Balanced Filtering via Disclosure-Controlled Proxies

Siqi Deng[1]
Amazon AWS AI, Palo Alto, CA, USA

Emily Diana
Toyota Technological Institute at Chicago, IL, USA

Michael Kearns
University of Pennsylvania, Philadelphia, PA, USA
Amazon AWS AI, Palo Alto, CA, USA

Aaron Roth
University of Pennsylvania, Philadelphia, PA, USA
Amazon AWS AI, Palo Alto, CA, USA

Abstract

We study the problem of collecting a cohort or set that is *balanced* with respect to sensitive groups when group membership is unavailable or prohibited from use at deployment time. Specifically, our deployment-time collection mechanism does not reveal significantly more about the group membership of any individual sample than can be ascertained from base rates alone. To do this, we study a learner that can use a small set of labeled data to train a proxy function that can later be used for this filtering or selection task. We then associate the range of the proxy function with sampling probabilities; given a new example, we classify it using our proxy function and then select it with probability corresponding to its proxy classification. Importantly, we require that the proxy classification does not reveal significantly more information about the sensitive group membership of any individual example compared to population base rates alone (i.e., the level of disclosure should be controlled) and show that we can find such a proxy in a sample- and oracle-efficient manner. Finally, we experimentally evaluate our algorithm and analyze its generalization properties.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Security and privacy → Human and societal aspects of security and privacy

Keywords and phrases Algorithms, Sampling, Ethical/Societal Implications

Digital Object Identifier 10.4230/LIPIcs.FORC.2024.4

Related Version *Full Version*: https://arxiv.org/abs/2306.15083

1 Introduction

There are a variety of situations in which we would like to select a cohort or set that is *balanced* or *representative* (having an approximately equal number of samples from different groups) with respect to race, sex, or other sensitive attributes – but, we cannot explicitly select based on these attributes. This could be because the attributes are sensitive so were never collected, they could be redacted from the information we see, they could be too resource intensive to collect, or selecting based on these attributes could be illegal.

Consider the context of college admissions. Out of many qualified applicants, a college may prioritize racial diversity when deciding upon the final cohort to admit. However, in the United States Supreme Court decision for *Students for Fair Admissions, Inc. v. President and Fellows of Harvard College*, it was determined that "Harvard's and UNC's

[1] Corresponding author

[race-conscious] admissions programs violate the Equal Protection Clause of the Fourteenth Amendment" [36]. How might a college select a racially diverse cohort with affirmative action prohibited?

Our approach is based on training a proxy classifier – in the form of a decision-tree – with the following properties: (1) The set of points classified at each leaf should not be strongly correlated with the protected attribute (the *disclosure-control* part) and (2) The set of distributions on the protected attributes induced at each leaf should be such that the uniform distribution on protected attributes is in their convex hull (the *balancing* part). The second condition allows us to assign sampling probabilities to the leaves such that if we accept each example with probability corresponding to its proxy classification, in expectation the selected cohort will be balanced with respect to the protected attribute.[2]

1.1 Related Work

The proxy problem is a subject of ongoing debate in the philosophy of science and causal inference literatures (e.g. [18, 3, 32, 24, 41, 29, 31, 9]), and our work engages with this literature methodologically – we do not believe that it is our role to take a philosophical or legal stance but rather to broaden the set of available tools. Using proxy variables for sensitive attributes in settings where diversity or equity is a concern has been standard practice, yet in many cases, existing features are chosen for the proxies (such as surname, first name, or geographic location [13, 40, 44]). Rather than using an existing feature as a proxy, we propose deliberately *constructing* a proxy. Several works take this perspective – in [10], for example, the authors produce a proxy that can be used during training to build a fair model downstream. But often, proxies for protected attributes are explicitly intended to be good predictors for those attributes; it is not clear that using an accurate "race predictor" is an acceptable solution to making decisions in which race should not be used (and is often explicitly prohibited). Our primary point of departure is that we train a model to make classifications that are *minimally correlated* with the protected attribute.

While our intended use cases are primarily curation or cohort selection, one may also use our method for collecting balanced data sets for machine learning applications. However, we recommend caution in these scenarios, as our approach does not give guarantees about the level of distortion of the final filtered data set. In order to provide comparisons to existing empirical techniques, however, we do measure our approach against a common data pre-processing technique, SMOTE (Synthetic Minority Oversampling Technique) [8]. Other re-sampling methods for data balancing include ADASYN [17], MIXUP [43], SMOTE adaptations [26, 4, 5, 11, 15, 23]) and cluster-based approaches that under-sample disproportionately represented classes [16, 42, 33, 21, 34]. In the causal literature, propensity score re-weighting [22] is also a popular approach to account for group size differences. Each of these techniques, however, requires access to the sensitive attribute. Our approach's primary point of departure is that we do not use the sensitive attribute – or a direct prediction or imputation of it – at the final collection time when we are deploying our method.

