Popularizing Fairness: Group Fairness and Individual Welfare

Andrew Estornell[1], **Sanmay Das**[2], **Brendan Juba**[1] **Yevgeniy Vorobeychik**[1]

[1] Washington University in Saint Louis, [2] George Mason University

Abstract

Group-fair learning methods typically seek to ensure that some measure of prediction efficacy for (often historically) disadvantaged minority groups is comparable to that for the majority of the population. When a principal seeks to adopt a group-fair approach to replace another, more conventional approach, the principal may face opposition from those who feel that they have been disadvantaged as a result of the switch, and this, in turn, may deter adoption. We propose to mitigate this concern by ensuring that a group-fair model is also *popular*, in the sense that it yields a preferred distribution over outcomes compared with the conventional model for a majority of the target population. First, we show that state of the art fair learning approaches are often unpopular in this sense. We then present several efficient algorithms for postprocessing an existing group-fair learning scheme to improve its popularity while retaining fairness. Through extensive experiments, we demonstrate that the proposed postprocessing approaches are highly effective.

1 Introduction

Increasing adoption of machine learning approaches in high-stakes domains, such as healthcare and social assistance, has led to increased scrutiny of their impact on vulnerable groups. A number of studies demonstrating the disparate impact of automation on such groups (Citron and Pasquale 2014; Angwin et al. 2016; Dastin 2018; Lee 2018; Koenecke et al. 2020) has motivated an extensive literature that aims at achieving group fairness of machine learning (Kearns et al. 2018; Agarwal et al. 2018; Pleiss et al. 2017; Hardt, Price, and Srebro 2016; Chouldechova and Roth 2018; Mehrabi et al. 2021; Barocas, Hardt, and Narayanan 2017; Angwin et al. 2016; Dwork et al. 2012) by imposing an explicit constraint that prediction efficacy (which can be measured in many different ways) is similar across groups. However, a principal contemplating a change from a conventional, and potentially biased, prediction model to a group-fair approach must contend with the perception that such a switch could inadvertently harm many individuals in the process of improving fairness (for example, making them less likely to receive a scarce resource such as a welfare benefit or admission to a college). Such perceptions could make any change from the status quo contentious and, consequently, less likely. Our central question is whether it is possible to make group-fair classifiers sufficiently *popular*—reducing the prevalence of realized or perceived harm—so as to make their adoption less contested.

We model the principal's problem as a comparison between a conventional approach f_C and a group-fair approach f_F, with the principal considering a switch from the former to the latter. Both algorithms select a subset of individuals from a target population to obtain a particular desirable outcome (e.g., a resource, such as admission to a college). We examine popularity in this context through the lens of preferences of individuals in a target population over selection outcomes (which we can encode as positive outcomes of binary classification): an individual *weakly* prefers f_F to f_C if the probability of being selected is not lower under the former than under the latter. Popularity of a group-fair approach f_F then amounts to ensuring that a given fraction (e.g., majority) of a target population prefers f_F to f_C.

To illustrate the relationship between fairness, accuracy, and popularity, consider the following example. Let G_1 and G_0 have four and two members respectively, with true labels $\langle 1, 1, 1, 1\rangle$ and $\langle 0, 0\rangle$. A randomized conventional classifier f_C, predicts each member of G_1 to be positive with probability 0.75 and each member of G_0 to be positive with probability 0.25. Under demographic parity fairness, G_1 is advantaged as this group has a positive rate 0.5 greater than that of G_0. Consider two choices for a fair model. f_{F_1} predicts members of G_1 to be positive with probability 0.75 and members of G_0 to be positive with probability 0.55. f_{F_2} predicts one member of G_1 to be positive with probability 1, and the others with probability $\frac{2}{3}$; it predicts one member of G_0 to be positive with probability 1 and the other with probability 0.1. Note that both models have identical accuracy and unfairness, namely .65 and .2 respectively. However, f_{F_1} has *not* decreased the score of any agent in the population; all six prefer f_{F_1} at least as much as the original f_C. In contrast, f_{F_2} has decreased the scores of three agents from G_1 and one agent from G_0; only two agents prefer f_{F_2} at least as much as f_C. This example illustrates that popularity should be viewed as a different axis than either accuracy or fairness, and there may be space to innovate by enabling popularity comparisons among fair(er) models.

We start this paper by asking an empirical question: Do typical group-fair classification approaches yield models that are, in fact, unpopular in the sense above? We demonstrate that they are: in experiments on several standard datasets, more than half the target population can strictly prefer the conventional scheme to several prominent group-fair learning methods. Given that the group-fair approaches have significant motivation and momentum behind them, instead of designing an entirely new approach to finding popular and fair classifier, we ask whether it is possible to *minimally postprocess* the output of a group-fair classifier in order to achieve some target popularity while maintaining a high level of fairness. We answer this question in the affirmative. Specifically, we describe two approaches to efficiently postprocess the outputs from a given group-fair classifier in order to boost its popularity. The first approach formalizes the problem as a minimal change of outcome probabilities over the target population to guarantee a target level of fairness and popularity. We show that this problem can be solved in polynomial time. Our second approach involves a form of regularized empirical risk minimization with fairness and popularity constraints. This approach relies on partitioning prediction scores into a set of quantiles, and we show that, in general, the problem is strongly NP-Hard. However, we also show that if the number of quantiles is constant, this problem can be solved in polynomial time. Our methods are applicable in both the classification and scarce resource allocation settings, and allow a model designer to directly control the level of popularity and fairness.

In summary, our contributions are:

1. We propose the notion of *popularity* of group-fair classifiers and allocation schemes, measuring the fraction of a population that is weakly better off when switching from a conventional to a fair learning scheme.
2. We demonstrate the degree to which state of the art group-fair approaches are *unpopular* compared to their conventional counterparts.
3. We introduce two postprocessing algorithms which allow a principal to directly control the popularity of a given fair model, while maintaining good fairness properties. The first post-processing technique, dubbed DOS (Direct Outcome Shift), is polynomial time solvable for both deterministic and randomized classifiers, and can also be applied to the scarce resource allocation setting. The second technique, k-QLS (k-Quantile Lottery Shift), works by grouping agents into k quantiles (where k is chosen by the model designer), and running lotteries on each quantile. k-QLS is polynomial time solvable for deterministic classifiers. While we show that k-QLS is NP-hard in the randomized case, it becomes tractable for constant k, as would be standard in practice.
4. We empirically demonstrate that the proposed postprocessing techniques can achieve high levels of popularity and fairness with minimal impact on prediction accuracy.

Related Work: Our work is broadly related to the field of algorithmic group fairness which is concerned with both defining what it means for a model to be fair, as well as operationalizing these definitions to produce fair models (Hardt, Price, and Srebro 2016; Pleiss et al. 2017; Feldman et al. 2015; Dwork et al. 2012; Agarwal et al. 2018; Kearns et al. 2018; Jang, Shi, and Wang 2021; Kusner et al. 2017). In particular our algorithms work through postprocessing, a common technique for for achieving fairness (Pleiss et al. 2017; Hardt, Price, and Srebro 2016; Kamiran, Karim, and Zhang 2012; Canetti et al. 2019; Lohia et al. 2019; Jang, Shi, and Wang 2021). In these works, the scores or decisions of a conventional classifier are modified in order to achieve fairness. Most post processing techniques for fairness work through “inclusion/exclusion” systems where a potentially randomized procedure is uniformly applied across groups, e.g. random selection of group-specific thresholds (Hardt, Price, and Srebro 2016; Jang, Shi, and Wang 2021), or randomly selecting agents from one group to receive positive classification with constant probability (Pleiss et al. 2017). Our postprocessing techniques, while concerned not exclusively with fairness, follow a similar inclusion/exclusion system.

More generally, randomized prediction methods are common in prior literature. In some cases, randomization is inherently desirable, for example, to explore or correct existing bias in domains such as hiring (Berger et al. 2020; Tassier and Menczer 2008; Hong and Page 2004) or lending (Karlan and Zinman 2010a,b). In other settings, the aim is to increase model robustness (Pinot et al. 2019; Salman et al. 2019), or to achieve better trade-offs between model performance and fairness, as is common in many group-fair classification approaches (Agarwal et al. 2018; Kearns et al. 2018; Pleiss et al. 2017).

Several recent papers look at the potential negative consequences of applying group fairness (Liu et al. 2018; Zhang et al. 2020; Corbett-Davies and Goel 2018; Kasy and Abebe 2021; Ben-Porat, Sandomirskiy, and Tennenholtz 2019). In particular (Liu et al. 2018; Ben-Porat, Sandomirskiy, and Tennenholtz 2019) demonstrate that specific types of group equity can be decreased by the use of fair algorithms. Others have merged notions of welfare and fairness (Hu and Chen 2020; Cousins 2021; Chen and Hooker 2021). Both the notion of popularity, as well as our proposed techniques for satisfying popularity and fairness, differ from these lines of work in that popularity casts welfare in terms of the fraction of a population which prefers a fair model compared with a fairness-agnostic model. While the idea of agent preference over models has received some recent attention (Ustun, Liu, and Parkes 2019) (which aims to classify a population using multiple models such that each agent prefers their assigned model over all others), popularity in the context of group fair learning has remained unexplored thus far.

2 Preliminaries

We begin by formalizing our models of conventional and fair learning, as well as our definition of popularity. Let $D = (\mathbf{X}, Y, G)$ be a dataset of n examples where the i^{th} example $(\mathbf{x}_i, y_i, g_i)$ consists of features $\mathbf{x}_i \in \mathcal{X} \subset \mathbb{R}^d$, binary labels $y_i \in \{0, 1\}$ and binary group membership $g_i \in \{0, 1\}$. We assume throughout that the positive label $y = 1$ corresponds to the *preferred outcome*, such as being selected to receive a valuable resource (e.g., college admission). Consider two learning schemes, say C and F, where

C is a *conventional* learning scheme, designed to minimize some fairness-agnostic objective, and F is a *fair* learning scheme, designed to achieve a desired level of fairness between groups. Then, C solves a problem of the form:

$$f_C \in \arg\min_{f \in \mathcal{H}_C} \mathcal{L}_C\big(f, \mathbf{X}, Y\big) \tag{1}$$

i.e., choosing an optimal model f_C from the hypothesis class $\mathcal{H}_C$ with respect to the loss $\mathcal{L}_C$. However, we do not *require* that f_C is the result of strict error minimization, only that it maps $\mathcal{X}$ to $\{0, 1\}$. In the context of conventional learning, the objective $\mathcal{L}_C$ and learning scheme C may have exogenous considerations aside from error minimization, such as robustness or interoperability.

We further assume that the learned classifier is of the type that produces a *score function* $h : \mathcal{X} \to [0, 1]$ which is used to induce the classification $f(\mathbf{X})$. Most classifiers used in practice yield such score functions (e.g., SVM, Logistic Regression, Neural Nets, Decision Trees, etc.). We study both deterministic and randomized classifiers in this framework. While deterministic predictions are most common, randomization can offer flexibility that can play a useful role both in achieving fairness (Dwork et al. 2012; Kearns et al. 2018) and robustness (Pinot et al. 2019; Li and Vorobeychik 2015; Salman et al. 2019; Vorobeychik and Li 2014). A deterministic classifier f can be thought of as a threshold on scores from h, i.e., $f(\mathbf{x}) = \mathbb{I}\big[h(\mathbf{x}) \geq \theta\big]$ for threshold θ. A randomized classifier f, in turn, can be viewed as a Bernoulli random variable with a mean given by h, i.e., $\mathbb{E}[f(\mathbf{x})] = h(\mathbf{x})$.

In addition to the classification setting above in which, in principle, anyone can be selected (i.e., assigned a positive outcome $y = 1$), we consider scarce resource allocation (henceforth simply *allocation*). In the allocation setting, unlike the classification setting, the model designer is limited in the number of positive predictions—that is, the number of individuals that can be selected. Specifically, the score function h is used to allocated $k < n$ homogeneous, indivisible, goods among a population of n agents. This follows a well-established paradigm of allocating scarce resources among individuals using a score function learned on a binary prediction task (Kube, Das, and Fowler 2019). Let $I_i(\mathbf{X}, h, k) \in \{0, 1\}$ indicate allocation of a resource to agent i when score function h is applied to a population $\mathbf{X}$ and there are k resources. Similar to the classification setting, the allocation function I can be deterministic or randomized. In the case of deterministic allocation, I_i is obtained directly from the set of scores $h(\mathbf{X})$, e.g., allocating resources to the k highest scoring individuals. In randomized allocation, I_i is a Bernoulli random variable, but unlike in the classification setting, I_i may have an arbitrary joint relationship with allocation decisions made for other agents, e.g., sampling without replacement weighted by $h(\mathbf{X})$.

Let $\mathcal{M}(f(\mathbf{X}), Y; g)$ be an efficacy metric computed with respect to group membership $g \in \{0, 1\}$ (for example, false positive rate (FPR) or error rate (ERR)). Define *group disparity* $\mathcal{U}(f, D) = |\mathcal{M}(f(\mathbf{X}), Y; 1) - \mathcal{M}(f(\mathbf{X}), Y; 0)|$, i.e., the difference in efficacy between two groups. Then the group-fair learning scheme F solves a problem of the form

$$f_F = \arg\min_{f \in \mathcal{H}_F} \mathcal{L}_F\big(f, \mathbf{X}, Y\big) \quad \text{s.t. } \mathcal{U}(f, D) \leq \beta. \tag{2}$$

i.e., f_F is an optimal *group-fair* model from hypothesis class $\mathcal{H}_F$, with fairness captured by the constraint that group disparity $\mathcal{U}$ is bounded by β. We refer to the fair learning scheme and model f_F as β-fair. Note that when resources are scarce, fairness is defined over allocation decisions $I_i(\mathbf{X}, h, k)$, not over scores h; an example of a fairness objective would be selection rate parity of $I_i(\mathbf{X}, h, k)$ between groups. Our analysis that follows applies to the broad class of *additive efficacy metrics* in both the classification and allocation settings.

Definition 2.1. *(Additive Efficacy Metric): An efficacy metric $\mathcal{M}$ is* additive *if for any population $(\mathbf{X}, Y, G)$,*

$$\mathcal{M}\big(f(\mathbf{X}), Y; g\big) = \sum_{y \in \{0,1\}} \sum_{\substack{i \in G_g: \\ y_i = y}} f(\mathbf{x}_i) c_{y,1}^{(g)} + (1 - f(\mathbf{x}_i)) c_{y,0}^{(g)}$$

for some $c_{y,0}^{(g)}, c_{y,1}^{(g)} \in [0, 1]$. In the case of scarce resources $f(\mathbf{x}_i)$ is interchangeable with $I_i(\mathbf{X}, h, k)$. In the case of randomized models, $f(\mathbf{x}_i)$ is replaced with $\mathbb{E}[f(\mathbf{x}_i)]$ or $\mathbb{E}[I_i(\mathbf{X}, h, k)]$.

In an additive efficacy metric, the coefficients $c_{y,0}^{(g)}, c_{y,1}^{(g)}$ give the respective "costs" of classifying an example from group G_g, with true label y, as negative or positive, respectively. Thus, unfairness $\mathcal{U}$ is given as the difference in the total efficacy cost between groups. Additive metrics are widely studied in the literature and include metrics such as error rate (ER), positive (or selection) rate (PR), false positive rate (FPR), and true positive rate (TPR). As an example, in the case of PR fairness $c_{y,1}^{(g)} = {}^{1}\!/_{|G_g|}$ and $c_{y,0}^{(g)} = 0$ for each $y, g \in \{0, 1\}$.

We consider the situation in which a conventional learning scheme C is initially in place, and a *principal* considers a switch from C to a group-fair scheme F, and wishes to ensure that F is *γ-popular* in the sense that it is preferred to C by at least a fraction γ of the target population. We formalize preference over learning schemes by assuming that an individual prefers schemes which yield higher expected outcomes for them, that is, they prefer being selected to not being selected, as in Hardt, Price, and Srebro (2016). Thus, an individual i with features $\mathbf{x}_i$ prefers F over C if

$$f_C(\mathbf{x}_i) \leq f_F(\mathbf{x}_i) \text{ or } I_{C,i}(\mathbf{X}, h, k) \leq I_{F,i}(\mathbf{X}, h, k) \tag{3}$$

when decisions are deterministic and

$$\begin{gathered} \mathbb{E}\big[f_C(\mathbf{x}_i)\big] \leq \mathbb{E}\big[f_F(\mathbf{x}_i)\big] \quad \text{or} \\ \mathbb{E}\big[I_{C,i}(\mathbf{X}, h, k)\big] \leq \mathbb{E}\big[I_{F,i}(\mathbf{X}, h, k)\big] \end{gathered} \tag{4}$$

when decisions are stochastic.

Note that our analysis is in the space of outcomes, rather than scores. Consequently, if decisions are deterministic, either in classification or allocation settings, agents only have a definitive preference over scores produced by h if this is consequential to outcomes (e.g., pushing them above or below θ). In the stochastic case, on the other hand, agents prefer the classifier or allocation scheme which yields the higher expected outcome (that is, higher probability of being selected).

Armed with this model of individual preference, we now define what it means for F to be popular.

Definition 2.2. ***(γ-popularity)****: A learning scheme F is said to be γ-popular with respect to a population $(\mathbf{X}, Y, G)$ and conventional scheme C, if Condition* (3) *(for deterministic models), or Condition* (4) *(for randomized models), holds for at least $\gamma|\mathbf{X}|$ individuals.*

Popularity thus captures the fraction γ of a population which is weakly better off (or, equivalently, *not* made worse) from the use of F over C. Similar to the concept of β-fairness, in which a model designer can specify the desired level of fairness β, the definition of popularity, as well as our postprocessing techniques described later, allow the model designer to *directly* specify, and control, the desired level of popularity. Note that we do not capture the *degree* to which individuals are made better or worse off as a result of switching from C to F, but only *whether* they are.

As mentioned earlier, our setting is one of a concrete choice by a principal between a particular conventional approach C and a particular group-fair approach F. This reflects a decision by the principal to switch from C—which is currently deployed—to F in order to reduce impact to a disadvantaged group (or groups). Of course, different pairs of C and F (e.g., using different loss functions, different learning algorithms, etc) would yield different judgments about popularity of F, which is, by construction, relative to C. Consequently, these will also yield different decisions about improving popularity of F based on algorithms we discuss below. Nevertheless, our framework generalizes immediately to a setting in which neither C nor F are fixed, and there is uncertain about either, or both. In such a case, we treat uncertainty about either C or F as a distribution over approaches and, consequently, over outcomes induced. This can then be immediately captured within our framework dealing with randomized schemes, and all definitions above, and technical results below, go through unchanged.

Our goal is to investigate the following three questions: 1) Are common group-fair learning techniques popular? 2) For a given γ and β, can we compute β-fair and γ-popular decisions in polynomial time? 3) What is the nature of the tradeoff between popularity, fairness, and accuracy?

3 Improving Popularity through Postprocessing

We consider two approaches to minimally postprocess a β-fair scheme f_F such that the resulting decisions also become γ-popular, for exogenously specified β and γ: 1) *direct outcome shift (DOS)* and 2) *k-quantile lottery shift (k-QLS)*. Postprocessing is performed in a transductive setting, in which the populations' features $(\mathbf{X}, G)$ (and possibly also labels Y) are known in advance. Throughout, we use f_P to refer to either approach we propose that combines both popularity and group fairness.

