAS** and the protected attribute is sex.

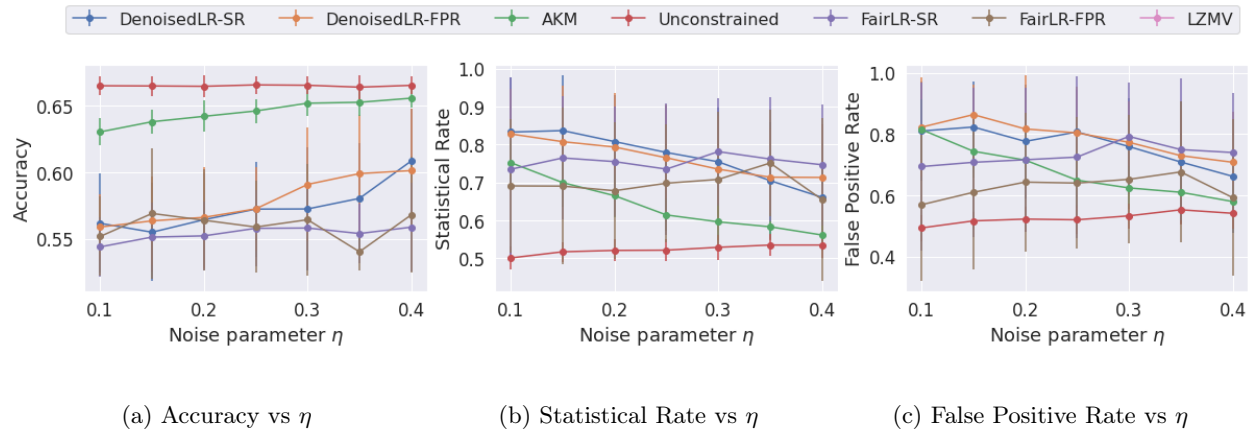


Figure 7: Performance of **DenoisedLR-SR**, **DenoisedLR-FPR** ( $\tau = 0.9$ ) and baselines with respect to statistical rate, false positive rate and accuracy for different noise parameters  $\eta$ . The dataset used is **COMPAS** and the protected attribute is race.

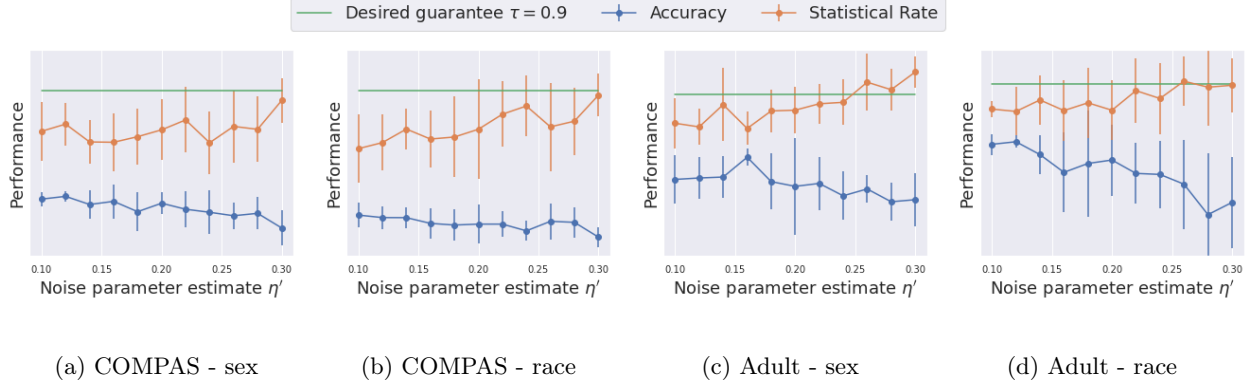


Figure 8: Performance of **DenoisedLR-SR** ( $\tau = 0.9$ ) with respect to statistical rate and accuracy for different noise parameter estimate  $\eta'$ . The true noise parameters are  $\eta_0 = \eta_1 = 0.3$ .

For COMPAS dataset, the plots show that for all noise parameters and both attributes, **DenoisedLR-SR** and **DenoisedLR-FPR** achieve high fairness (statistical and false positive rate) at a lower cost to accuracy than other baselines. For the Adult dataset with race as the protected attribute, once again **DenoisedLR-SR** and **DenoisedLR-FPR** achieve high fairness at a relatively low cost to accuracy. When sex is the protected attribute, **AKM** performs relatively better with respect to false positive rate; however, for statistical rate fairness metric, **DenoisedLR-SR** achieves the highest fairness value.

### D.2.2 Error in noise parameter estimation

As discussed in Section 4.3, the scale of error in the noise parameter estimation can affect the fairness guarantees. In this section, we empirically look at the impact of estimation error on the statistical rate of the generated classifier.

We set the true noise parameters  $\eta_0 = \eta_1 = 0.3$ . The estimated noise parameter ranges  $\eta'$  ranges from 0.1 to 0.3. The variation of accuracy and statistical rate with noise parameter estimate of **DenoisedLR-SR** for COMPAS and Adult datasets is presented in Figure 8a,b. The plots show that, for both protected attributes, the best statistical rate (close to the desired guarantee of 0.90) is achieved when the estimate matches the true noise parameter value. However, even for estimates that are even considerably lower than true estimate (for instance,  $\eta' < 0.15$ ), the average statistical rate is still quite high ( $\sim 0.80$ ).

The results shows that if the error in noise parameter estimate is reasonable, the framework ensures that the fairness of the generated classifier is still high.