[2] A natural first approach is to add noise to the predictor for the protected attribute, against which we compare. However, we are motivated by the need to have strategies that never involve training a classifier for the protected attribute, especially if it could be used outside of the intended system.

1.2 Limitations and Discussion

Our contributions are twofold. First, for situations where balance is desired but disclosure is not a concern, we introduce a sampling scheme optimized to collect a balanced cohort. Second, for when disclosure is a concern, we present a method to produce a proxy function for the balanced selection task that is *guaranteed* not to be too disclosive. Below, we discuss important considerations having to do with appropriate usage of our methodology, limitations, and areas for expansion.

- **The sensitive attribute is still used to inform the proxy, and our approach relies on accessing a small sample of data with this attribute:** The proxy training algorithm we propose is not blind to the sensitive attributes, which it must access during *training*. Rather, the proxy does not use these attributes at the time of *deployment*.[3] We emphasize that there is no contradiction between (1) being able to obtain (once) a small data set labeled with sensitive attributes and then using it to train a classification algorithm (in this case, our proxy model) and (2) having the inability to collect or use sensitive attributes when gathering the bulk of one's data. This is especially true when the final selection criterion is not closely correlated with the sensitive attribute, which is one of our primary objectives. In the algorithmic fairness literature in particular, there is a substantial and growing body of work on learning classifiers that satisfy fairness constraints by sensitive attributes but that do not use these attributes at test time (e.g. [1, 20, 27, 28]). *These methods still require access to the attributes at training time.*
  The distinction between using sensitive attributes at train versus test time is essential. In certain financial applications in the United States, using race or gender at test time (i.e., when making lending decisions) is illegal. But it is not illegal to use these attributes at training time to audit models for statistical bias and to remove it if found. The distinction in our case is similar: We use these attributes to find a statistical selection criterion but do not use sensitive attributes of individuals to make selection decisions about them.
- **The filtered cohort or data set will likely exhibit within-group distortion:** This is an important consideration that should be taken into account when using our method. Our theoretical guarantees provide bounds on the level of balance and disclosure when measured with respect to the sensitive attributes, but they do not guarantee that the distribution over selected individuals matches that of the true population. In fact, this is perhaps a necessary effect of our process and in many cases may be natural. For example, in the context of college admissions or interview selection, a university or firm is intentionally selecting a pool that is *not* representative of the base population. The use cases for which this quality may create the greatest challenge is in curating data sets for training machine learning models. While our method can improve *representation* in data sets, it will not necessarily lead to improvements in downstream fairness of models trained on the balanced data. We provide a detailed analysis of this in Appendix B.[4]
- **Affirmative Action and Legal Challenges:** We do not propose our method as a way to circumvent the intent of legislation, nor do we make claims regarding the legal or moral appropriateness of its usage in any particular affirmative action setting. Rather, we view it as a tool that can be used, when permitted legally, in settings where diversity or

[3] It would be impossible to give an algorithm making no use of the protected attribute during deployment or training and yet promising any sort of balance – it would have to behave identically on any distributions with the same marginals over non-protected attributes, even if they differed on the protected attribute.

[4] One note, however, is that our method does allow for balancing multiple attributes at a time – therefore, one could ask for a cohort that has the same number of positive and negative examples in each group.

balance is desired but when the sensitive attribute can or should be used only minimally. For example, in the college admissions example, it is also undesirable to use explicit race-based predictors in lieu of observing the sensitive attribute. However, students can include race considerations in their admissions essays, which admissions officers see. Given the stakes of college admissions and the strategic behavior of both sides (applicants and schools), it is very likely that an ad-hoc system will still develop to indirectly make use of racial information for the sake of diversity. In a situation such as this, our method provides a controlled way to achieve such desiderata without unintentionally revealing too much information or making use of highly correlated proxies.

2 Model and Preliminaries

Let $\Omega = \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$ be an arbitrary data domain and $\mathcal{P}$ be the probability distribution over Ω. $\mathcal{P}_x$ will refer to the marginal distribution over $\mathcal{X}$, $\mathcal{P}_z$ will refer to the marginal distribution over $\mathcal{Z}$, and $\mathcal{P}_{z|x}$ will refer to the conditional distribution over $\mathcal{Z}|\mathcal{X}$. Each data point is a triplet $\omega = (x, y, z) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Z}$, where $x \in \mathcal{X}$ is the non-sensitive feature vector and $y \in \mathcal{Y} = \{0, 1\}$ is the label. The label y is not required in training or applying our filtering method, but it will be used in the analysis of downstream fairness effects of the filtering process provided in Appendix B. In the paper body, therefore, we omit it for clarity.