Direct Outcome Shift (DOS) DOS-based postprocessing arises from solving the problem of finding a minimal perturbation to the agents' outcomes that achieves both fairness and popularity, e.g. Program 5 for randomized classification. For a target population with feature vectors $\mathbf{X}$, we shift individuals' outcomes $f_F(\mathbf{X})$ or expected outcomes $\mathbb{E}[f_F(\mathbf{X})]$ by a *perturbation* vector $\mathbf{p}$. For deterministic decisions, $\mathbf{p} \in \{-1, 0, 1\}^n$, while for stochastic decisions $\mathbf{p} \in [-1, 1]^n$. The optimization goal in either case is to minimize $\|\mathbf{p}\|_q$ for some ℓ_q-norm ($q \in \{1, 2, \infty\}$) such that the final decisions, whether they involve predictions ($f_F(\mathbf{X})+\mathbf{p}$, or $\mathbb{E}[f_F(\mathbf{X})]+\mathbf{p}$) or allocations ($I(\mathbf{X}, h, k)+\mathbf{p}$, or $\mathbb{E}\big[I(\mathbf{X}, h, k)\big] + \mathbf{p}$) are both β-fair and γ-popular. Since DOS does not use knowledge of true labels Y, it can be applied directly at prediction time to a population of individuals. However, this also means that it can only be applied when the measure of fairness is independent of the true labels Y (for example, ensuring equality of positive rates).

k**-Quantile Lottery Shift (*****k*****-QLS)** Another option for creating popular and fair classifiers is to directly minimize a loss function regularized by the distance of the fair-and-popular classifier from the fair classifier (distance is measured on predictions at training time), e.g. Program 12 for randomized classifiers. k-QLS-based postprocessing achieves this goal by partitioning scores $h_F(\mathbf{X})$ for a population $\mathbf{X}$ into k bins (based on quantiles). The goal is then to compute probabilities $p_\ell^{(g)}$ for each bin ℓ and group g, which minimize empirical risk and change to each agent's outcome, while achieving γ-popularity and β-fairness. This is done at training time. Then at prediction time, we take all agents in group g with scores in bin ℓ and run a lottery, where each agent is classified as 1 with probability $p_\ell^{(g)}$, and 0 otherwise. Since k-QLS is applied on the training dataset, it also allows us to use fairness metrics that depend on labels Y; for this reason k-QLS is not used in allocation, where Y is typically unknown.

k-QLS is motivated by works such as (Hardt, Price, and Srebro 2016; Pleiss et al. 2017; Kamiran, Karim, and Zhang 2012; Canetti et al. 2019; Lohia et al. 2019) which aim to postprocess a conventional model to achieve β-fairness by running an "inclusion/exclusion" lottery on groups of agents. However, k-QLS differs from these approaches: shifting all outcomes of a group, even in a randomized manner, is too granular to achieve γ-popularity, and thus we shift outcomes within k quantiles. In Section C.3 of the Supplement we demonstrate the poor performance of group level shifts compared the higher precision shifts of both the quantile shifts of k-QLS and the individual shifts of DOS.

Remark 3.1. *Achieving γ-popularity and β-fairness may be infeasible in general. However, for common efficacy metrics (e.g., PR, FPR, and TPR), doing so is always possible. Both DOS and k-QLS have a feasible solution for any level of γ-popularity and β-fairness, for both randomized and deterministic models.*

3.1 Postprocessing for Deterministic Models

When the conventional model f_C, and β-fair model f_F are deterministic, the optimization problems defined for both the DOS approach and the k-QLS approach can be efficiently solved for any $\mathcal{U}$ defined by an additive efficacy metric $\mathcal{M}$. In both cases, since model decisions are binary, post processing amounts to finding some set of agents negatively classified by f_C, which minimally impact loss while not violating fairness, when positively classified.

Theorem 3.2 (Informal)**.** *When classifiers produce deterministic outcomes and $\mathcal{U}$ is defined by an additive fairness metric, the optimization problems for both DOS and k-QLS can be solved in polynomial time.*

We defer the formal statement of this claim, and a full discussion of deterministic postprocessing, to the Supplement.

3.2 DOS for Randomized Classification

Next we investigate popularity as it relates to randomized classifiers. Recall that in the case of randomized classifiers DOS aims to minimally shift the expected outcomes of f_F on a population $(\mathbf{X}, G)$, with unknown true labels Y, to produce the γ-popular β-fair model, which we denote by f_P, where $\mathbb{E}\big[f_P(\mathbf{x}_i)\big] = \mathbb{E}\big[f_F(\mathbf{x}_i)\big] + p_i$, and $0 \leq \mathbb{E}\big[f_P(\mathbf{x}_i)\big] \leq 1$. Thus, DOS aims to solve the following optimization problem:

$$\min_{\mathbf{p}\in[-1,1]^n} \|\mathbf{p}\|_q \tag{5}$$

$$\text{s.t. } \mathcal{U}\big(\mathbb{E}\big[f_F(\mathbf{X})\big] + \mathbf{p},\ G\big) \leq \beta \tag{6}$$

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\big[\mathbb{E}\big[f_C(\mathbf{x}_i)\big] \leq \mathbb{E}\big[f_F(\mathbf{x}_i)\big] + p_i\big] \geq \gamma \tag{7}$$

for $q \in \{1, 2, \infty\}$. A key challenge is that the popularity constraint (7) is discrete and non-convex, amounting to a combinatorial problem of identifying a subset of $\gamma|\mathbf{X}|$ individuals who prefer the f_P to its conventional counterpart f_C. Nevertheless, this problem can be solved in polynomial time.

Algorithm 1: **(Randomized DOS)** Postprocessing technique for converting a β-fair model f_F into a γ-popular β-fair model f_P.

```
Input: population: $(\mathbf{X}, Y, G)$, $\beta$-fair model: $f_F$, conventional model: $f_C$, popularity: $\gamma$
Result: weights $\mathbf{p}$ s.t. $f_P = f_F + \mathbf{p}$ is $\gamma$-popular and $\beta$-fair
$G_g := \{i : g_i = g\}$ s.t.
    $\mathbb{E}\big[f_C(\mathbf{x}_i)\big] - \mathbb{E}\big[f_F(\mathbf{x}_i)\big] \leq \mathbb{E}\big[f_C(\mathbf{x}_{i+1})\big] - \mathbb{E}\big[f_F(\mathbf{x}_{i+1})\big]$
$m := \lceil \gamma n \rceil$
for $i = 1$ to $m$ do
    $S_i = \big\{\mathbb{E}\big[f_C(\mathbf{x}_j)\big] \leq \mathbb{E}\big[f_F(\mathbf{x}_j)\big] + p_j : j \in G_1[: i]\big\}$
        $\cup \big\{\mathbb{E}\big[f_C(\mathbf{x}_j)\big] \leq \mathbb{E}\big[f_F(\mathbf{x}_j)\big] + p_j : j \in G_0[: m - i]\big\}$
    build Program 5 and replace Constraint 7 with $S_i$
    $\mathbf{p}_i$ = solution to the modified program
end for
return $\mathbf{p}^* = \arg\min_i \|\mathbf{p}_i\|$
```

Theorem 3.3. *Let f_C and f_F be respectively a conventional and β-fair randomized classifier. Let $\mathcal{U}$ be derived from an additive efficacy metric $\mathcal{M}$ which is independent of Y (e.g., PR). Then for $q \in \{1, 2, \infty\}$ Program 5 can be solved in time $\Theta(\gamma n T)$ (where $\Theta(T)$ is the time required to solve a linear program or semi-definite program, as appropriate) by Algorithm 1, which returns a γ-popular, β-fair model f_P.*

Proof Sketch. Recall that $\mathbb{E}[f(\mathbf{x})] = h(\mathbf{x})$, an agent i prefers f_P to f_C if $h_C(\mathbf{x}_i) \leq h_P(\mathbf{x}_i) = h_F(\mathbf{x}_i) + p_i$, and if this holds for at least $m = \gamma n$ agents then f_P is γ-popular. In the case of DOS postprocessing, if a *specific* set of m constraints is required to hold, rather than *any* m constraints, the problem is tractable as it is a linear program ($q = 1, \infty$) or semi-definite program ($q = 2$).

To order the set of possible constraints such that only a polynomial number must be examined, we make use of the following observations: for any two agents $i, j \in G_g$, 1.) since $\mathcal{U}$ is additive and independent of Y, unfairness is invariant under any change to p_i, p_j which preserves $p_i + p_j$, and 2.) if $h_C(\mathbf{x}_i) - h_F(\mathbf{x}_i) \geq h_C(\mathbf{x}_j) - h_F(\mathbf{x}_j)$ then $h_C(\mathbf{x}_i) \leq h_F(\mathbf{x}_i) + p_i$ iff $h_C(\mathbf{x}_j) \leq h_F(\mathbf{x}_j) + p_i$. Thus, for any solution $\mathbf{p}$ where $h_C(\mathbf{x}_i) \leq h_F(\mathbf{x}_i) + p_i$, but $h_C(\mathbf{x}_j) > h_F(\mathbf{x}_j) + p_j$, permuting p_i and p_j does not affect loss, fairness, or popularity, (when permutation is infeasible, shifting the maximum allowed weight from p_i to p_j is sufficient). Since the problem is invariant under such permutations, we need only consider imposing $h_C(\mathbf{x}_i) \leq h_F(\mathbf{x}_i) + p_i$ if $h_C(\mathbf{x}_j) \leq h_F(\mathbf{x}_j) + p_j$ is already imposed.

Thus, each G_g can be ordered such that for $i, j \in G_g$, if $j < i$ then $h_C(\mathbf{x}_j) - h_F(\mathbf{x}_j) \leq h_C(\mathbf{x}_i) - h_F(\mathbf{x}_i)$. Since the intragroup decisions are made trivial via this ordering, only the intergroup decisions remain. Since at least m popularity constraints need to hold, and there are m ways to select exactly m total constraints between the two groups while preserving the intragroup ordering, there are only m sets of constraints that need investigation. Each set corresponds to solving either a LP or SDP which takes time $\Theta(T)$ to solve. The specific running time of each program type is outlined in the Supplement. Thus the total running time of DOS is $\Theta(\gamma n T)$. □

3.3 DOS for Randomized Resource Allocation

Next we turn our attention to resource allocation, in which $k < n$ equally desirable resources are allocated to a population of size n. Recall that the randomized allocation scheme given by $I(\mathbf{X}, G)$ assigns resources to agents where $\mathbb{E}\big[I_i(\mathbf{X}, G)\big] \in [0, 1]$ gives the probability that agent i receives a resource with allocation performed over population $(\mathbf{X}, G)$. For notational convenience, we use $I(i) = \mathbb{E}\big[I_i(\mathbf{X}, G)\big]$ to represent the probability that agent i receives the resources and suppress the expectation and implicit dependence on the population $(\mathbf{X}, G)$.

Scarce resource allocation is particularly well suited for DOS as true labels (with respect to the allocation decision) are typically unknown. In this case, DOS postprocessing is given by,

$$\min_{\mathbf{p}\in[-1,1]^n} \|\mathbf{p}\|_q \tag{8}$$

$$\text{s.t. } \sum_{i=1}^{n} I_F(i) + p_i \leq k \tag{9}$$

$$\mathcal{U}\big(I_F + \mathbf{p},\ G\big) \leq \beta \tag{10}$$

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\big[I_C(i) \leq I_F(i) + p_i\big] \geq \gamma \tag{11}$$

We now show that DOS in resource allocation settings remains tractable.

Theorem 3.4. *Let I_C and I_F be a conventional and β-fair allocation scheme, respectively, and $\mathcal{U}$ be derived from an additive efficacy metric $\mathcal{M}$ which is independent of Y (e.g., PR). Then for $q \in \{1, 2, \infty\}$ Program 8 can be solved in time $\Theta(\gamma n T)$ by Algorithm 1 which returns a γ-popular, β-fair allocation if one exists.*

Proof Sketch. In the case of scarce resources, agents can again be ordered in an identical fashion to the classification setting (Theorem 3.3). Note that for any solution $\mathbf{p}$ and any $i, j \in G_g$, the resource constraint $\sum_{i=1}^n I_F(i) + p_i \leq k$ is invariant to any change in p_i, p_j, which preserves $p_i + p_j$. Thus a similar argument to Theorem 3.3, with a few caveats relating to infeasible solutions, holds. Specifically, this yields γn programs (either LPs or SDPs), each of which is solvable in time $\Theta(T)$. Thus DOS post processing for resource allocation can be computed in time $\Theta(\gamma n T)$. □

3.4 k-QLS for Randomized Classification

Finally, we explore k-QLS postprocessing for randomized classifiers. k-QLS creates k intervals by the quantiles of $h_F(\mathbf{X})$, where k is chosen by the model designer. Specifically, let ρ_ℓ be the maximum score associated with quantile ℓ of $h_F(\mathbf{X})$. Each interval is given as $I_\ell = [\rho_{\ell-1}, \rho_\ell]$, with the understanding that $\rho_0 = 0$ and $\rho_k = 1$. On each interval I_ℓ, and for each group g, a parameter $p_\ell^{(g)}$ is learned. At prediction time, $\mathbb{E}\big[f_P(\mathbf{x}_i)\big] = p_\ell^{(g_i)}$ for i s.t. $h_F(\mathbf{x}_i) \in I_\ell$, .

Finding the optimal lottery probabilities can formulated as the following optimization problem:

$$\min_{\mathbf{p} \in [0,1]^{2k}} \mathcal{L}\big(f_P, \mathbf{X}, Y\big) + \lambda \|f_F(\mathbf{X}) - f_P(\mathbf{X})\|_q^q \qquad (12)$$

$$\text{s.t. } \mathcal{U}\big(f_P, D\big) \leq \beta \qquad (13)$$

$$\frac{1}{n} \sum_{i=1}^n \mathbb{I}\big[f_C(\mathbf{x}_i) \leq f_P(\mathbf{x}_i, g_i)\big] \geq \gamma, \qquad (14)$$

where $\mathcal{L}$ is expected training error. As was the case for DOS postprocessing with randomized classifiers, the constraint that γ fraction of the population prefers f_P over f_C is discrete and non-convex. Indeed, unlike DOS, the k-QLS problem becomes strongly NP-hard.

Theorem 3.5. *Postprocessing to achieve γ-popularity and β-fairness with k-QLS (i.e., solving Program 12) is strongly NP-hard when models are randomized, and $\mathcal{U}$ is derived from an additive efficacy metric.*

We defer this proof to Section B.3 of the Supplement.

The intractability stems entirely from the model designer's ability to choose the number of quantiles k: if k is fixed, the problem can be solved in polynomial time as shown in the following theorem. In practice, we can fix k to be small, thus obtaining a tractable algorithm.

Theorem 3.6. *Let f_C and f_F be a conventional and a β-fair randomized classifier respectively. Let U be derived from an additive efficacy metric $\mathcal{M}$. Then for a fixed number of quantiles k, Program 12 for $q = \{1, 2, \infty\}$ can be solved in polynomial time, thus obtaining γ-popular β-fair decisions.*

Proof. As was the case for DOS applied to randomized classifiers, k-QLS applied to randomized classifiers is tractable if a specific set of $m = \gamma n$ agents is required to prefer f_P, rather than any m agents. When the number of intervals is constant it is straightforward to induce an ordering on agents which explores only a polynomial number of constraint sets. Specifically, let $G_{(g,\ell)} = \{i \in [n] : g_i = g \text{ and } h_F(\mathbf{x}_i) \in I_\ell\}$. Then agents in each G_g can be ordered by the magnitude of $p_\ell^{(g)}$ required such that they prefer f_P to f_C. Order G_g such that for $i, j \in G_g$ if $i < j$ then $h_C(\mathbf{x}_j) \leq h_C(\mathbf{x}_i)$, then if agent $i \in G_g$ prefers f_P to f_C, so does every $j \leq i$. There are $2k$ such sets, each containing at most n/k agents. Since the popularity over each G_g can be parameterized by the identity of the agent with the largest value of $h_C(\mathbf{x})$ who prefers f_P, there are no more than $(\gamma n)^k$ unique values under this parameterization, and thus no more than $(\gamma n)^k$ sets of constraints need be examined; each examination requires only polynomial time. □

4 Experiments

In this section we empirically investigate the relationship between popularity and fairness, and evaluate the efficacy of the proposed postprocessing algorithms. Each experiment is conducted on four data sets: 1) the **Recidivism** dataset, 2) the **Income** dataset, 3) the **Community Crime** dataset, and 4) the **Law School** dataset. In each dataset features can be continuous or categorical; each label is binary and defined such that 1 is always the more desirable outcome, e.g. in the Recidivism dataset $y = 1$ indicates *not* reoffending. A specific description of the label is given in the Supplement. Group membership is defined by race for Community Crime and Law School, and by gender for Recidivism and Income; either feature is assumed to be binary. All other sensitive features, such as age, are removed from the dataset. We consider three fair learning schemes: the **Reductions** algorithm (Agarwal et al. 2018), the **CalEqOdds** algorithm (Pleiss et al. 2017), the **KDE** algorithm (Cho, Hwang, and Suh 2020). Results for the latter two are provided in Section C of the supplement.

Popularity of Current Fair Learning Schemes: We begin by considering popularity of group-fair classifiers. The fractions of the overall population, and subgroup population, which prefer the fair classifier are shown in Figure 1, where fairness is achieved using the **Reductions** method.

Not surprisingly, we see that in all instances the disadvantaged group G_0 prefers f_F at far higher rates than G_1. With the exception of the **CalEqOdds** algorithm (which achieves fairness via group specific score shifts, resulting in far stronger group-level preference over classifiers), results for other methods are similar; these are provided in the supplement. Overall, randomized fair classifiers frequently have popularity of less than 50% . On the other hand, fair deterministic classifiers are relatively popular in most cases.

In either case, however, postprocessing can be used to further boost popularity of group-fair methods.

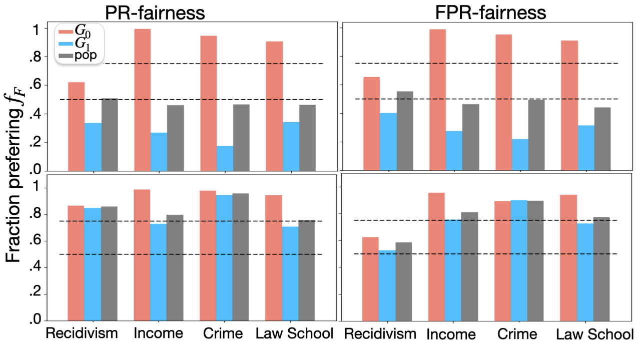

Figure 1: Fraction of each population or group preferring f_F over f_C for randomized classifiers (top) and deterministic classifiers (bottom), when f_F is learned via the Reductions algorithm.

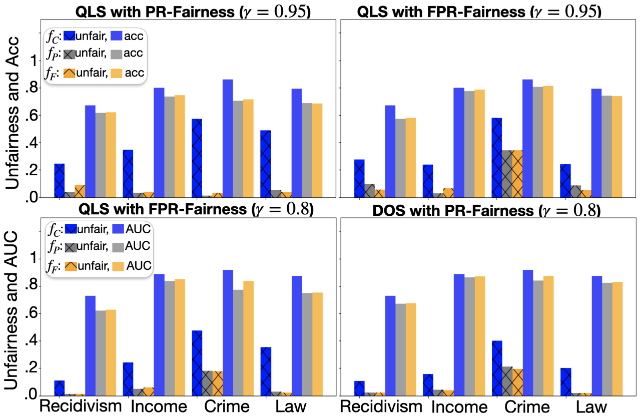

Figure 2: Model performance and unfairness on test data (3-fold average) for deterministic models with $\gamma = 0.95$ (top) and randomized models with $\gamma = 0.8$ (bottom). The conventional classifier f_C, fair classifier f_F (learned via reductions), and the fair popular classifier f_P (learned via our postprocessing technique), each using Logistic Regression.

Postprocessing for Fairness and Popularity: Next we examine the efficacy of our proposed postprocessing techniques DOS and k-QLS (k=10). When classifiers are deterministic, performance is measured using balanced accuracy (balanced w.r.t. Y). When classifiers are randomized, performance is measured using ROC-AUC, calculated over model scores (i.e., expected outcomes).

Remark 4.1. *Both k-QLS and DOS may require solving a large number of LPs or SDPs, which may be expensive. However, both methods can be efficiently implemented in practice by either solving the programs in parallel, trimming down the number of programs with heuristics, or replacing all programs with a single integer program. The latter being the most efficient, typically finishing in under 60 seconds. Further details on these methods, and exact running times, are provided in Section C.3 of the Supplement.*

Figure 2 shows that both k-QLS and DOS are able to achieve high levels of γ-popularity and β-fairness with little degradation in performance. In particular, deterministic classifiers (due to their higher natural popularity) are able to achieve greater levels of popularity compared to randomized models, with similar levels of degradation to performance. We observe similar results for other combinations of dataset, efficacy metric, and classifier type (Section C of the supplement).

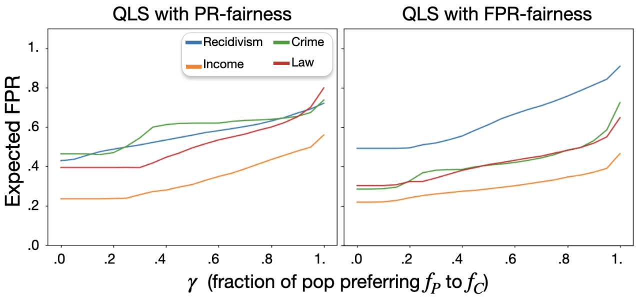

Figure 3: Expected False Positive Rate (FPR) of k-QLS, on randomized classifiers,as a function of γ.

Finally, we consider the extent to which popularity may skew model efficacy. In particular, as the popularity coefficient γ increases, a larger fraction of the population is guaranteed to have scores from f_P, which are at least as large as those from f_C. Since popularity constraints ensure that agents scores do not decrease, achieving higher levels of popularity (i.e., higher γ) also incentivize the resulting f_P to maintain false positive errors made by f_C. Thus one would expect FPR to increase with γ. This phenomenon is shown in Figure 3, which demonstrates that as γ increases, so does expected FPR. Although the expected FPRs vary between datasets and fairness definitions, the rate of increase is relatively similar across instances.

In Section C of the supplement, we further explore the tradeoffs between error, fairness, and popularity via the Pareto frontiers of these values. Similar to the classic results involving fairness and accuracy, we find that there is a fundamental tradeoff between model accuracy and popularity.

5 Conclusion

The deployment of group-fair classifiers, in place of conventional classifiers, may result large fractions of a population perceiving that they are made worse off by the change. We introduce the notion of popularity, which captures the fraction of agents preferring one classifier over another, and propose two postprocessing techniques (DOS and k-QLS) for achieving popularity while retaining good fairness properties. Both techniques provide efficient solutions for both deterministic and randomized classifiers. We note that while in practice postprocessing can achieve popularity and fairness with minimal degradation to model performance, requiring higher levels of popularity can actually entrench any false positive errors made by the conventional model. Consequently, application of the proposed techniques need to carefully analyze the tradeoffs not merely between popularity, group fairness, and overall accuracy, but also with specific measures of error, particularly the false positive rate.

Acknowledgments

This research was partially supported by the National Science Foundation (IIS-1905558, IIS-1903207, IIS-1939677, IIS-2127752, IIS-2127754, IIS-1908287 and IIS-1942336) and Amazon.

References

Agarwal, A.; Beygelzimer, A.; Dudík, M.; Langford, J.; and Wallach, H. 2018. A reductions approach to fair classification. In *International Conference on Machine Learning*, 60–69. PMLR.

Angwin, J.; Larson, J.; Mattu, S.; and Kirchner, L. 2016. Machine bias. In *Ethics of Data and Analytics*, 254–264. Auerbach Publications.

Barocas, S.; Hardt, M.; and Narayanan, A. 2017. Fairness in machine learning. *Nips tutorial*, 1: 2.

Ben-Porat, O.; Sandomirskiy, F.; and Tennenholtz, M. 2019. Protecting the protected group: Circumventing harmful fairness. *arXiv preprint arXiv:1905.10546*.

Berger, J.; Osterloh, M.; Rost, K.; and Ehrmann, T. 2020. How to prevent leadership hubris? Comparing competitive selections, lotteries, and their combination. *The Leadership Quarterly*, 31(5): 101388.

Canetti, R.; Cohen, A.; Dikkala, N.; Ramnarayan, G.; Scheffler, S.; and Smith, A. 2019. From soft classifiers to hard decisions: How fair can we be? In *Proceedings of the conference on fairness, accountability, and transparency*, 309–318.

Chen, V.; and Hooker, J. 2021. A Guide to Formulating Equity and Fairness in an Optimization Model.

Cho, J.; Hwang, G.; and Suh, C. 2020. A fair classifier using kernel density estimation. *Advances in Neural Information Processing Systems*, 33: 15088–15099.

Chouldechova, A.; and Roth, A. 2018. The frontiers of fairness in machine learning. *arXiv preprint arXiv:1810.08810*.

Citron, D. K.; and Pasquale, F. 2014. The scored society: Due process for automated predictions. *Wash. L. Rev.*, 89: 1.

Corbett-Davies, S.; and Goel, S. 2018. The measure and mismeasure of fairness: A critical review of fair machine learning. *arXiv preprint arXiv:1808.00023*.

Cousins, C. 2021. An axiomatic theory of provably-fair welfare-centric machine learning. *Advances in Neural Information Processing Systems*, 34.

Dastin, J. 2018. Amazon scraps secret AI recruiting tool that showed bias against women. In *Ethics of Data and Analytics*, 296–299. Auerbach Publications.

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

Feldman, M.; Friedler, S. A.; Moeller, J.; Scheidegger, C.; and Venkatasubramanian, S. 2015. Certifying and removing disparate impact. In *proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 259–268.

Hardt, M.; Price, E.; and Srebro, N. 2016. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29.

Hong, L.; and Page, S. E. 2004. Groups of diverse problem solvers can outperform groups of high-ability problem solvers. *Proceedings of the National Academy of Sciences*, 101(46): 16385–16389.

Hu, L.; and Chen, Y. 2020. Fair classification and social welfare. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 535–545.

Jang, T.; Shi, P.; and Wang, X. 2021. Group-Aware Threshold Adaptation for Fair Classification. *arXiv preprint arXiv:2111.04271.*

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

Karlan, D.; and Zinman, J. 2010a. Expanding credit access: Using randomized supply decisions to estimate the impacts. *The Review of Financial Studies*, 23(1): 433–464.

Karlan, D.; and Zinman, J. 2010b. Expanding credit access: Using randomized supply decisions to estimate the impacts. *The Review of Financial Studies*, 23(1): 433–464.

Kasy, M.; and Abebe, R. 2021. Fairness, equality, and power in algorithmic decision-making. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 576–586.

Kearns, M.; Neel, S.; Roth, A.; and Wu, Z. S. 2018. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International Conference on Machine Learning*, 2564–2572. PMLR.

Koenecke, A.; Nam, A.; Lake, E.; Nudell, J.; Quartey, M.; Mengesha, Z.; Toups, C.; Rickford, J. R.; Jurafsky, D.; and Goel, S. 2020. Racial disparities in automated speech recognition. *Proceedings of the National Academy of Sciences*, 117(14): 7684–7689.

Kube, A.; Das, S.; and Fowler, P. J. 2019. Allocating interventions based on predicted outcomes: A case study on homelessness services. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 622–629.

Kusner, M. J.; Loftus, J.; Russell, C.; and Silva, R. 2017. Counterfactual Fairness. In Guyon, I.; Luxburg, U. V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Lee, N. T. 2018. Detecting racial bias in algorithms and machine learning. *Journal of Information, Communication and Ethics in Society*.

Li, B.; and Vorobeychik, Y. 2015. Scalable optimization of randomized operational decisions in adversarial classification settings. In *Artificial Intelligence and Statistics*, 599–607.

Liu, L. T.; Dean, S.; Rolf, E.; Simchowitz, M.; and Hardt, M. 2018. Delayed impact of fair machine learning. In *International Conference on Machine Learning*, 3150–3158. PMLR.

Lohia, P. K.; Ramamurthy, K. N.; Bhide, M.; Saha, D.; Varshney, K. R.; and Puri, R. 2019. Bias mitigation post-processing for individual and group fairness. In *Icassp 2019-2019 ieee international conference on acoustics, speech and signal processing (icassp)*, 2847–2851. IEEE.

Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K.; and Galstyan, A. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)*, 54(6): 1–35.

Pinot, R.; Meunier, L.; Araujo, A.; Kashima, H.; Yger, F.; Gouy-Pailler, C.; and Atif, J. 2019. Theoretical evidence for adversarial robustness through randomization. In *Neural Information Processing Systems*.

Pleiss, G.; Raghavan, M.; Wu, F.; Kleinberg, J.; and Weinberger, K. Q. 2017. On fairness and calibration. *Advances in neural information processing systems*, 30.

Salman, H.; Li, J.; Razenshteyn, I.; Zhang, P.; Zhang, H.; Bubeck, S.; and Yang, G. 2019. Provably robust deep learning via adversarially trained smoothed classifiers. In *Neural Information Processing Systems*, volume 32.

Tassier, T.; and Menczer, F. 2008. Social network structure, segregation, and equality in a labor market with referral hiring. *Journal of Economic Behavior & Organization*, 66(3-4): 514–528.

Ustun, B.; Liu, Y.; and Parkes, D. 2019. Fairness without harm: Decoupled classifiers with preference guarantees. In *International Conference on Machine Learning*, 6373–6382. PMLR.

Vorobeychik, Y.; and Li, B. 2014. Optimal randomized classification in adversarial settings. In *International Conference on Autonomous Agents and Multiagent Systems*, 485–492.

Zhang, X.; Tu, R.; Liu, Y.; Liu, M.; Kjellstrom, H.; Zhang, K.; and Zhang, C. 2020. How do fair decisions fare in long-term qualification? *Advances in Neural Information Processing Systems*, 33: 18457–18469.

Supplement

A Postprocessing for Deterministic Classifiers

A.1 DOS for Deterministic Classification

Recall that DOS post processing, given conventional model f_C, β-fair model f_F and population $(\mathbf{X}, G)$, aims to select a vector $\mathbf{p} \in \{-1, 0, 1\}^n$ such that the classifier $f_P(\mathbf{x}_i) = f_F(\mathbf{x}_i) + p_i$ is both γ-popular and β-fair, while minimizing $\|\mathbf{p}\|_q$. For deterministic DOS we study $q = 1$ as each $0 \le q < \infty$ are equivalent, namely in that each yields the Hamming distance between f_P and f_F, and $q = \infty$ is simply an indicator of $f_P \neq f_F$. For deterministic classifiers DOS can be formulated as

$$\min_{\mathbf{p} \in \{-1,0,1\}^n} \|\mathbf{p}\|_q \tag{15}$$

$$\text{s.t. } \mathcal{U}\big(f_F + \mathbf{p},\ D\big) \le \beta \tag{16}$$

$$\frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\big[f_C(\mathbf{x}_i) \le f_F(\mathbf{x}_i) + p_i\big] \ge \gamma \tag{17}$$

$$0 \le f_F(\mathbf{x}_i) + p_i \le 1 \tag{18}$$

Objective 15 can be solved by Algorithm 3. Since decisions are binary, DOS is effectively selecting some minimum number of decisions from $f_F(\mathbf{X})$ to flip. In the deterministic case, DOS postprocessing is not technically difficult, but is illustrative of some key ideas used in other, more complex, cases. Specifically, the selection of which agents to flip decisions for is made straightforward by two observations. First, popularity increases only when flipping the decisions of agents with $f_F(\mathbf{x}) = 0$ and $f_C(\mathbf{x}) = 1$. Second, agents within a group are exchangeable with respect to fairness in the sense that for $i, j \in G_g$, either setting of $f_P(\mathbf{x}_i) = 1 - f_P(\mathbf{x}_j)$ results in identical fairness since $\mathcal{U}$ is derived from an additive metric. Combining these observations implies that no optimal solution can have $p_i = 1$ and $p_j = -1$ for $i, j \in G_g$. Moreover, an optimal solution will only choose to flip agents to negative classification, i.e. $p_i = -1$, if doing so is required to rebalance fairness. Thus, with respect to flipping decisions, agents are equivalent up to group membership, $\mathbb{I}[f_F(\mathbf{x}) < f_C(\mathbf{x})]$, and $\mathbb{I}[f_F(\mathbf{x}) = 1]$; implying DOS reduces to deciding whether to increase or decrease positive classifications on each G_g.

Using these facts, it is straightforward to alternate between groups and either positively classify an agent from $S_1 = \{i : f_F(\mathbf{x}_i) < f_C(\mathbf{x}_i)\}$, or negatively classify an agent from $S_2 = \{i : f_F(\mathbf{x}_i) = 1\}$. When positively classifying two agents from different groups has a cancellation-like affect on unfairness (e.g. PR), DOS will never negatively classify an agent with $f_F(\mathbf{x}) = 1$. In such cases f_P is a Pareto-impairment from f_F with respect to agent preference.

Theorem A.1. *Let f_C and f_F be a conventional and β-fair classifier respectively, both of which are deterministic. Let U be derived from an additive efficacy metric $\mathcal{M}$ which is independent of Y (e.g., PR). Then DOS, given by Program 15, returns a γ-popular β-fair model f_P and can be solved by Algorithm 3 in time $\Theta(n)$.*

Proof. Let $m = \lceil \gamma n \rceil$, i.e., m is the number of popularity constraints that must be satisfied. Each such constraint involves a single variable $p_i \in \{-1, 0, 1\}$ and thus is independent from any other popularity constraint. Moreover since unfairness $\mathcal{U}$ is additive, it can be expressed as

$$\begin{aligned}
&\mathcal{U}\big(f_F(\mathbf{X}) + \mathbf{p}, G\big) = \big|\mathcal{M}\big(f_F(\mathbf{X}) + \mathbf{p} : g = 1\big) - \mathcal{M}\big(f_F(\mathbf{X}) + \mathbf{p} : g = 0\big)\big| \\
&= \Big| \sum_{i \in G_1} c_1^{(1)}\big(f_F(\mathbf{x}_i) + p_i\big) + c_0^{(1)}\big(1 - \big(f_F(\mathbf{x}_i) + p_i\big)\big) - \sum_{j \in G_0} c_1^{(0)}\big(f_F(\mathbf{x}_j) + p_j\big) + c_0^{(0)}\big(1 - \big(f_F(\mathbf{x}_j) + p_j\big)\big) \Big| \\
&= \left| \left(\sum_{i \in G_1} (c_1^{(1)} - c_0^{(1)}) p_i - \sum_{j \in G_0} (c_1^{(0)} - c_0^{(0)}) p_j \right) + \left(\sum_{i \in G_1} c_1^{(1)} f_F(\mathbf{x}_i) + c_0^{(1)}\big(1 - f(\mathbf{x}_i)\big) - \sum_{j \in G_0} c_1^{(0)} f_F(\mathbf{x}_j) - c_0^{(0)}\big(1 - f_F(\mathbf{x}_j)\big) \right) \right|
\end{aligned}$$

for scalars $c_1^{(g)}, c_0^{(g)}$ which give the respective cost of positively or negatively classifying an agent from group G_g. Note that

$$u := \sum_{i \in G_1} c_1^{(1)} f_F(\mathbf{x}_i) + c_0^{(1)}\big(1 - f(\mathbf{x}_i)\big) - \sum_{j \in G_0} c_1^{(0)} f_F(\mathbf{x}_j) - c_0^{(0)}\big(1 - f_F(\mathbf{x}_j)\big)$$

is constant for fixed f_F and $(\mathbf{X}, Y, G)$. Then the fairness constraint can be expressed as

$$\begin{aligned}
&\mathcal{U}\big(f_F(\mathbf{X}) + \mathbf{p}, G\big) \le \beta \\
\iff\ & -\beta - u \le \sum_{i \in G_1} (c_1^{(1)} - c_0^{(1)}) p_i - \sum_{j \in G_0} (c_1^{(0)} - c_0^{(0)}) p_j \le \beta - u
\end{aligned} \tag{19}$$

Due to the additive nature of this fairness term, each member of group g is exchangeable, meaning that for any two agents $i_1, i_2 \in G_g$, fairness is invariant under any alteration to p_{i_1}, p_{i_2} which preserves the value of $p_{i_1} + p_{i_2}$. More specifically, for any $i_1, i_2 \in G_g$ and any feasible solution $\mathbf{p}$ with $p_{i_1} = -1$ and $p_{i_2} = 1$, let $\mathbf{p}'$ be defined by $p'_k = p_k$ for all $k \neq i_1, i_2$ and

$p'_{i_1} = p'_{i_2} = 0$. Then $\mathbf{p}'$ is both a feasible solution and has $\|\mathbf{p}'\| \leq \|\mathbf{p}\|$. The latter part of which is straightforward; to see the former we need only consider the popularity constraints since fairness is satisfied by the feasibility of $\mathbf{p}$ and $p_{i_1} + p_{i_2} = p'_{i_1} + p'_{i_2} = 0$. Although it may be the case that $f_C(\mathbf{x}_i) > f_F(\mathbf{x}_i) + p'_i = f_F(\mathbf{x}_i) + p_i - 1$, i.e., agent i no longer prefers f_F, it must be the case that $f_C(\mathbf{x}_j) \leq 1 \leq f_F(\mathbf{x}_i) + p_j + 1 = f_F(\mathbf{x}_i) + p'_j$, i.e., agent j prefers f_F.

Since agents from the same group are exchangeable and no optimal solution has both $p_{i_1} = -1$ and $p_{i_2} = 1$ for $i_1, i_2 \in G_g$, the optimal score shift $\mathbf{p}$ can be found by alternating between groups and greedily assigning either $p_i = 1$ or $p_i = -1$, as outlined by Algorithm 3. To see the optimality of this greedy selection procedure, let $U(f_F(\mathbf{X}) + \mathbf{p}, G) = \mathcal{M}(f_F(\mathbf{X}) + \mathbf{p} : g = 1) - \mathcal{M}(f_F(\mathbf{X}) + \mathbf{p} : g = 1)$, i.e. the function U is equivalent to $\mathcal{U}$ without absolute value. With respect to greedy selection, only two cases need be considered: 1.) $\text{sign}(c_1^{(1)} - c_0^{(1)}) = \text{sign}(c_1^{(0)} - c_0^{(0)})$ and 2.) $\text{sign}(c_1^{(1)} - c_0^{(1)}) = -\text{sign}(c_1^{(0)} - c_0^{(0)})$.

In case (1) if choosing to positively classify an agent from group G_1 increases (decreases) the value of $U(f_F(\mathbf{X}) + \mathbf{p}, G)$ then positively classifying an agent from group G_0 decreases (increases) the value of $U(f_F(\mathbf{X}) + \mathbf{p}, G)$. Thus if increasing the number of positive classifications on G_0, or on G_1, violates unfairness, the only way to resatisfy fairness is to increase the number of positive classifications on the other group. In case (2) if choosing to positively classify an agent from group G_1 increases (decreases) the value of $U(f_F(\mathbf{X}) + \mathbf{p}, G)$ then positively classifying an agent from group G_0 also increases (decreases) the value of $U(f_F(\mathbf{X}) + \mathbf{p}, G)$. Thus if increasing the number of positive classifications on G_0, or on G_1, violates unfairness, the only way to resatisfy fairness is to decrease the number of positive classifications on the other group.

The selection process examines at most n agents, and each decision on an agent takes constant time. Thus DOS can be solved in time $\Theta(n)$ for deterministic classifiers. □

A.2 k-QLS for Deterministic Classification

Recall that k-QLS is a postprocessing technique which, given a conventional model f_C, a β-fair model f_F, and a training set $D = (\mathbf{X}, Y, G)$, postprocesses the predictions of f_F such that they are γ-popular and β-fair. This is achieved by running a lottery on k quantiles defined by the scores of f_F, namely $h_F(\mathbf{X})$, the resulting model after postprocessing is refereed to as f_P. Specifically, the scores $h_F(\mathbf{X})$ are partitioned into k intervals in the following manner: let ρ_ℓ be the maximum score associated with quantile ℓ of $h_F(\mathbf{X})$, and let $I_\ell = [\rho_{\ell-1}, \rho_\ell]$ with the understanding that $\rho_{-1} = 0$ and $\rho_k = 1$. The resulting classifier f_P makes predictions $f_P(\mathbf{x}) = p_\ell^{(g')}$ for $h_F(\mathbf{x}) \in I_\ell$ with $g = g'$. In the case of deterministic models, $p_\ell^{(g')} \in \{0, 1\}$. The optimal β-fair γ-popular model can be found by solving:

$$\min_{\mathbf{p}^{(0)}, \mathbf{p}^{(1)} \in \{0,1\}^{2k}} \quad \mathcal{L}\big(f_P, D\big) + \lambda \|f_P(\mathbf{X}) - f_F(\mathbf{X})\|_q^q \tag{20}$$

$$\text{s.t.} \quad \mathcal{U}\big(f_P, D\big) \leq \beta \tag{21}$$

$$\frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\big[f_C(\mathbf{x}_i) \leq f_P(\mathbf{x}_i, g_i)\big] \geq \gamma \tag{22}$$

where $\mathcal{L}$ is balanced accuracy. Unlike DOS, k-QLS does not admit a straightforward solution, but is still polynomial time solvable. The key difference between these two techniques is that k-QLS makes decisions over sets of agents, rather than individual agents, and each interval may contain any number of agents with any combination of true labels and predicted outcomes under both f_C and f_F. Thus much of the symmetry from the DOS case is lost, however enough symmetry remains that a dynamic programming solution can produce the optimal f_P in polynomial time.