Unless otherwise specified, we take group membership as disjoint such that $z \in \mathcal{Z} = [K]$ is an integer indicating sensitive group membership, but our framework can easily be extended to the case where group membership need not be disjoint. We consider the uniform distribution, U, to be our target distribution over sensitive attributes, where $U = (\frac{1}{K}, ..., \frac{1}{K})$. We also provide a brief extension to the intersecting case below.

We imagine we can sample unlimited data from $\mathcal{P}_x$, but the corresponding value z can only be obtained from self-report, authorized agencies, or human annotation. We use r_k to denote the base rate $\Pr_{z\sim\mathcal{P}_z}[z = k]$ in the underlying population distribution. We also assume that we can obtain a limited sample of data D of n samples $\{(x_i, z_i)\}_{i=1}^n \subset \Omega$ for which we can observe the true sensitive attribute z. We would like to use this sample to collect a much larger set $S \subset \Omega$ such that even if we cannot observe the sensitive attributes, S is balanced with respect to z.

Formally, we define a balanced set as follows, where $\Pr_{z\sim S}[z = k] = \frac{1}{|S|}\sum_{i=1}^{|S|} \mathbb{1}_{z_i=k}$ is the empirical distribution of z drawn uniformly from S.

▶ **Definition 1** (Balance). *A set S is balanced with respect to K disjoint groups $z \in [K]$ if* $\Pr_{z\sim S}[z = k] = \frac{1}{K}\ \forall k$.

Due to finite sampling, Definition 1 will rarely be met, even if the underlying *distribution* is uniform over sensitive attributes. Therefore, we also discuss approximate balance:[5]

▶ **Definition 2** (β-Approximate Balance). *A set S is β-approximately balanced with respect to K disjoint groups $z \in [K]$ if* $\|(\Pr_{z\sim S}[z = 1], ..., \Pr_{z\sim S}[z = K]) - (1/K, .., 1/K)\|_2 \leq \beta$.

Note that β-approximate balance involves the *distribution* of sensitive groups in S: as this distribution deviates farther from the uniform, the imbalance, β, increases.

In the intersecting groups case, we let the sensitive attribute domain $\mathcal{Z}$ be a binary vector of length K, such that $z \in \mathcal{Z} = \{0, 1\}^K$. We will assume $\mathcal{Z}$ is composed of G group classes $\{Z_i\}_{i=1}^G$ (e.g., sex, race, etc), and each group class Z_i has K_i groups. Thus, the vector z will

[5] We use the L_2 norm due to operational reasons, as it allows us to make useful geometric arguments.

indicate all possible group memberships, where the length of the vector of group memberships is $K = \sum_{i=1}^{G} K_i$. We also replace U with $U_{\text{int}} = (\frac{1}{K_1}, ..., \frac{1}{K_1}, ..., \frac{1}{K_G}, ..., \frac{1}{K_G})$ to indicate the target distribution over intersecting sensitive attributes.

▶ **Definition 3** (Multi-Class Balance). *We will say that a set S is balanced with respect to K intersecting groups if, for any group class $\{Z_i\}_{i=1}^{G}$ composed of groups $\{Z_{i_j}\}_{j=1}^{K_i}$, $\Pr_{z\sim S}[z[Z_{i_j}] = 1] = \frac{1}{K_i}$ $\forall j$.*

▶ **Definition 4** (β-Approximate Multi-Class Balance). *We say that a set S is β-approximately balanced with respect to K intersecting groups if*

$$\|(\Pr_{z\sim S}[z[Z_{1_1}] = 1], \Pr_{z\sim S}[z[Z_{1_2}] = 1], ..., \Pr_{z\sim S}[z[Z_{G_{K_G-1}}] = 1], \Pr_{z\sim S}[z[Z_{G_{K_G}}] = 1]) - U_{int}\|_2 \leq \beta$$

▶ Remark 5. This definition aims to take into account the fact that for different group classes, there may be a different number of potential groups. For example, there may only be two sex groups but eight income groups. Asking that the representation of each of those categories be one-tenth of the final sample would not make sense. However, our definition does not prevent certain intersections being more represented than others. When the number of groups is small, this can always be dealt with by using the Cartesian product over group classes to enumerate intersectional groups.

In addition to desiring that our proxy allows us to select an approximately *balanced* cohort, we would also like the classification outcomes of the proxy not to be overly disclosive. We model this by asking that the posterior distribution on group membership is close to the prior distribution when conditioning on the outcome of the proxy classifier.