Theorem A.2. *Let f_C and f_F be a conventional and β-fair classifier respectively, both of which are deterministic. Let U be derived from an additive efficacy metric $\mathcal{M}$ (e.g. FPR). Then k-QLS, given by Program 20, returns a γ-popular and β-fair model f_P time $\Theta\big(\gamma k n^6\big)$ via dynamic programming. Moreover, when $\mathcal{M}$ is given by FPR, TPR, PR, or ERROR, f_P can be found in time $\Theta\big(\gamma k n^4\big)$.*

Before proving this theorem we first mention that while k-QLS admits a polynomial time solution, k-QLS can also be transformed into a MILP and solvers such as CPLEX may be more efficient in practice since k, the number of variables, will typically be constant (e.g., breaking scores into 10 intervals) and the program contains only two constraints (one for popularity and one for fairness).

Proof of Theorem A.2. When post processing with k-QLS the model designer creates k intervals based on the quantiles of $h_F(\mathbf{x})$ and aims to shift the scores of agents in each interval such that γ-popularity and β-fairness are achieved. Let ρ_ℓ be the maximum score associated with quantile ℓ of $h_F(\mathbf{X})$, and let $I_\ell = [\rho_{\ell-1}, \rho_\ell]$ with the understanding that $\rho_{-1} = 0$.

Thus k-QLS aims to find binary vectors $\mathbf{p}^{(g)} \in \{0, 1\}^k$ for each group G_g, such that the model $f_P(\mathbf{x}) = p_\ell^{(g)}$ for $h_F(\mathbf{x}) \in I_\ell$, in group g is γ-popular and β-fair. Since unfairness $\mathcal{U}$ is given in terms of an additive efficacy metric $\mathcal{M}$, the unfairness of

Algorithm 2: **(Deterministic k-QLS)** Postprocessing technique, learned at training time and later applied at prediction time, for converting a deterministic β-fair model f_F into γ-popular β-fair model f_P.

```
input: population: $(\mathbf{X}, G)$, $\beta$-fair model: $f_F$, score function of $f_F$: $h_F$,  conventional model: $f_C$, popularity: $\gamma$, quantiles $k$
result: Weight $\mathbf{p} \in \{0,1\}^{2k}$ of the $\gamma$-popular $\beta$-fair $f_P$
$\rho_\ell$ := maximum score in quantile $\ell$ of $h_F(\mathbf{X})$   $\forall \ell \in [k]$ /* partition scores $h_F(\mathbf{X})$ in $k$ intervals based on quantile */
$I_\ell = [\rho_{\ell-1}, \rho_\ell]$   $\forall \ell \in [k]$
$\mathbf{p}^{(0)}, \mathbf{p}^{(1)} := \mathbf{0}$
/* parameters indicating the effects of setting $p_\ell^{(g)} := 1$ */
$N_\ell^{(g)}$ := # of agents in $G_g$ with $h_f(\mathbf{x}) \in I_\ell$ and $f_C(\mathbf{x}) = 1$
$C_\ell^{(g)}$ := increase to unfairness (without absolute value)
$L_\ell^{(g)}$ := increase to loss
/* partition each group and interval according according to effect on fairness */
$S_+ := \{(g, \ell) : 0 \leq C_\ell^{(g)}\}$
$S_- := \{(g, \ell) : C_\ell^{(g)} < 0\}$
/* loss independent on each $I_\ell$, and thus on $S_+$ and $S_-$ */
build a knapsack-like problem over each $S$ using weights $C_\ell^{(g)}$, $N_\ell^{(g)}$, and values $L_\ell^{(g)}$
/* $p_\ell^{(g)}$ corresponds to selecting item $(g, \ell)$ polynomial number of possibilities for each */
$\mathbf{m}_+, \mathbf{m}_-$ := all possible # of agents preferring $f_F$ corresponding to solution from $S_+, S_-$
$\mathbf{u}_+, \mathbf{u}_-$ := all possible values of unfairness corresponding to solution from $S_+, S_-$
for each $m_- \in \mathbf{m}_-$ and each $u_- \in \mathbf{u}_-$ do
    dynamically compute optimal solution from $S_-$ using exactly $m_-$ agents and $u_-$ unfairness
    for each $m_+ \in \mathbf{m}_+$ and $u_+ \in \mathbf{u}_+$ do
        dynamically compute optimal solution from $S_+$ using exactly $m_+$ agents and $u_+$ unfairness.
        if solution from $S_-$ and $S_+$ is feasible then
            save the combined solution
        end if
    end for
end forreturn $\mathbf{p}$ corresponding to solution with the lowest loss
```

f_P over population $D = (\mathbf{X}, Y, G)$ can be expressed as

$$
\begin{aligned}
&\mathcal{U}\big(f_P, D\big) = \big|\mathcal{M}\big(f_P(\mathbf{X}, g), Y : g = 1\big) - \mathcal{M}\big(f_P(\mathbf{X}, g), Y : g = 0\big)\big| \\
=&\Bigg|\sum_{\ell=1}^{k} \sum_{y\in\{0,1\}} \Bigg(\sum_{\substack{i\in G_1 \\ h_F(\mathbf{x}_i)\in I_\ell}} c_{y,1}^{(1)} p_\ell^{(1)} \big(1 - |y - y_i|\big) + c_{y,0}^{(1)} \big(1 - p_\ell^{(1)}\big)\big(1 - |y - y_i|\big) \\
&\quad - \sum_{\substack{j\in G_0 \\ h_F(\mathbf{x}_i)\in I_\ell}} c_{y,1}^{(0)} p_\ell^{(0)} \big(1 - |y - y_i|\big) + c_{y,0}^{(0)} \big(1 - p_\ell^{(0)}\big)\big(1 - |y - y_i|\big)\Bigg)\Bigg|
\end{aligned}
$$

$$
\begin{aligned}
=&\Bigg|\sum_{\ell=1}^{k} \Bigg(p_\ell^{(1)} \sum_{y\in\{0,1\}} (c_{y,1}^{(1)} - c_{y,0}^{(1)}) \sum_{\substack{i\in G_1 \\ h_F(\mathbf{x}_i)\in I_\ell}} \big(1 - |y - y_i|\big) \\
&- p_\ell^{(0)} \sum_{y\in\{0,1\}} (c_{y,1}^{(0)} - c_{y,0}^{(0)}) \sum_{\substack{i\in G_0 \\ h_F(\mathbf{x}_i)\in I_\ell}} \big(1 - |y - y_i|\big) \\
&+ \sum_{y\in\{0,1\}} \sum_{\substack{i\in G_1: \\ h_F(\mathbf{x}_i)\in I_\ell}} c_{y,0}^{(1)} \big(1 - |y - y_i|\big) - \sum_{y\in\{0,1\}} \sum_{\substack{i\in G_0: \\ h_F(\mathbf{x}_i)\in I_\ell}} c_{y,0}^{(0)} \big(1 - |y - y_i|\big)\Bigg)\Bigg|
\end{aligned}
$$

for scalar costs $c_{y,1}^{(g)}, c_{y,0}^{(g)}$. Note that

$$
u := \sum_{\ell=1}^{k} \sum_{y\in\{0,1\}} \sum_{\substack{i\in G_1: \\ h_F(\mathbf{x}_i)\in I_\ell}} c_{y,0}^{(1)} \big(1 - |y - y_i|\big) - \sum_{y\in\{0,1\}} \sum_{\substack{i\in G_0: \\ h_F(\mathbf{x}_i)\in I_\ell}} c_{y,0}^{(0)} \big(1 - |y - y_i|\big)
$$

and each

$$
C_\ell^{(g)} := (1 - 2g) \sum_{y\in\{0,1\}} (c_{y,1}^{(g)} - c_{y,0}^{(g)}) \sum_{\substack{i\in G_g \\ h_F(\mathbf{x}_i)\in I_\ell}} \big(1 - |y - y_i|\big)
$$

are constants. Thus the fairness constraint on f_P can be expressed as

$$
\begin{aligned}
&\mathcal{U}\big(f_P, D\big) \leq \beta \\
\iff\ & -\beta - u \leq \sum_{\ell=1}^{k} \big(2(1) - 1\big) p_\ell^{(1)} C_\ell^{(1)} - \big(2(0) - 1\big) p_\ell^{(0)} C_\ell^{(0)} \leq \beta - u && (23) \\
\iff\ & -\beta - u \leq \sum_{\ell=1}^{k} p_\ell^{(1)} C_\ell^{(1)} + p_\ell^{(0)} C_\ell^{(0)} \leq \beta - u && (24)
\end{aligned}
$$

Thus, unfairness of f_P is given by a linear constraint on the vectors $\mathbf{p}^{(1)}$ and $\mathbf{p}^{(0)}$.

Similar to the unfairness term, the optimization objective

$$
\begin{aligned}
&\mathcal{L}(f_P, \mathbf{X}, Y, G) + \lambda \|f_F(\mathbf{X}) - f_P(\mathbf{X})\|_q^q \\
=&\sum_{\ell}^{k} \sum_{g\in\{0,1\}} \Bigg(\sum_{\substack{i\in G_g \\ h_F(\mathbf{x}_i)\in I_\ell}} (1 - y_i) p_\ell^{(g)} + y_i (1 - p_\ell^{(g)}) \Bigg) + \sum_{g\in\{0,1\}} \sum_{\substack{i\in G_g \\ h_F(\mathbf{x}_i)\in I_\ell}} \lambda \big|f_F(\mathbf{x}_i) - p_\ell^{(g)}\big|^q
\end{aligned}
$$

can, by shifting and rescaling, be equivalently expressed as

$$
\sum_{\ell}^{k} \sum_{g\in\{0,1\}} p_\ell^{(g)} \Bigg(\sum_{\substack{i\in G_g \\ h_F(\mathbf{x}_i)\in I_\ell}} (1 - y_i - \lambda f_F(\mathbf{x}_i)) \Bigg) \tag{25}
$$

due to the fact that f_F, y, and $\mathbf{p}$ are binary; each term

$$L_\ell^{(g)} := \sum_{\substack{i \in G_g \\ h_F(\mathbf{x}_i) \in I_\ell}} (1 - y_i - \lambda f_F(\mathbf{x}_i))$$

is constant. Lastly, let

$$N_\ell^{(g)} = \left|\left\{i \in G_g : f_C(\mathbf{x}_i) = 0 \text{ and } h_F(\mathbf{x}_i) \in I_\ell\right\}\right|$$

Thus the optimization of k-QLS can be is equivalently formulated as,

$$\min_{\mathbf{p}^{(0)}, \mathbf{p}^{(1)} \in \{0,1\}^k} \sum_{\ell=1}^{k} \sum_{g \in \{0,1\}} p_\ell^{(g)} L_\ell^{(g)} \tag{26}$$

$$\text{s.t.} \quad -\beta - u \leq \sum_{\ell=1}^{k} p_\ell^{(1)} C_\ell^{(1)} + p_\ell^{(0)} C_\ell^{(0)} \leq \beta - u \tag{27}$$

$$\sum_{\ell=1}^{k} \sum_{g \in \{0,1\}} p_\ell^{(g)} N_\ell^{(g)} \leq \left\lfloor (1-\gamma)\big(1 - \text{PR}(f_C)\big) n \right\rfloor \tag{28}$$

The popularity term $\sum_{\ell=1}^{k} \sum_{g \in \{0,1\}} p_\ell^{(g)} N_\ell^{(g)}$ can take on at most $\lceil n(1 - \text{PR}(f_C)) \rceil$ different values. In the fairness constraint, each term $\sum_{\ell=1}^{k} p_\ell^{(g)} C_\ell^{(g)}$ can take on at most $1/2 |G_g| \big(|G_g| + 1\big)$ unique values since each can be written as

$$\sum_{\ell=1}^{k} p_\ell^{(g)} C_\ell^{(g)} = \sum_{y \in \{0,1\}} a_y (c_{y,1}^{(g)} - c_{y,0}^{(g)}) \quad \text{for some } a_0, a_1 \in \mathbb{N} \text{ with } a_0 + a_1 \leq |G_g|$$

Next we create two index sets which keep track of which groups g and intervals ℓ have positive and negative coefficients $C_\ell^{(g)}$. Let $S_+ = \{(g, \ell) : C_\ell^{(g)} \geq 0\}$ and $S_- = \{(g, \ell) : C_\ell^{(g)} \geq 1\}$. Thus S_+ and S_- indicate whether $p_\ell^{(g)}$ will increase or decrease the value of Equation 27. Specifically, suppose that for each $(g, \ell) \in S_-$, $r_\ell^{(g)}$ is a solution. Let $R = \sum_{(g,\ell) \in S_-} r_\ell(g) p_\ell^{(g)}$, and $\theta = \left\lfloor (1-\gamma)\big(1 - \text{PR}(f_C)\big) n \right\rfloor - \sum_{(g,\ell) \in S_-} r_\ell^{(g)} N_\ell^{(g)}$. Then the problem reduces to solving

$$\begin{aligned} \min \sum_{g,\ell \in S_+} & p_\ell^{(g)} L_\ell^{(g)} \\ & \text{s.t.} \; -\beta - u - R \leq \sum_{g,\ell \in S_+} p_\ell^{(g)} C_\ell^{(g)} \leq \beta - u - R \\ & \sum_{g,\ell} p_\ell^{(g)} N_\ell^{(g)} \leq \theta \end{aligned}$$

which yields a knapsack problem with two constraints, with weights $C_\ell^{(g)}$ and $N_\ell^{(g)}$. Since there are at most k decision variables, $\sum_{g,\ell} p_\ell^{(g)} N_\ell^{(g)}$ can take on at most γn unique feasible values, and $\sum_{g,\ell \in S_+} p_\ell^{(g)} C_\ell^{(g)}$ can take on at most n^2 unique values. This problem is therefore solvable in $\Theta(k\gamma n^3)$ time. Moreover, since any solution set generated from S_- can produce at most n^2 values of R and n values of θ, any configuration of variables from S_- produce γn^3 unique subproblems, each of which can be solved in time $\Theta(k\gamma n^3)$. Thus, Algorithm 2 solves k-QLS in time $\Theta\big(\gamma k n^6\big)$ for general additive metrics. Moreover for PR, TPR, FPR, and ER, each $\sum_{g,\ell \in S_+} p_\ell^{(g)} C_\ell^{(g)}$ can take on at most n unique values (rather than n^2), implying there are only n^2 unique subproblems, each requiring $\Theta(k\gamma n^2)$ time to solve, thus k-QLS is solvable in time $\Theta\big(\gamma k n^4\big)$. □

B Randomized Classifiers

B.1 DOS for Randomized Classification

Theorem (3.3)**.** *Let f_C and f_F be respectively a conventional and β-fair randomized classifier. Let U be derived from an additive efficacy metric $\mathcal{M}$ which is independent of Y (e.g., PR). Then for $q \in \{1, 2, \infty\}$ Program 5 can be solved in time $\Theta(\gamma n^3)$ by Algorithm 1 which returns a γ-popular, β-fair model f_P.*

Algorithm 3: **(Deterministic DOS)** Postprocessing technique, applied directly at prediction time, for converting a deterministic β-fair model f_F into γ-popular β-fair model f_P.

```
input population: $(\mathbf{X}, G)$, $\beta$-fair model: $f_F$, conventional model: $f_C$, popularity: $\gamma$ result: Weight vector $\mathbf{p} \in \{0,1\}^n$ s.t. $f_P = f_F + \mathbf{p}$ is $\gamma$-popular and $\beta$-fair
$\mathbf{p} := \mathbf{0}$
/* positively classifying agents from different groups has a cancelling effect with respect to unfairness */
if sign$(c_1^{(1)} - c_0^{(1)})$ = sign$(c_0^{(0)} - c_0^{(0)})$ then
    $S_g := \{i \in G_g : f_F(\mathbf{x}_i) < f_C(\mathbf{x}_i)\}$
    $a :=$ # of agents that prefer $f_F$
    /* less than $\gamma n$ agent prefer $f_F$ or unfairness is violated */
    while $a < \gamma n$ or $\mathcal{U}\big(f_F(\mathbf{X}) + \mathbf{p}, G\big) > \beta$ do
        $i_0, i_1 := S_0[0], S_1[0]$
        /* positively classify the agents resulting in the lowest increase to unfairness */
        if $\mathbf{p}[i_0] := 1$ increases unfairness less than $\mathbf{p}[i_1] := 1$ then
            $g := 0$
        else
            $g := 1$
        end if
        $\mathbf{p}[i_g] := 1$
        $S_g$.delete$(i_g)$
        $a$ += 1
    end while
    return $\mathbf{p}$
    /* positively classifying agents from different groups has a monotonic effect with respect to unfairness */
else if sign$(c_1^{(1)} - c_0^{(1)}) = -$sign$(c_1^{(0)} - c_0^{(0)})$ then
    /* increase positives on one group and decrease on the other (for each group) */
    for $g' \in \{0,1\}$ do
        $S_{g'} := \{i \in G_g : f_F(\mathbf{x}_i) < f_C(\mathbf{x}_i)\}$/* all agents from group $G_{g'}$ who prefer $f_C$ */
        /* all agents in $G_{(1-g')}$ positively classified under $f_F$, sorted by $f_C$ */
        $A_{(1-g')} := \{i \in G_{(1-g')} : f_F(\mathbf{x}_i) == 1\}$   s.t. $f_C(\mathbf{x}_i) > f_C(\mathbf{x}_{i+1})$
        $a :=$ # of agents preferring $f_F$
        /* less than $\gamma n$ agents prefer $f_F$ or unfairness is violated */
        while $a < \gamma n$ or $\mathcal{U}\big(f_F(\mathbf{X}) + \mathbf{p}, G\big) > \beta$ do
            /* if unfairness is violated, attempt to fix it */
            if $\mathcal{U}(f_F(\mathbf{X}) + \mathbf{p}, G) > \beta$ then
                $i, j = S_{g'}[0], A_{(1-g')}[0]$
                if unfairness decreased by $\mathbf{p}^{(g')}[i] := 1$ then
                    $\mathbf{p}^{(g')}[i] := 1$
                else
                    $\mathbf{p}^{(1-g')}[j] := -1$
                end if
                /* if unfairness is not violated, increase the positive rate on $G_{g'}$ */
            else
                $\mathbf{p}^{(g')}[i] := 1$
            end if
            update $a$, (+1 or -1)
        end while
    end for
end ifreturn $\mathbf{p}$
```

Algorithm 4: Algorithm to solve programs associated with DOS in the randomized classification setting, when given S, a specific set of γn that must prefer f_P to f_C.

```
input: population: $(\mathbf{X}, Y, G)$, $\beta$-fair model: $f_F$,  conventional model: $f_C$, $\gamma n : S$
result: Weight vector $\mathbf{p}$ s.t. $f_P = f_F + \mathbf{p}$
$\mathbf{p} = \mathbf{0}$
$m := \gamma n$
$\boldsymbol{\delta} := h_F(\mathbf{X})$ /* lower bound on perturbation to agents' scores */
for $i \in S$ do
    $p_i := \max\big(0, h_C(\mathbf{x}_i) - h_F(\mathbf{x}_i)\big)$; /* minimum score increase for i to prefer $f_P$ */
    $\delta_i := h_C(\mathbf{x}_i) - h_F(\mathbf{x}_i)$; /* lowest value of $p_i$ such that i prefers $f_P$ */
end for
/* check "direction" in which fairness is violated */
if $\big|\mathcal{M}\big(f_F(\mathbf{X}) + \mathbf{p}, Y; g = 1\big) - \mathcal{M}\big(f_F(\mathbf{X}) + \mathbf{p}, Y; g = 0\big) < -\beta$ then
    /* $s_g$=1 ($s_g$=-1) indicates increasing (decreasing) scores on $G_g$ */
    $s_g := \text{sign}\big((1-2g)(c_1^{(g)} - c_0^{(g)})\big)$ for $g \in \{0, 1\}$
else if $\big|\mathcal{M}\big(f_F(\mathbf{X}) + \mathbf{p}, Y; g = 1\big) - \mathcal{M}\big(f_F(\mathbf{X}) + \mathbf{p}, Y; g = 0\big) > \beta$ then
    $s_g := \text{sign}\big((1-2g)(c_0^{(g)} - c_1^{(g)})\big)$ for $g \in \{0, 1\}$
end if
while $\mathcal{U}\big(f_F + \mathbf{p}, \mathcal{D}\big) > \beta$ do
    /* agents from each group whose scores can still be perturbed */
    if $s_g = 1$ then
        $G_g := \{i \in G_g : h_F(\mathbf{x}_i) + p_i < 1\}$
        /* maximum additional perturbation to $G_g$ which is feasible */
        $\varepsilon_g := \min\left(\big\{1 - h_F(\mathbf{x}_i) : i \in G_g\big\} \cup \big\{\frac{\beta - \mathcal{U}\left(f_F + \mathbf{p}, \mathcal{D}\right)}{(c_1^{(g)} - c_0^{(g)})|G_g|}\big\}\right)$
    else
        $G_g := \{i \in G_g : \delta_i < h_F(\mathbf{x}_i) + p_i\}$
        /* maximum additional perturbation to $G_g$ which is feasible */
        $\varepsilon_g := \min\left(\big\{\delta_i : i \in G_g\big\} \cup \big\{\frac{\beta - \mathcal{U}\left(f_F + \mathbf{p}, \mathcal{D}\right)}{(c_0^{(g)} - c_1^{(g)})|G_g|}\big\}\right)$
    end if
    /* check if increasing scores by $\varepsilon_0, \varepsilon_1$ would fix fairness */
    $\mathbf{p}' := \mathbf{p}$
    $\mathbf{p}'[G_g] += \varepsilon_g$ for $g \in \{0, 1\}$
    if $\mathcal{U}\big(f_F + \mathbf{p}', \mathcal{D}\big) < \beta$ then
        /* in the case that fairness is achieved, $\varepsilon_g$ may be too large */
        decrease magnitude of each $\varepsilon_g$ s.t. $\mathcal{U}\big(f_F + \mathbf{p}', \mathcal{D}\big) = \beta$ and $\sum_{i \in G_0} \big|p_i + \varepsilon_0\big|^q = \sum_{i \in G_1} \big|p_i + \varepsilon_1\big|^q$
        return $\mathbf{p}'$
    end if
    $\mathbf{p}_g := \mathbf{p}[G_g] + \varepsilon_g$     for $g \in \{0, 1\}$
    /* ratio of "fairness repair" to increase in loss */
    $g^* := \arg\min_{g \in \{0,1\}} \big(\frac{\varepsilon_g |G_g| s_g (c_1^{(g)} - c_0^{(g)})}{\|\mathbf{p}_g\|_q - \|\mathbf{p}\|_q}\big)$ /* where $\frac{a}{0} := \infty$ for $a \neq 0$ and $\frac{0}{0} := 0$ */
    $\mathbf{p} := \mathbf{p}_{g^*}$
end while
return $\mathbf{p}$
```

Proof of Theorem 3.3. Recall that $\mathbb{E}\big[f_F(\mathbf{x})\big] = h_F(\mathbf{x})$ and $\mathbb{E}\big[f_C(\mathbf{x})\big] = h_C(\mathbf{x})$, i.e., the expected outcome of each classifier is given by is respective score function. For notational convince we use we use h_F and h_C throughout this proof. Program 5 is non-convex with respect to $\mathbf{p}$ due to the constraint that γ-fraction of the population needs to prefer f_F over f_C, namely that $m = \gamma n$ of the constraints

$$h_C(\mathbf{x}_i) \leq h_F(\mathbf{x}_i) + p_i$$

need to be satisfied. However, note that if instead of needing to satisfy *any* m constraints, we needed to satisfy a specific set of m constraints, say

$$S = \big\{h_C(\mathbf{x}_{i_1}) \leq h_F(\mathbf{x}_{i_1}) + p_{i_1}, \dots, h_C(\mathbf{x}_{i_m}) \leq h_F(\mathbf{x}_{i_m}) + p_{i_m}\big\},$$

then the resulting program would be trivial to solve as it amounts to ℓ_q-norm minimization subject to linear constraints. Thus, if the optimal set of popularity constraints can be found efficiently, the problem is polynomial time solvable.

To find this set of constraints, we make use of the fact that the metric $\mathcal{M}$ defining $\mathcal{U}$ is additive, specifically the fact that $\mathcal{U}$ can be expressed in terms of scalars $c_1^{(g)}, c_0^{(g)} \in [0,1]$ which give the respective cost of positively or negatively classifying an agent from group G_g. That is, given a perturbation $\mathbf{p} \in [-1,1]^n$ and fair model f_F, unfairness can be written as,

$$\begin{aligned}
&\mathcal{U}\big(h_F(\mathbf{X}) + \mathbf{p}, G\big)\\
=&\big|\mathcal{M}\big(h_F(\mathbf{X}) + \mathbf{p} : g = 1\big) - \mathcal{M}\big(h_F(\mathbf{X}) + \mathbf{p} : g = 0\big)\big|\\
=&\Bigg|\sum_{i \in G_1} c_1^{(1)}\big(h_F(\mathbf{x}_i) + p_i\big) + c_0^{(1)}\big(1 - \big(h_F(\mathbf{x}_i) + p_i\big)\big) - \sum_{j \in G_0} c_1^{(0)}\big(h_F(\mathbf{x}_j) + p_j\big) + c_0^{(0)}\big(1 - \big(h_F(\mathbf{x}_j) + p_j\big)\big)\Bigg|\\
=&\Bigg|\bigg(\sum_{i \in G_1}(c_1^{(1)} - c_0^{(1)})p_i - \sum_{j \in G_0}(c_1^{(0)} - c_0^{(0)})p_j\bigg)\\
&\qquad + \bigg(\sum_{i \in G_1} c_1^{(1)} h_F(\mathbf{x}_i) + c_0^{(1)}\big(1 - h_F(\mathbf{x}_i)\big) - \sum_{j \in G_0} c_1^{(0)} h_F(\mathbf{x}_j) - c_0^{(0)}\big(1 - h_F(\mathbf{x}_j)\big)\bigg)\Bigg|
\end{aligned}$$

Since $c_0^{(g)}$, $c_1^{(g)}$, and $h_F(\mathbf{X})$ are constant

$$u := \sum_{i \in G_1} c_1^{(1)} h_F(\mathbf{x}_i) + c_0^{(1)}\big(1 - h_F(\mathbf{x}_i)\big) - \sum_{j \in G_0} c_1^{(0)} h_F(\mathbf{x}_j) - c_0^{(0)}\big(1 - h_F(\mathbf{x}_j)\big)$$

is also constant. Thus the fairness constraint can be expressed as

$$\begin{aligned}
&\mathcal{U}\big(h_F(\mathbf{X}) + \mathbf{p}, G\big) \leq \beta\\
\iff& -\beta - u \leq \sum_{i \in G_1}(c_1^{(1)} - c_0^{(1)})p_i - \sum_{j \in G_0}(c_1^{(0)} - c_0^{(0)})p_j \leq \beta - u.
\end{aligned} \tag{29}$$

With this formulation of unfairness, we see that for any two agents i_1, i_2 from the same group, increasing or decreasing the score of i_1 has the same effect on unfairness as equivalently increasing or decreasing the score of i_2. More specifically, let $i_1, i_2 \in G_g$, then for any potential solution $\mathbf{p}$, let $\mathbf{p}'$ be any other potential solution with $p'_j = p'_j$ if $j \neq i_1, i_2$, and $p_{i_1} + p_{i_2} = p'_{i_1} + p'_{i_2}$. Both $\mathbf{p}$ and $\mathbf{p}'$ have equivalent fairness. This observation can be used to order both groups in terms of increase in p_i required for agent i to prefer f_P over f_C.

To induce this ordering, consider any two agents $i_1, i_2 \in G_g$ with $h_C(\mathbf{x}_{i_1}) - h_F(\mathbf{x}_{i_1}) \leq h_C(\mathbf{x}_{i_2}) - h_F(\mathbf{x}_{i_2})$, i.e., i_1 requires at least as large a score shift as i_2 to prefer f_F over f_C. Let S_1 be any set of m popularity constraints which include $h_C(\mathbf{x}_{i_1}) \leq h_F(\mathbf{x}_{i_1}) + p_{i_1}$, but not $h_C(\mathbf{x}_{i_2}) \leq h_F(\mathbf{x}_{i_2}) + p_{i_2}$, and let

$$S_2 = \bigg(S_1 \setminus \{h_C(\mathbf{x}_{i_1}) \leq h_F(\mathbf{x}_{i_1}) + p_{i_1}\}\bigg) \cup \{h_C(\mathbf{x}_{i_2}) \leq h_F(\mathbf{x}_{i_2}) + p_{i_2}\}.$$

Let $\mathbf{p}_1$ and $\mathbf{p}_2$ be the solutions corresponding to Program 5 with constraint set S_1 and S_2 respectively. Then $\|\mathbf{p}_2\|_q \leq \|\mathbf{p}_1\|_q$. That is, choosing to enforce that i_2 prefers f_F over f_C is at least as good as choosing to enforce the preference of i_1 for f_F over f_C. To see this, consider the the scores of agents i_1 and i_2 under solution $\mathbf{p}_1$, i.e. $f_F(\mathbf{x}_{i_1}) + p_{1,i_1}$ and $f_F(\mathbf{x}_{i_2}) + p_{1,i_2}$. Suppose that scores p_{1,i_1} and p_{1,i_2} are permuted creating $\mathbf{p}'_1$, i.e. $p'_{1,i_1} = p_{1,i_2}$ and $p'_{1,i_2} = p_{1,i_1}$. Then $\mathbf{p}_1$ and $\mathbf{p}'_1$, have equal unfairness. Moreover, by the construction of S_1 and S_2 it must be the case that

$$p_{1,i_1} \geq f_C(\mathbf{x}_{i_1}) - f_F(\mathbf{x}_{i_1}) \geq f_C(\mathbf{x}_{i_2}) - f_F(\mathbf{x}_{i_2}),$$

implying that $\mathbf{p}'_1$ constitutes to a feasible solution to the program corresponding to popularity constraints S_2, i.e. $\|\mathbf{p}'_1\|_q \geq \|\mathbf{p}_2\|_q$. Since permuting elements of $\mathbf{p}_1$ does not affect the value of any ℓ_q-norm, it must be the case that $\|\mathbf{p}_1\|_q = \|\mathbf{p}'_1\|_q \geq$

$\|\mathbf{p}_2\|_q$. Thus if groups are ordered such that for $i \in G_g$ we have $h_C(\mathbf{x}_i) - h_F(\mathbf{x}_i) < h_C(\mathbf{x}_{i+1}) - h_F(\mathbf{x}_{i+1})$, then one need only consider adding the constraint $h_C(\mathbf{x}_{i+1}) \leq h_F(\mathbf{x}_{i+1}) + p_{i+1}$ if the constraint $h_C(\mathbf{x}_i) \leq h_F(\mathbf{x}_i) + p_i$ has already been selected.

Suppose G_1 and G_0 are ordered in such a manner. Then, to decide which constraints to include, it suffices to determine the intergroup decisions since the intragroup decisions are then determined by the agent order. Since there are at most $m = \gamma n$ unique sets of constraints which preserve orderings within groups, and each set of constraints corresponds to a polynomial time solvable program, Program 5 is solvable in time $\Theta(\gamma n T)$ where $\Theta(T)$ is the time required to solve a single program (either a linear program or a semidefinite program). Moreover, each corresponding program (namely Program 5 with Constraint 7 replaced by S) can be solved by Algorithm 4. At high level this algorithm takes the agents in S (i.e., the set of agents which should prefer f_P) and sets each p_i to the minimum value, in terms of magnitude, such that $i \in S$ prefers f_P. If fairness is not violated by this change to p_i is optimal. In the case when fairness is violated, the algorithm iteratively increases (or decreases) elements of $\mathbf{p}$ such that unfairness is strictly decreasing while minimally increasing $\|\mathbf{p}\|_q$.

To see the optimality and running time of Algorithm 4, consider the first step, namely $p_i = \min\big(0, h_C(\mathbf{x}_i) - h_F(\mathbf{x}_i)\big)$ for all $i \in S$. This setting of $\mathbf{p}$ causes all agents in S to prefer f_P and is clearly the minimum perturbation to do so. Therefore, if fairness is not violated, then $\mathbf{p}$ is optimal, and the running time is $\Theta(\gamma n)$.

In the case that fairness is violated, the scores on groups G_0 and G_1 need to be further altered. In particular, let

$$U(f_p, \mathbf{X}, G) = \mathcal{M}(f_P(\mathbf{X}); g = 1) - \mathcal{M}(f_P(\mathbf{X}); g = 0)$$

i.e., U is $\mathcal{U}$ without absolute value. The rate of change in U with respect to increasing p_i is given by $c_1^{(g_i)} - c_0^{(g_i)}$ for each $i \in [n]$. Therefore, if $U(f_p, \mathbf{X}, G) < -\beta$, unfairness can only be fixed by increasing scores on each group G_g with $c_1^{(g)} - c_0^{(g)} > 0$ and decreasing scores on each group G_G with $c_1^{(g)} - c_0^{(g)} < 0$. On the other hand, if $U(f_p, \mathbf{X}, G) > \beta$, then unfairness can only be fixed by increasing scores on each group G_g with $c_1^{(g)} - c_0^{(g)} < 0$ and decreasing scores on each group G_G with $c_1^{(g)} - c_0^{(g)} > 0$. When increasing scores on G_g the only constraint is that $h_F(\mathbf{x}_i) + p_i \leq 1$ for all $i \in [n]$, but when decreasing scores the constraint $0 \leq h_F(\mathbf{x}_i) + p_i$ for all $i \in [n]$ needs to be considered as well as $h_C(\mathbf{x}_j) \leq h_F(\mathbf{x}_j) + p_j$ for all $j \in S$.

With respect to fairness agents from the same group are exchangeable in the sense that increasing (or decreasing) the score of any agent in G_g has the same effect on unfairness as increasing (or decreasing) the score of any other agent in G_g. Specifically, for $i, j \in G_g$ unfairness is invariant to any change in p_i, p_j which preservers $p_i + p_j$. Therefore, ignoring popularity, no optimal solution will set $p_i < 0$ and $p_j > 0$. Moreover, when $p_i + p_j$ must be preserved, the terms $|p_i| + |p_j|$, $p_i^2 + p_j^2$ and $\max\big(|p_i|, |p_j|\big)$ are all minimized when $p_i = p_j = \frac{p_i + p_j}{2}$. Therefore for $q \in \{1, 2, \infty\}$, if one where to increase $|p_i|$, say by value ε, then $p_i + \text{sign}(p_i)\varepsilon$ is no better than $p_j + \frac{\text{sign}(p_i)\varepsilon}{|G_{g_i}|}$ for each $j \in G_{g_i}$. That is, ignoring popularity, it is optimal to uniformly distribute the weight of $\mathbf{p}$ over each group.

When considering both popularity constraints, as well as the need for perturbations to constitute valid probabilities, it then optimal to uniformly increase the weight on all agents $i \in G_g$ such that neither of these constraints is violated. This value is given as ε_g at each iteration. Let $\mathbf{p}$ be the solution produced by Algorithm 4 and let $\mathbf{p}^*$ be any optimal solution. Since both are solutions $h_F(\mathbf{X}) + \mathbf{p}$ and $h_F(\mathbf{X}) + \mathbf{p}^*$ are both β-fair, and for each $j \in S$ $h_F(\mathbf{x}_j) + p_j \geq h_C(\mathbf{x}_j)$ and $h_F(\mathbf{x}_j) + p_j^* \geq h_C(\mathbf{x}_j)$. If $\|\mathbf{p}\|_q \leq \|\mathbf{p}^*\|_q$, then $\mathbf{p}$ is also an optimal solution. Assume, by way of contradiction, that $\|\mathbf{p}\|_q > \|\mathbf{p}^*\|_q$, and consider two cases: 1.) $\mathcal{U}(h_F(\mathbf{X}) + \mathbf{p}^*, G) < \beta$ and 2.) $\mathcal{U}(h_F(\mathbf{X}) + \mathbf{p}^*, G) = \beta$.

In case (1), if $q = 1, 2$ then $i \notin S$ implies $p_i^* = 0$, and if $q = \infty$ then $i \notin S$ implies $|p_i^*| \leq \max_{j \in S}\big(|p_j^*|\big)$. To see this, let $q = 1, 2$ and $j \notin S$. Suppose that $|p_i^*| > 0$ and $u = \beta - \mathcal{U}(h_F(\mathbf{X}) + \mathbf{p}^*, G)$. Then $|p_i^*|$ can be decreased by at least $\frac{u}{|c_1^{(g_i)} - c_0^{(g_i)}|}$ without violating fairness. Doing so results in a strict decrease to $\|\mathbf{p}^*\|_q$. When $q = \infty$ and identical argument holds for $|p_i^*| > \max_{j \in S}\big(|p_j^*|\big)$.

In case (2), we order each group according to the maximum feasible perturbation to each agent, w.l.o.g. we show this for G_0 when $c_1^{(0)} - c_0^{(0)} > 0$ (a symmetric argument holds in other cases). For $i \in G_0$ let $\delta_i = -h_F(\mathbf{x}_i)$ if $i \notin S$ and $\delta_j = h_C(\mathbf{x}_i) - h_F(\mathbf{x}_i)$ if $i \in S$. Order G_0 such that for $i, j \in G_0$, $i < j$ implies $\delta_i \geq \delta_j$. Suppose that $\delta_j \leq p_i^* \leq p_j^* \leq 0$. Then any solution which has $p_i = \frac{p_i^* + p_j^*}{2}$ (such as $\mathbf{p}'$) is both feasible and at least as optimal as $\mathbf{p}^*$.

At each iteration the sets G_g represent the set of agents whose scores can feasibly still be perturbed, i.e. further perturbing will not push the score below 0, above 1, or violate a constraint in S. The entries of $\mathbf{p}$ corresponding to either G_0 or G_1 are updated by ε_0 or ε_1 respectively. By the definition of ε_g, at least one agent is removed from either G_0 or G_1. There are at most n agents between the two sets, and thus at most n iterations are run. Each iteration takes at most time time $\Theta(n)$, since ε_g is computed as the minimum over at most n choices and at most n entries of $\mathbf{p}$ are updated. Therefore Algorithm 4 runs in time $\Theta(n^2)$.

Thus, since Algorithm 4 may be used to solve the instances of Program 5 arising in Algorithm 1 in time $\Theta(n^2)$, DOS can be solved by Algorithm 1 in time $\Theta(\gamma n^3)$. □

B.2 DOS for Randomized Resource Allocation

In practice, when sampling using these probabilities to produce an actual allocation, an invalid solution may be obtained, i.e. if X_i is a binary indicator of agent i receiving a resource, sampling can yield $\sum_{i=1}^{n} X_i > k$. Provided that the number of resources is large enough, we can control this probability by solving for a slightly smaller bound. Suppose we instead use the constraint $\sum_{i=1}^{n} I_F(i) + p_i \leq (1-\varepsilon)k$ for some $\varepsilon \in [0,1]$. Then applying the Chernoff bound yields

$$\mathbb{P}\bigg(\sum_{i=1}^{n} X_i > k\bigg) \leq \exp\big(-\frac{\epsilon^2}{2(1-\varepsilon)}k)\big),$$

which, for example, implies that when $k \geq 500$ and any $\varepsilon \geq 0.2$ the probability of getting an invalid solution is no greater than 4×10^{-6}.

Theorem (3.4)**.** *Let I_C and I_F be, respectively, conventional and β-fair allocation schemes, and U be derived from an additive efficacy metric $\mathcal{M}$ which is independent of Y (e.g., PR). Then for $q \in \{1, 2, \infty\}$ Program 8 can be solved in time $\Theta(\gamma nT)$ (where $\Theta(T)$ is the time required to solve either a linear program or a semidefinite program) by Algorithm 1 which returns a γ-popular, β-fair allocation if one exists.*

Proof of Theorem 3.4. The proof of this theorem follows a similar line of reasoning to that used in Theorem 3.3, namely in that if a specific set of $m = \lceil \gamma n \rceil$ popularity constraints (rather than *any* m popularity constraints) needed to be satisfied then the problem is tractable. To select these m constraints efficiently, we use the ordering in Theorem 3.3. The key difference in this case is the existence of the resource constraint $\sum_{i=1}^{n} I_F(i) + p_i \leq k$, and thus we need only show that the ordering induced on agents is not invalidated by the addition of this resource constraint. To see that this is indeed the case, note that each p_i has the same coefficient in this constraint, namely 1. Therefore, permuting any p_i and p_j does not affect this constraint. Next consider any two agents $i_1, i_2 \in G_g$ with $I_C(i_1) - I_F(i_1) \leq I_C(i_2) - I_F(i_2)$, i.e., i_1 requires at least as large a score shift as i_2 to prefer I_F over I_C. Let S_1 be any set of m popularity constraints which includes $I_C(i_1) \leq I_F(i_1) + p_{i_1}$ but not $I_C(i_2) \leq I_F(i_2) + p_{i_2}$, and let

$$S_2 = \bigg(S_1 \setminus \{I_C(i_1) \leq I_F(i_1) + p_{i_1}\}\bigg) \cup \{I_C(i_2) \leq I_F(i_2) + p_{i_2}\}.$$

Let $\mathbf{p}_1$ and $\mathbf{p}_2$ be the solutions corresponding to Program 8 with constraint set S_1 and S_2 respectively. By identical reasoning to the classification case, if both programs are feasible, then $\|\mathbf{p}_2\|_q \leq \|\mathbf{p}_1\|_q$, since score permutations within a group do not affect the resource constraint, objective, or unfairness. Moreover, since $I_C(i_1) - I_F(i_1) \geq I_C(i_2) - I_F(i_2)$, if the program corresponding to S_1 is feasible, then the program corresponding to S_2 must also be feasible, and thus $\mathbf{p}_1$ can be transformed into a feasible solution to S_2 via score permutation. Similarly, in the case when S_2 is infeasible, S_1 must also be infeasible.

Since adding constraints can never cause an infeasible program to become feasible, if infeasible programs are considered to have solution value ∞, then including agent i_2 in place of i_1 will never result in an increase in the objective value. Thus each group can again be ordered by $I_C(i) - I_F(i)$. Iterating over each of the the m possible choices of constraints from G_0 and G_1 which preserve this intragroup ranking, will return an optimal solution if one exists, or determine that the problem has no solution. Hence, Algorithm 1 solves Program 8. Since at most $m = \gamma n$ such matchings are examined and each matching corresponds to a program which requires $\theta(T)$ time to solve Algorithm 1 runs in time $\Theta(\gamma nT)$. □

B.3 k-QLS

Theorem B.1 (3.5)**.** *Postprocessing to achieve γ-popularity β-fairness with k-QLS (i.e., solving Program 12) is strongly NP-hard when models are randomized, $\mathcal{U}$ is derived from an additive fairness metric, and the number of quantiles k is determined by the input.*

Proof of Theorem 3.5. We reduce from the NP-hard problem exact m knapsack (EmKP), which is strongly NP-hard when coefficients are rational, which consists of n items, each with weight and value $w_i, v_i \in \mathbb{Q}_{\geq 0}$, a capacity $W \in \mathbb{Q}_{\geq 0}$, and a target m. The objective is to select exactly m items such that total value is maximized and the weight limit is not exceeded. To transform an instance of EmKP into an instance of k-QLS postprocessing, we map each item to an interval where the item weight corresponds to unfairness, item value corresponds to loss, and popularity is achieved when exactly m intervals have nonzero values of $p_\ell^{(g)}$. Specifically, for each item i create two agents i_0, i_1 such that for agent i_0, $g_{i_0} = y_{i_0} = 0$, and for i_1, $g_{i_1} = y_{i_1} = 1$. For the conventional and fair score function $h_C = \mathbb{E}[f_C], h_F = \mathbb{E}[f_F]$, let

$$h_C(\mathbf{x}_{i_0}) = \frac{w_i + \max_{j \in [n]}(w_j)}{2\max_{j \in [n]}(w_j)} \text{ and } h_F(\mathbf{x}_{i_0}) = \frac{v_i - 3W\big(1 + 2h_C(\mathbf{x}_{i_0})\big)\max_{j \in [n]}(v_j)}{4Wh_C(\mathbf{x}_{i_0})\max_{j \in [n]}(v_j)} \tag{30}$$

and,

$$h_C(\mathbf{x}_{i_1}) = h_F(\mathbf{x}_{i_1}) = 1.$$

In Equation 30 note that $1/2 \leq h_C(\mathbf{x}_{i_0}) \leq 1$ and as such $0 \leq h_F(\mathbf{x}_{i_0}) \leq 1$. The particular values of both variables is selected to ensure that both h_C and h_F correspond to valid probabilities, and so that the loss and fairness constraint cancel out to yield a weight constraint over w_i and a maximize over v_i. Let the efficacy costs be defined as $c_{0,1}^{(0)} = c_{0,1}^{(1)} = 1$ and all others are 0, i.e. false positive fairness. Lastly let the popularity coefficient be $\gamma = \frac{n+m}{2n}$, maximum unfairness be $\beta = \frac{W}{2\max_{j\in[n]}(w_j)} + \frac{m}{2}$, the number of intervals be $k = 2n$, and the regularization coefficient be $\lambda = 1/2$. Note that each of the k intervals then contains exactly one agent.

The key idea is that the construction of the groups, and choice of fairness definition, causes any optimal solution to positively classify all agents in G_1 since $g_j = y_j = 1$ for all $j \in G_1$. Doing so yields 0 loss on G_1 and makes no contribution to unfairness (since fairness is defined by FPR). Moreover, ignoring popularity, any optimal solution will negatively classify all agents in G_0 since $g_j = y_j = 0$ for all $j \in G_0$ and doing so yields 0 loss on G_0 and makes no false positive predictions. When adding the popularity constraint, i.e. $n + m$ of the $2n$ agents must have a an expected outcome under h_F which is at least as large as the expected outcome under h_C, the decisions on G_1 will remain invariant, but an optimal solution will select the lowest possible number of agents in G_0 (namely m) minimally increasing loss and not violating unfairness, and classify those agents positively with probability $h_C(\mathbf{x}_j)$ (i.e., their expected outcome under the conventional classifier). By the construction of the h_C and h_F in Equation 30, these m agents will correspond to most profitable m items which do not exceed the weight limit.

To see why this is the case we first consider the loss term on each agent j in G_0 when that agent has expected outcome $p_j^{(0)}$,

$$\begin{aligned}&p_j^{(0)}(1-y_j) + (1-p_j^{(0)})y_j + \lambda(h_F(\mathbf{x}_j) - p_j^{(0)})^2\\ =&p_j^{(0)}\big(1 + 1/2 p_j^{(0)} - h_F(\mathbf{x}_j)\big) + 1/2 h_F(\mathbf{x}_j)^2\end{aligned}$$

since $0 \le h_F(\mathbf{x}_j) \le 1$, this term is monotonically increasing in $p_j^{(0)}$ and is minimized at $p_j^{(0)} = 0$. Thus without consideration of popularity or unfairness, the optimal solution is to set $p_j^{(0)} = 0$ for all $j \in G_0$. Moreover, by construction of the fairness cost coefficients $c_{0,1}^{(0)} = c_{0,1}^{(1)} = 1$, the fairness constraint can be written as

$$\begin{aligned}&\mathcal{U}(f_p, \mathbf{X}, Y, G) \le \beta\\ \iff& -\beta \le \sum_{i\in G_1} p_i^{(1)} c_{y_i,1}^{(1)} + (1-p_i^{(1)})c_{y_i,0}^{(1)} - \sum_{j\in G_0} p_j^{(0)} c_{y_j,1}^{(0)} + (1-p_j^{(0)})c_{y_j,0}^{(0)} \le \beta\\ \iff& \sum_{j\in G_0} p_j^{(0)} c_{0,1}^{(0)} \le \beta\end{aligned}$$

since $c_{0,1}^{(0)} = 1$, the left-hand side of the inequality is monotonically increasing in each $p_j^{(0)}$. Therefore, the fairness constraint adds no incentive to increase any $p_j^{(0)}$ on group 0, and thus only the popularity constraint will force $p_j^{(0)} > 0$ for some j.

Since $\gamma 2n = \frac{m+n}{2n} 2n = m + n$ number of agents need to prefer f_P to f_C (i.e., need $p_i^{(g)} \ge h_C(\mathbf{x}_i)$), and all n of the agents in G_1 trivially prefers f_P, the popularity constraint is satisfied only when m agents from G_0 prefer f_P.

Note that since both the unfairness term $\sum_{j\in G_0} p_j^{(0)} c_{0,1}^{(0)}$ and the loss term $\sum_{j\in G_0} p_j^{(0)}\big(1 + 1/2 p_j^{(0)} - h_F(\mathbf{x}_j)\big) + 1/2 h_F(\mathbf{x}_j)^2$ corresponding to G_0 are both monotonically increasing in each $p_j^{(0)}$, the optimal solution is to set exactly m of the n variables $p_j^{(0)}$ to $h_C(\mathbf{x}_j)$ (i.e. the lowest possible value such that agent j prefers f_P to f_C). Let $\alpha_j \in \{0, 1\}$ correspond to an indicator that $p_j^{(0)} = h_C(\mathbf{x}_j)$, then k-QLS is equivalent to

$$\min_{\boldsymbol{\alpha}} \sum_{j\in G_0} \alpha_j \Big(h_C(\mathbf{x}_j)\big(1 + 1/2 h_C(\mathbf{x}_j) - h_F(\mathbf{x}_j)\big) + 1/2 h_F(\mathbf{x}_j)^2\Big) + (1-\alpha_j)(1/2 h_F(\mathbf{x}_j)^2) \tag{31}$$

$$\text{s.t.} \sum_{j\in G_0} \alpha_j h_C(\mathbf{x}_j) \le \beta \tag{32}$$

$$\sum_{j\in G_0} \alpha_j = m. \tag{33}$$

Simplifying the objective and substituting the expressions for $h_C(\mathbf{x}_j)$ and $h_F(\mathbf{x}_j)$ yields

$$\begin{aligned}
&\min_{\boldsymbol{\alpha}} \sum_{j\in G_0} \alpha_j \Big(h_C(\mathbf{x}_j)\big(1 + {}^1\!/\!_2 h_C(\mathbf{x}_j) - h_F(\mathbf{x}_j)\big) + {}^1\!/\!_2 h_F(\mathbf{x}_j)^2 \Big) + (1-\alpha_j)({}^1\!/\!_2 h_F(\mathbf{x}_j)^2) \\
\iff &\min_{\boldsymbol{\alpha}} \sum_{j\in G_0} \alpha_j \Big(h_C(\mathbf{x}_j)\big(1 + {}^1\!/\!_2 h_C(\mathbf{x}_j) - h_F(\mathbf{x}_j)\big) \Big) + {}^1\!/\!_2 h_F(\mathbf{x}_j)^2 \\
\iff &\min_{\boldsymbol{\alpha}} \sum_{j\in G_0} \alpha_j \Big({}^3\!/\!_4 - \frac{v_j}{4W \max_{i\in G_0}(v_i)}) \\
\iff &\max_{\boldsymbol{\alpha}} \sum_{j\in G_0} \alpha_j \frac{v_j}{4W \max_{i\in G_0}(v_i)} - \sum_{j\in G_0} \alpha_j {}^3\!/\!_4 \\
\iff &\max_{\boldsymbol{\alpha}} \sum_{j\in G_0} \alpha_j \frac{v_j}{4W \max_{i\in G_0}(v_i)}
\end{aligned}$$

where the final line stems from the fact that $\sum_{j\in G_0} = m$ and is thus a constant term, not affecting the optimization. Moreover, note that the denominator $4W \max_{i\in G_0}(v_i)$ is also constant, thus minimizing (31) is equivalent to maximizing the value of the knapsack.

Lastly, we need only show that the fairness term is equivalent to the original capacity constraint. The fairness constraint can be written then as

$$\begin{aligned}
&\sum_{j\in G_0} \alpha_j h_C(\mathbf{x}_j) \le \beta \\
\iff &\sum_{j\in G_0} \alpha_j \frac{w_i + \max_{j\in[n]}(w_j)}{2\max_{j\in[n]}(w_j)} \le \frac{W}{2\max_{j\in[n]}(w_j)} + \frac{m}{2} \\
\iff &\sum_{j\in G_0} \alpha_j \big(w_i + \max_{j\in[n]}(w_j)\big) \le W + m \max_{j\in[n]}(w_j) \\
\iff &\sum_{j\in G_0} \alpha_j w_i \le W
\end{aligned}$$

where the last line is again due to $\sum_{j\in G_0} \alpha_j = m$. Thus the fairness constraint is satisfied if and only if the original capacity constraint is satisfied. Thus, any solution to k-QLS which successfully minimizes loss such that unfairness is not violated and at least m agents from G_0 prefer f_P can be used as an optimal solution to the original EmKP problem by simply selecting all items j which correspond to nonzero values of $p_j^{(0)}$. Since EmKP is strongly NP-hard, so is k-QLS postprocessing on randomized classifiers. □

C Experiments

C.1 Relationship Between Popularity and Fairness for Standard Fair Learning Schemes

In Figure 4 the percentage of the population who prefers for f_F over F_C (y-axis) is shown as a function of unfairness (left) and error (right) for three choices of fairness metric and dataset, when classifier decisions are randomized. The randomness in this example comes not from the stochastic nature of the classifiers, but from uncertainty in the training data. Each classifier, trained on a different down sample of the training data is deterministic. Expected classifier outcome is then given as the expectation over all classifiers, rather than a single classifier's predicted probability. We see that current fair learning schemes produce popular classifiers at rates less than chance. Moreover we see that in most cases there is no clearly defined Pareto front on which most examples sit, implying that finding a classifier which is both popular and fair may be feasible in practice.

In Figures 5, 6 we see the relative popularity of f_C and f_P for randomized classifiers (top) and deterministic classifiers (bottom).

C.2 Running time of DOS and k-QLS

For DOS and k-QLS, the corresponding polynomial time solutions for randomized models may require solving large numbers of linear programs or semidefinite programs (semidefinite programs appear only for l_2-norm regularization). Despite their polynomial time guarantees, these algorithms can sometimes be slow in practice. To deal with such cases model designers have two options: 1.) solve the programs in parallel, since each program is independent, and 2.) frame solve the integer program corresponding to DOS or k-QLS. In practice we find the latter to be much faster on average. For deterministic models this is not the case since neither algorithm requires the use of program solvers.

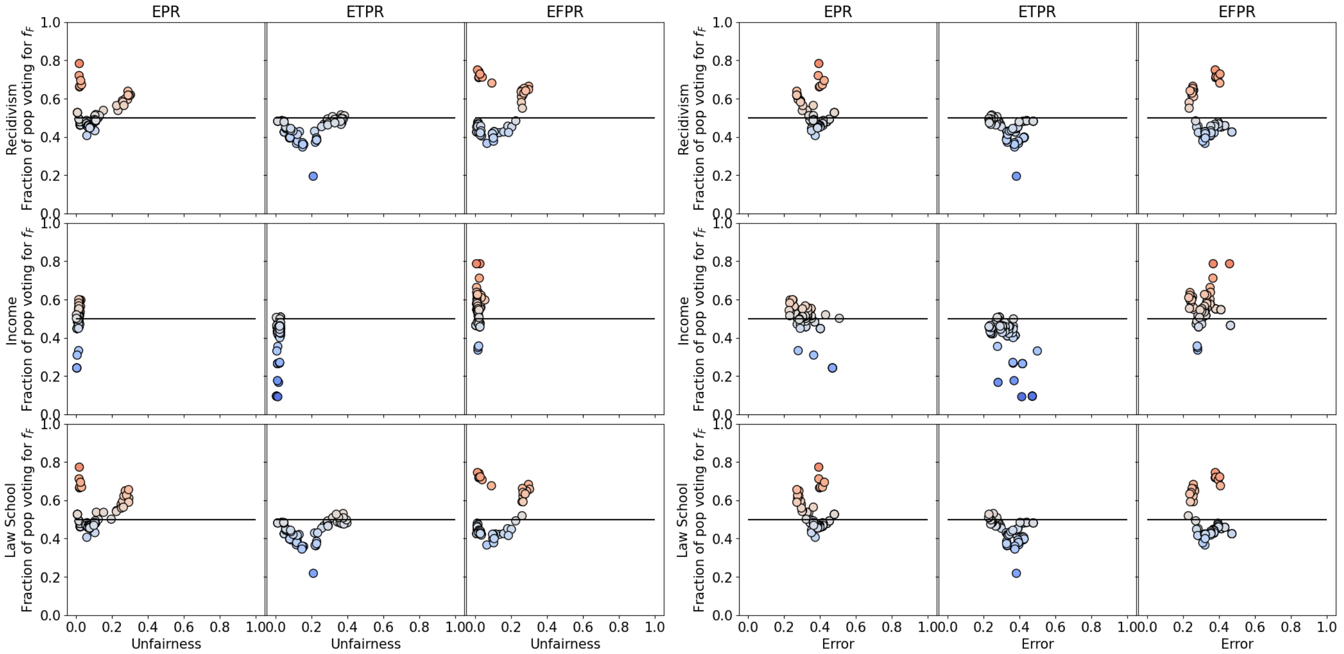

Figure 4: Percentage of the population voting for f_F (y-axis) as a function of classifier unfairness (x-axis left) and error (x-axis right) for classifiers where randomness stems from uncertainty over the training data. This uncertainty is modeled by downsampling 70% of the training 50 times. Each point in the graph reports a 3-fold average for each choice of hyperparameter and model type (20 choices of hyperparameters for Logistic Regression, Gradient Boosted Trees, and Support Vector Machines), where an expected outcome is the average number of times, out of 50, that they are are classified as a 1.

When formulating the integer programs, recall that the polynomial time algorithms achieve their run time by efficiently iterating over all possible sets of γn agents (out of a population of n). For each such group of γn the corresponding algorithm solves a program in which all γn agents prefer f_P (achieving unanimous preference over any subpopulation is tractable). Without popularity constraints, each program corresponding to DOS or k-QLS is polynomial time solvable. Rather than iterating over each potential set of constraints (for which there are γn possibilities) one could directly include the constraint,

$$\sum_{i=1}^{n} \mathbb{I}\big[f_P(\mathbf{x}_i) >= f_C(\mathbf{x}_i)\big] \geq \gamma n$$

Which is an integer linear constraint since $f_P(\mathbf{x}_i) \geq f_C(\mathbf{x}_i)$ is a binary variable which is linear in the decision variables of the program. By adding this constraint the corresponding integer program can be solved directly by modern solvers. In our experiments we solve the corresponding integer programs with CPLEX.

	Recidivism	Income	Crime	Law School
Deterministic DOS	0.001	0.001	0.001	0.001
Deterministic k-QLS	0.021	0.092	0.033	0.026
Randomized DOS	0.351	1.645	0.931	0.619
Randomized k-QLS	44.121	67.121	54.379	52.947

Table 1: Running time in seconds (rounded to three digits), of DOS and k-QLS for randomized models with $\gamma = 0.8$ and deterministic models with $\gamma = 0.95$. For deterministic models, the polynomial time algorithms are run, for randomized model a single integer program is run. Reported times are averaged across PR, TPR, and FPR fairness as well as all base model types (Logistic Regression, Gradient Boosted Trees, Support Vector Machine, and Neural Networks). Since DOS and k-QLS are postprocessing methods, reported running times do not include running time of the base models.

C.3 Performance of DOS and k-QLS

Figures 8, 9 show the accuracy and unfairness of the conventional model f_C, β-fair model f_F, and the γ-popular β-fair model f_P (learned via k-QLS) when model outcomes are deterministic. Figures 10, 10 show model AUC and unfairness of these model

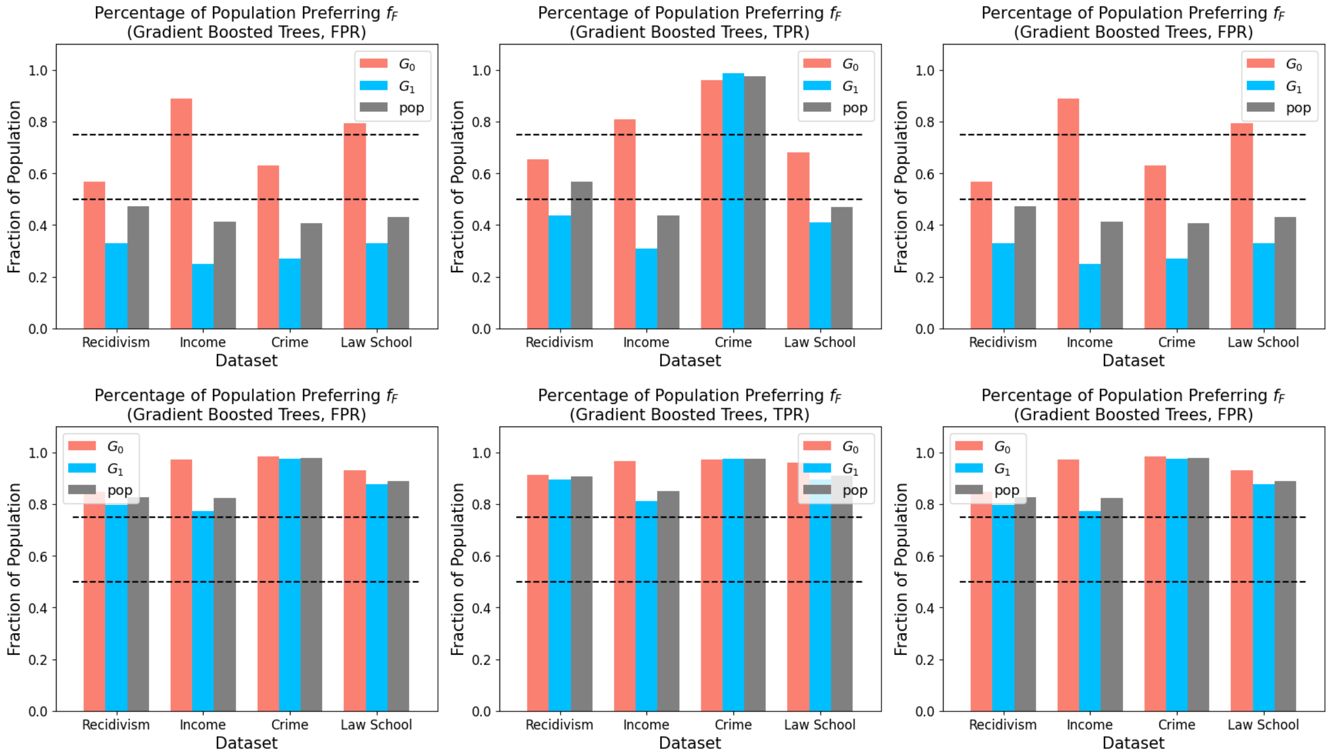

Figure 5: Fraction of each population or group voting for f_F over f_C for randomized classifiers (top) and deterministic classifiers (bottom), when f_F is learned via the Reductions algorithm and each classifier uses Gradient Boosted Trees.

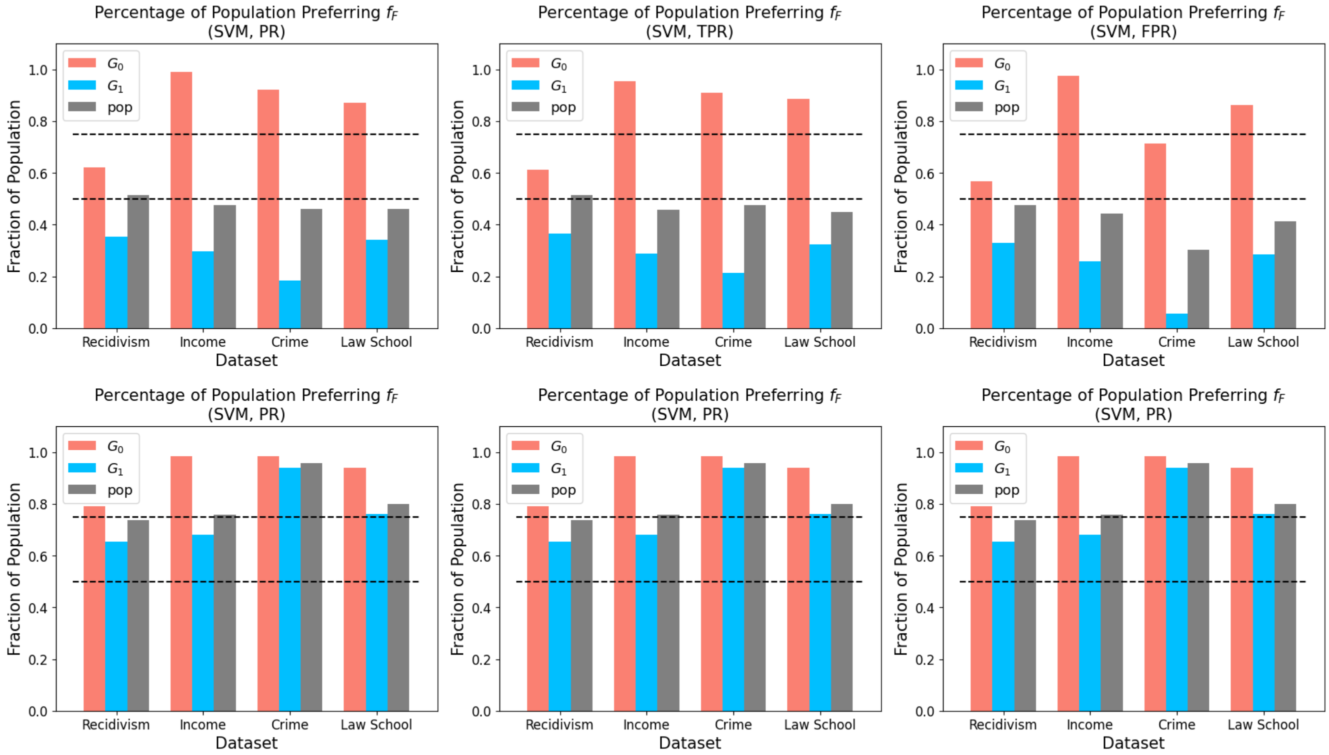

Figure 6: Fraction of each population or group voting for f_F over f_C for randomized classifiers (top) and deterministic classifiers (bottom), when f_F is learned via the Reductions algorithm and each classifier uses SVMs.

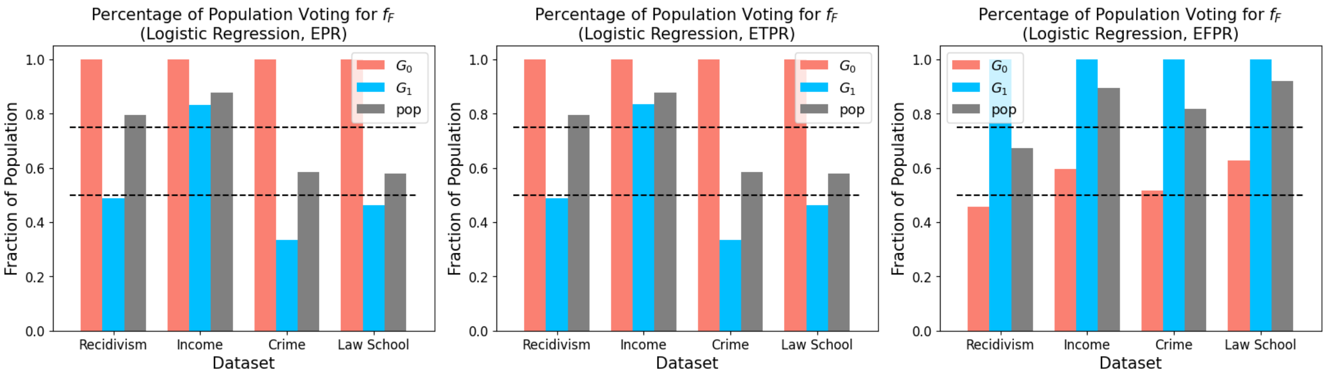

Figure 7: Fraction of each population or group voting for f_F over f_C for randomized classifiers (top) and deterministic classifiers (bottom), when f_F is learned via the EqOdds algorithm and each classifier uses Logistic Regression. Due to the way in which EqOdds achieves fairness, the entirety of one group will always prefer f_F, since $f_F = f_C$ on that group.

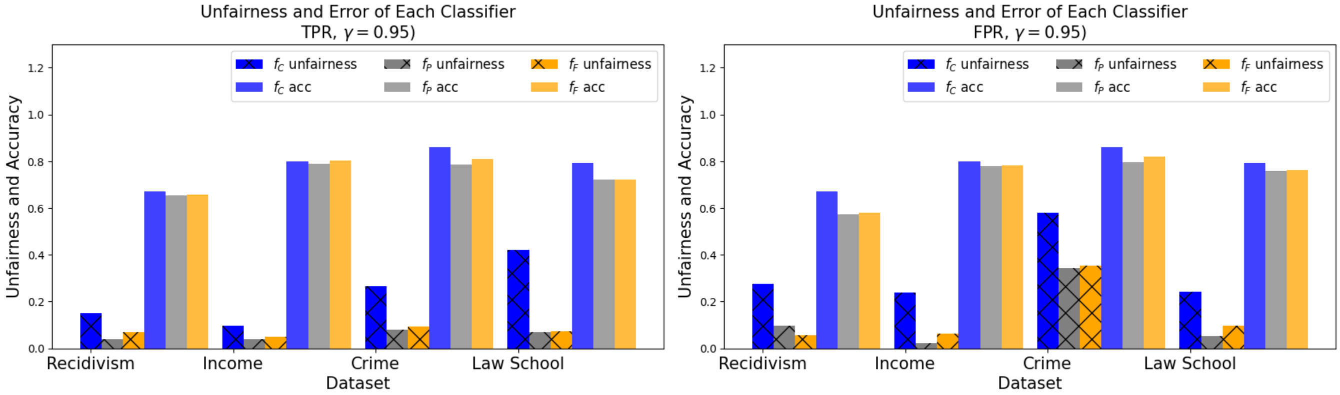

Figure 8: Model performance and unfairness on test data (3-fold average) for deterministic models with $\gamma = 0.95$. The conventional classifier f_C, fair classifier f_F learned via the reductions algorithm, and the fair popular classifier f_P learned via our postprocessing techniques k-QLS, each using Gradient Boosted Trees.

in the case of randomized classifiers. Figure 12, 13 show model AUC and unfairness when f_P is learned via the DOS algorithm in the case of randomized classifiers. Similar to the case of Logistic Regression, we see that f_P can achieve γ-popularity, and β-fairness, for relatively large levels of γ with minimal degradation to model performance.

Figures 16,17 show the increase in expected false positive rate (FPR) as the level of enforced popularity γ increases. Similar to the case of Logistic Regression we see that for larger γ that expected FPR is increases. The increase in expected FPR is due to the fact that a larger fraction of the population, namely γ fraction, must have $f_C(\mathbf{x}_i) \leq f_P(\mathbf{x}_i)$. Therefore, any false positives made by f_C on this γ fraction must persist (or increase) for f_P.

In Figure 14, we see that the in general, as the levels of unfairness decrease, and the levels of popularity increase, model performance declines across all three datasets. In some examples, such as the Law School dataset with an SVM classifier, model performance is far *less* affected by increased popularity, compared to stricter fairness. In contrast, we see that on the Crime dataset with a GBT classifier, model performance is far *more* affected by increased popularity, compared to stricter fairness. However, we observe in general, that there is a fundamental trade-off between the three values, implying that it may be challenging to produce models which have high levels of performance, fairness, and popularity.

As demonstrated previously, a primary contributing factor to low model performance, for higher levels of popularity, is the preservation of false positive errors. Instances in which higher levels of popularity correspond to lower model performance, typically coincide with baseline classifiers which have larger rates of false positive errors.

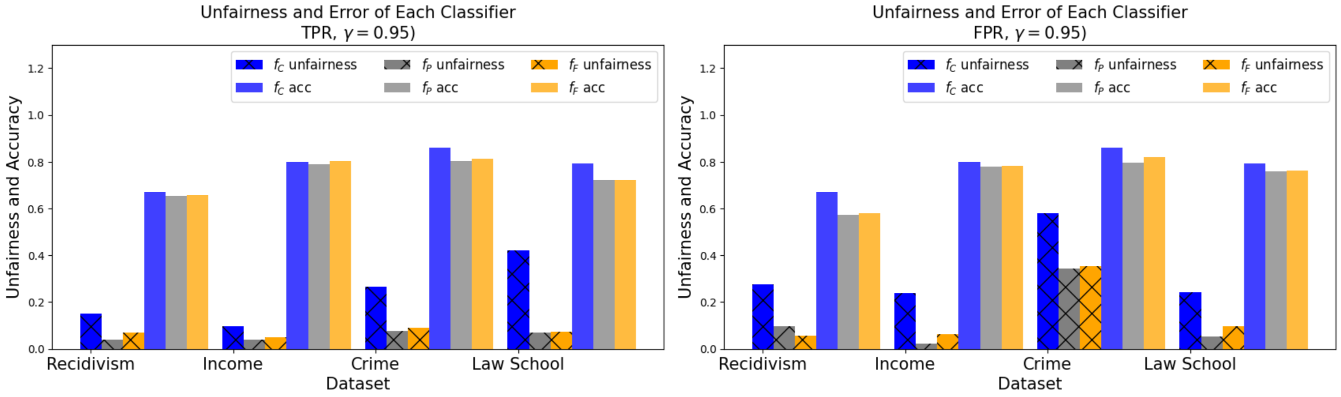

Figure 9: Model performance and unfairness on test data (3-fold average) for deterministic models with $\gamma = 0.95$. The conventional classifier f_C, fair classifier f_F learned via the reductions algorithm, and the fair popular classifier f_P learned via our postprocessing techniques k-QLS, each using Support Vector Machines.

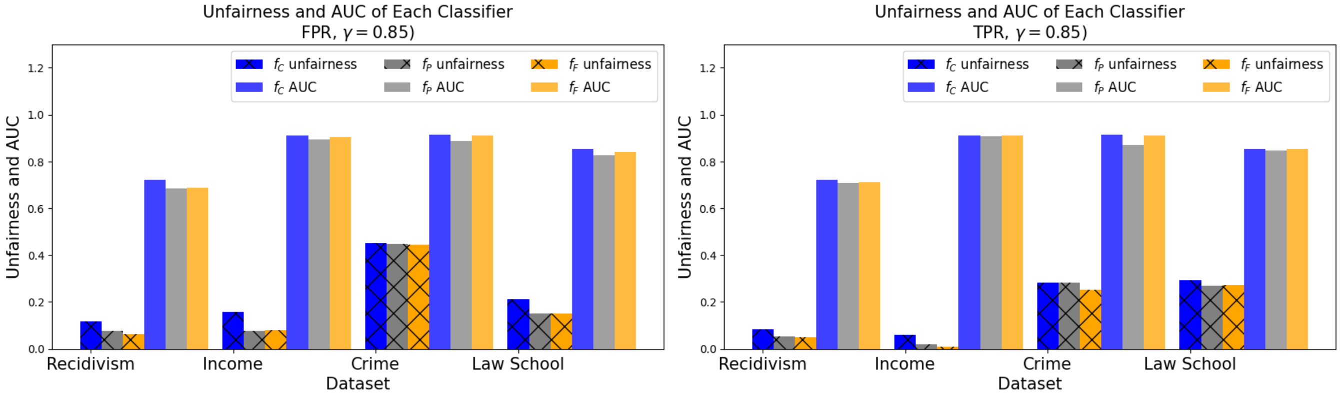

Figure 10: Model performance and unfairness on test data (3-fold average) for randomized models with $\gamma = 0.85$. The conventional classifier f_C, fair classifier f_F learned via the reductions algorithm, and the fair popular classifier f_P learned via our postprocessing techniques k-QLS, each using Gradient Boosted Trees.

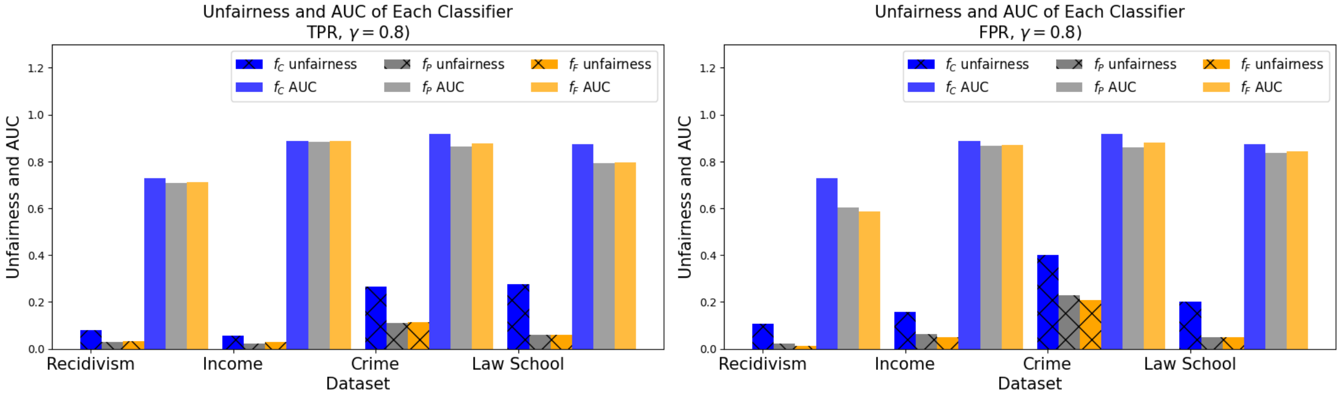

Figure 11: Model performance and unfairness on test data (3-fold average) for randomized models with $\gamma = 0.8$. The conventional classifier f_C, fair classifier f_F learned via the KDE algorithm, and the fair popular classifier f_P learned via our postprocessing techniques k-QLS, each using Support Vector Machines.

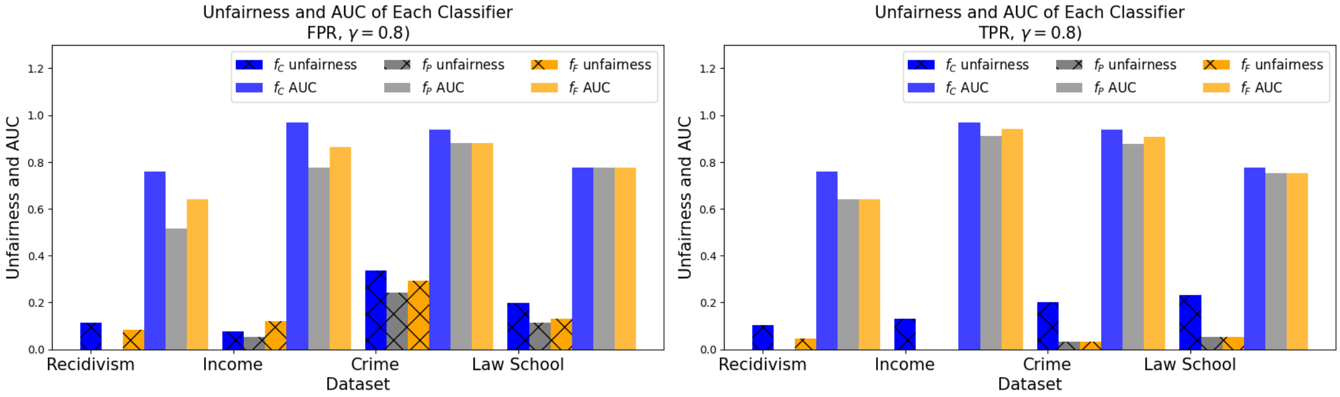

Figure 12: Model performance and unfairness on test data (3-fold average) for randomized models with $\gamma = 0.8$. The conventional classifier f_C, fair classifier f_F learned via the reductions algorithm, and the fair popular classifier f_P learned via our postprocessing techniques DOS, each using Gradient Boosted Trees.

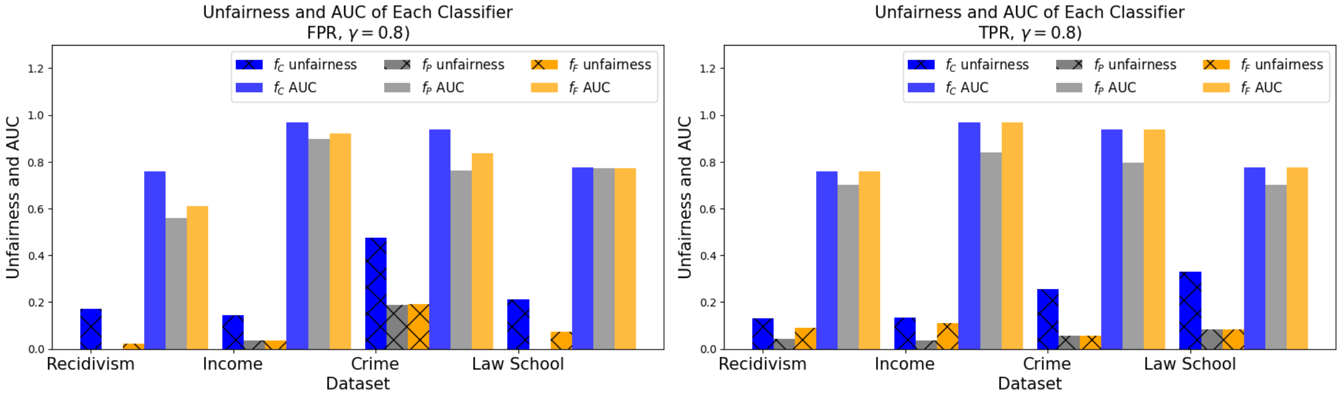

Figure 13: Model performance and unfairness on test data (3-fold average) for randomized models with $\gamma = 0.8$. The conventional classifier f_C, fair classifier f_F learned via the KDE algorithm, and the fair popular classifier f_P learned via our postprocessing techniques DOS, each using Support Vector Machines.

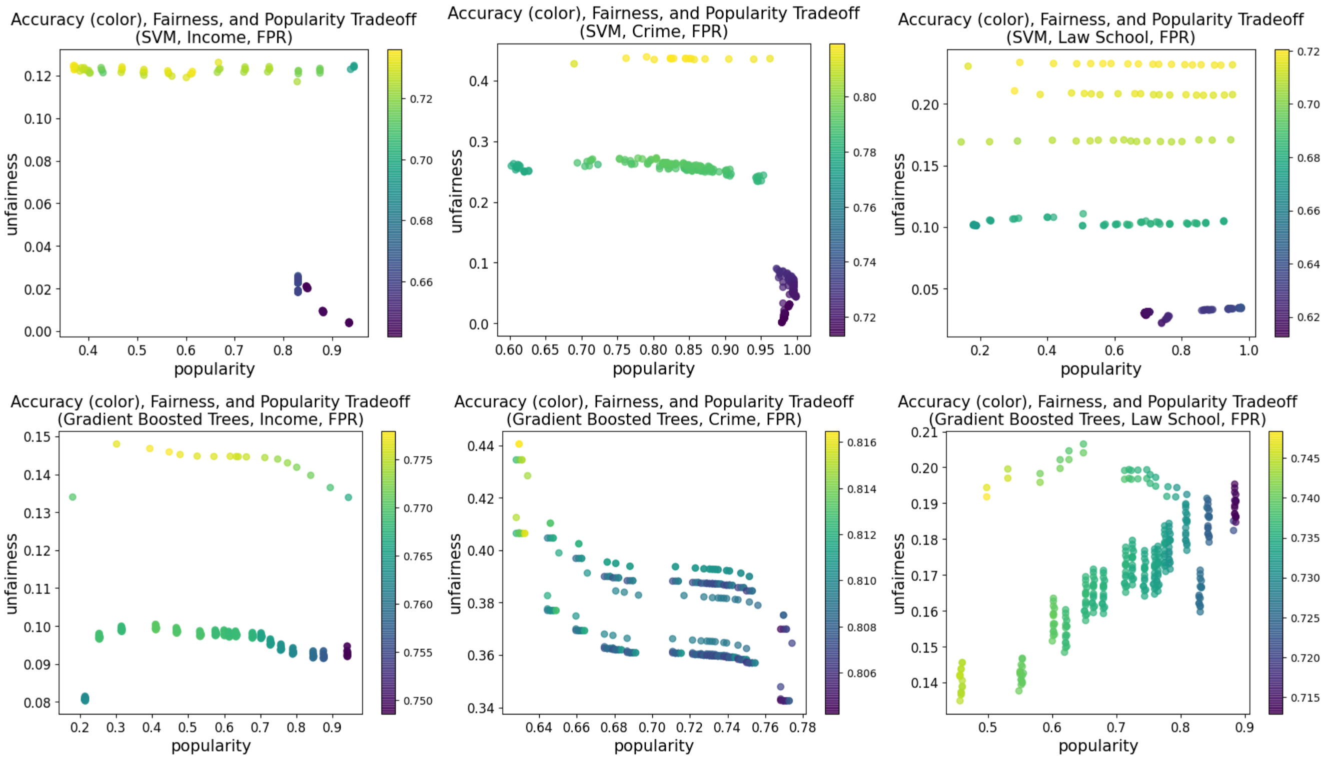

Figure 14: Frontier of fairness, popularity, and expected accuracy for randomized classifiers with our k-QLS postprocessing technique, using SVM (top) and Gradient Boosted Trees (bottom) for FPR fairness. Fair classifiers are trained using the Reductions algorithm.

C.4 Group Preference Over Fairness

A common method for learning β-fair classifiers is the so called Lagrangian penalty method with Lagrangian multiplier $\lambda \in \mathbb{R}$, i.e.,

$$f_F = \arg\min_{f \in \mathcal{H}} \mathcal{L}(f, X, Y) + \lambda\big(U(f, \mathbf{X}, Y, G) - \beta\big)$$

Here λ gives the relative "importance" of fairness. When $\lambda = 0$ the objective of the conventional classifier is recovered. To understand the preference of agents over fair and conventional models, we can also look at the the relative preference for fairness among each group. In particular, suppose that each agent prefers the value of λ yielding the highest expected outcome. Then the average of these preferred λ across each group gives the groups relative preference for fairness: higher values of λ corresponds to a stronger preference for fairness.

In Figure 18 we see the average preferred λ of each group and as well as the total population. These choices of λ then lead to the corresponding levels of unfairness in Figure 19. In each combination of hypothesis class and dataset, the advantaged group G_1 votes for smaller λ while the disadvantaged group G_0 votes for larger λ (larger than the principals choice in-fact). From this Figure we see that the advantaged group G_1 (advantaged in terms of either PR, FPR, or TPR) prefers smaller values of λ, while the disadvantaged group G_0, prefers larger values of λ. Mover due to the relative size of G_1, the total population on average also prefers smaller values of λ. Thus there is somewhat of a fundamental trade-off between fairness and preferred level of fairness.

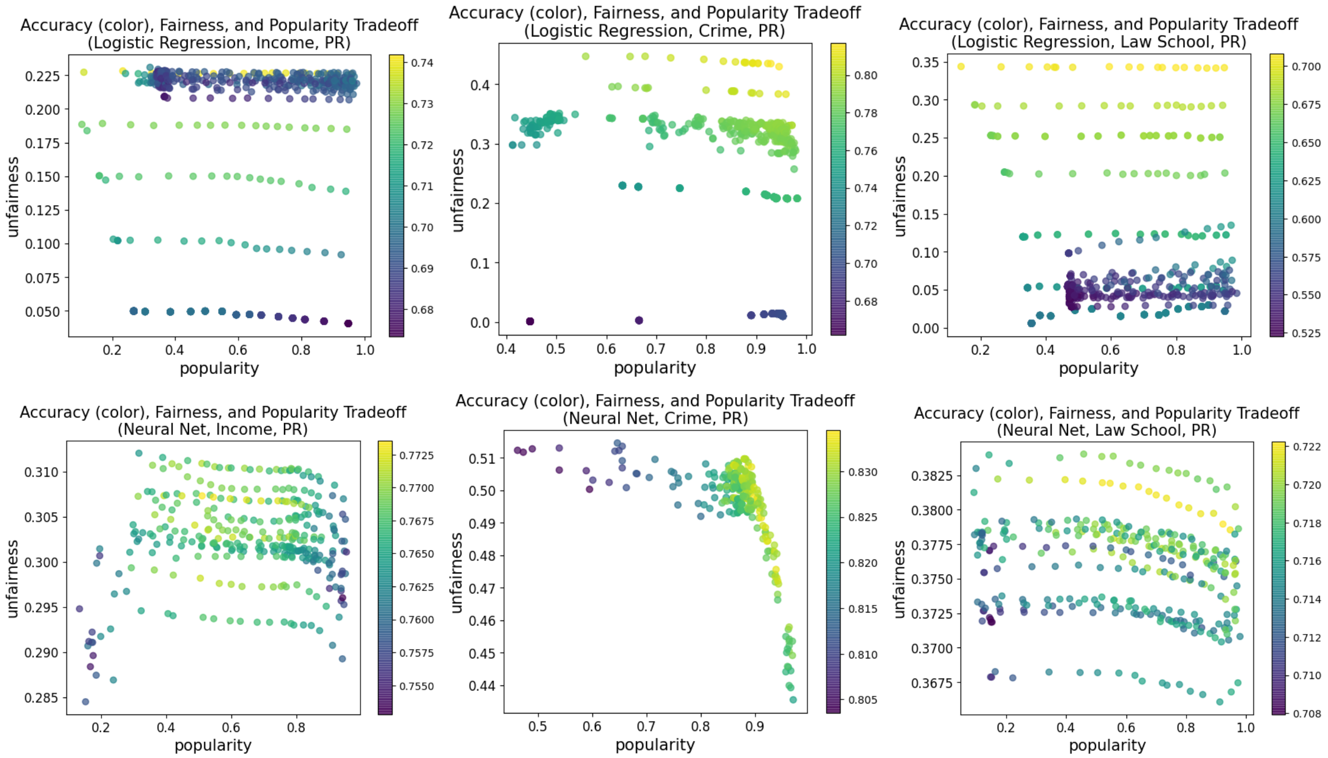

Figure 15: Frontier of fairness, popularity, and expected accuracy for randomized classifiers with our k-QLS postprocessing technique, using Logistic Regression (top) and Neural Networks (bottom) for FPR fairness. Fair classifiers are trained Using the KDE algorithm.

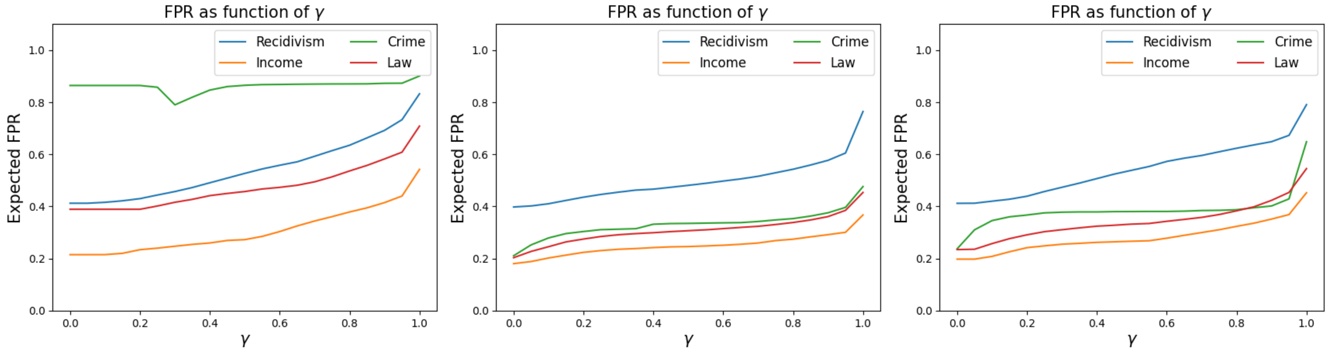

Figure 16: Expected False Positive Rate (FPR) of k-QLS, on randomized classifiers for PR-fairness (left) TPR-fairness (center) and FPR-fairness (right),as a function of γ (Gradient Boosted Trees).

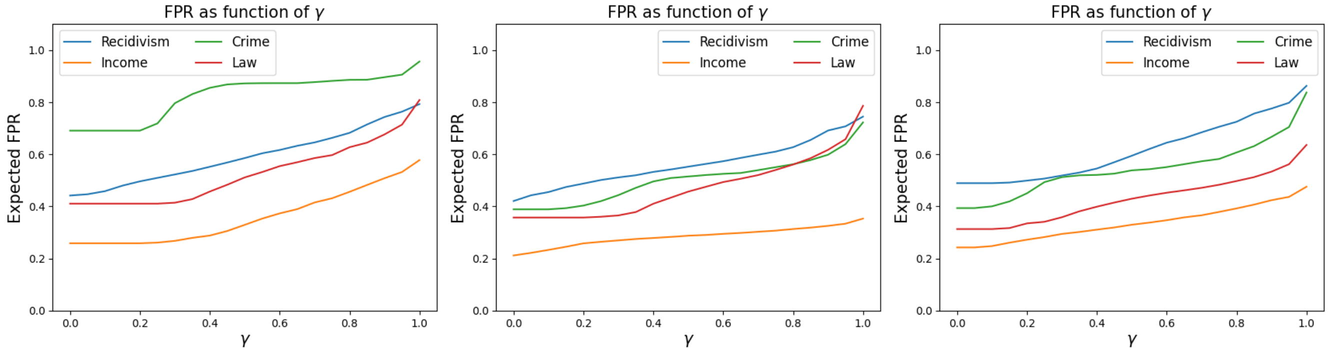

Figure 17: Expected False Positive Rate (FPR) of k-QLS, on randomized classifiers for PR-fairness (left) TPR-fairness (center) and FPR-fairness (right),as a function of γ (Support Vector Machines).

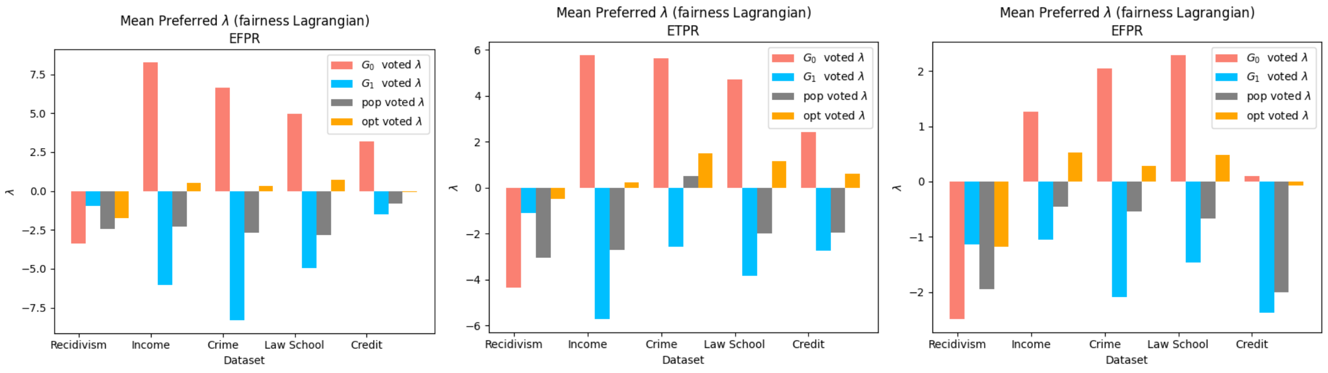

Figure 18: Average preferred λ of the population, or each group, for three choices of classifiers, Logistic Regression (left), Gradient Boosted Trees, (center), and SVMs (right), for each of the 5 datasets.

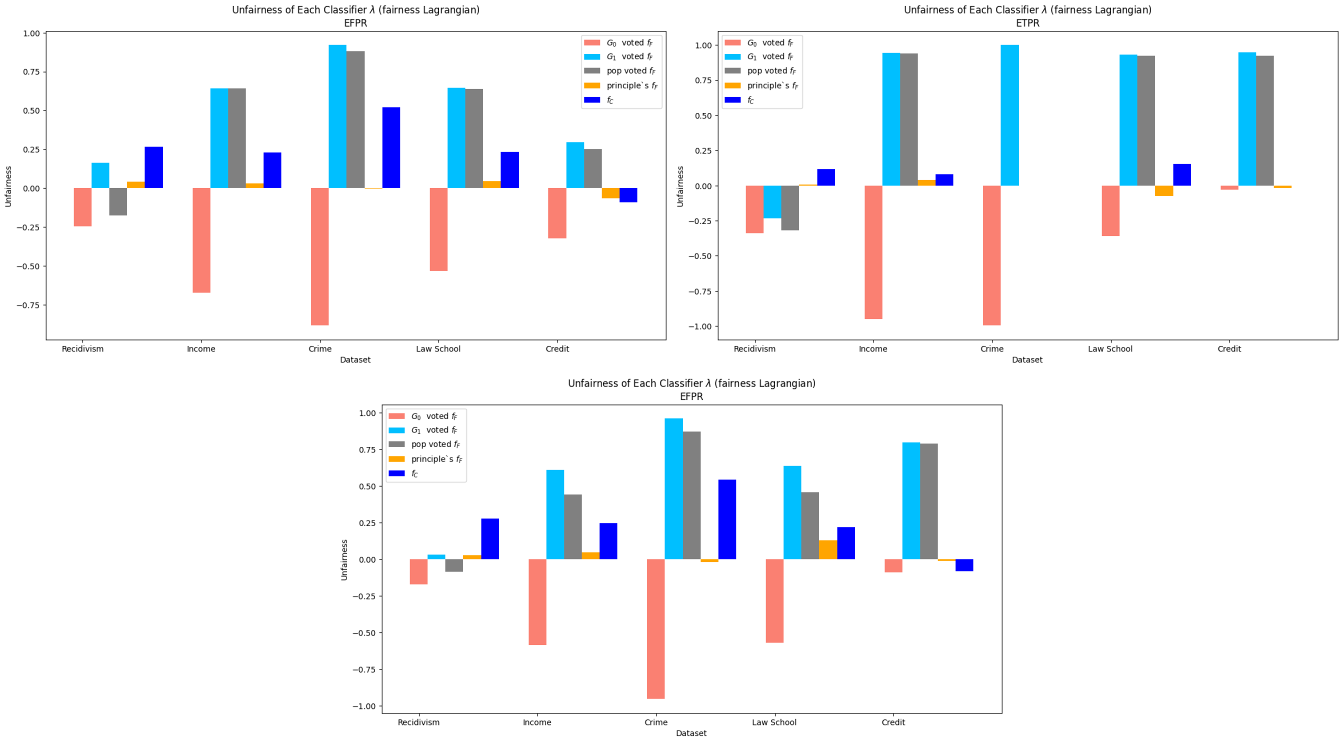

Figure 19: The unfairness of each preferred λ. (corresponding to those in Figure 18