▶ **Definition 6** (α-Disclosive Proxy). *A proxy g is α-disclosive (or has disclosure level at most α) on set S if, for all sensitive groups k and proxy values i, $|\Pr_{z|x\sim S}[z = k|g(x) = i] - \Pr_{z\sim S}[z = k]| \leq \alpha$.*

For any proxy function g, we can analyze the distribution of sensitive groups amongst points mapped to each value k in the range of the proxy. Call this conditional distribution a_k, let l be the number of unique elements in the range of the proxy, and let A be the $\ell \times K$ matrix whose k^{th} row is a_k. Denote the convex hull, defined in Definition 8, of the rows of A by $C(A)$. Then, we can add a notion of *balance* into our proxy definition in the following way:

▶ **Definition 7** ((α, β) Proxy). *$g : \mathcal{X} \to \mathbb{N}$ is an (α, β) proxy if it is α-disclosive and $\inf_{U'\in C(A)} \|U' - U\|_2 \leq \beta$.*

Here, $\inf_{U'\in C(A)} \|U' - U\|_2$ indicates the Euclidean distance between U and the closest point in $C(A)$. We will slightly abuse notation and refer to this as the distance $\|C(A) - U\|_2$. The disclosure parameter α controls the amount of additional information the proxy gives about group membership, while the balance parameter β quantifies the minimum distance from uniform achievable with any acceptance probabilities for a given proxy. There do exist limitations on how small α can be if we need full balance. With K sensitive groups, the final frequency of each group must be $\frac{1}{K}$ if we desire $\beta = 0$: if there is a group with initial frequency f, there is no avoiding that $\alpha \geq |f - \frac{1}{K}|$. For some data sets, this unavoidably can be quite large. For example, consider a data set of $\frac{1}{3}$ men and $\frac{2}{3}$ women and assume the proxy g takes values 0 or 1. Of the samples mapped to $g = 0$, $\frac{1}{4}$ are men and $\frac{3}{4}$ are women. Of those mapped to $g = 1$, $\frac{1}{2}$ are men and $\frac{1}{2}$ are women. Then, $\alpha = \frac{1}{6}$, because $|\Pr[z = \text{men}|g = 0] - \Pr[z = \text{men}]| = |\frac{1}{3} - \frac{1}{2}| = \frac{1}{6}$. In this example, β would be 0, because the convex hull of the conditionals $\Pr[z|g(x)]$ contains $[\frac{1}{2}, \frac{1}{2}]$. Next, we provide definitions for a convex hull and stochastic vector.

▶ **Definition 8** (Convex Hull [7]). *The convex hull of a set of points S in K dimensions is the intersection of all convex sets containing S. For l points $s_1, ..., s_l$, the convex hull C is given by the expression: $C \equiv \{\sum_{i=1}^{l} q_i s_i : q_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^{l} q_i = 1\}$*

▶ **Definition 9** (Stochastic Vector [6]). *$v = (v_i)_{i=1}^{\ell}$ is a stochastic vector if $\sum_{i=1}^{\ell} v_i = 1$ and $v_i \geq 0 \; \forall i$.*

Finally, we observe a necessary and sufficient condition for U to be in $C(A)$.

▶ **Lemma 10** (Inclusion in Convex Hull). *Let A be an $l \times K$ matrix and U be a $1 \times K$ vector with $\frac{1}{K}$ in each entry. There exists a stochastic vector q such that $qA = U$ if and only if $U \in C(A)$.*

Proof. Let a_i denote the i^{th} row of A. If U is in $C(A)$, then by Definition 8, there exists a non-negative vector q such that $\sum_{i=1}^{l} q_i a_i = U$ and $\sum_{i=1}^{l} q_i = 1$. Similarly, for any stochastic q, $qA \in C(A)$. Then if $qA = U$, $U \in C(A)$. ◀

Finally, we outline several key results that we will use in the derivations and proofs for our proxy training algorithm. We begin by considering a zero-sum game between two players, a Learner with strategies in S_1 and an Auditor with strategies in S_2. The payoff function of the game is $W : S_1 \times S_2 \to \mathbb{R}_{\geq 0}$.

▶ **Definition 11** (Approximate Equilibrium [14]). *A pair of strategies $(s_1, s_2) \in S_1 \times S_2$ is said to be a ν-approximate minimax equilibrium of the game if the following conditions hold: $U(s_1, s_2) - \min_{s_1' \in S_1} U(s_1', s_2) \leq \nu, \quad \max_{s_2' \in S_2} U(s_1, s_2') - U(s_1, s_2) \leq \nu$*

Freund and Schapire [14] show that if a sequence of actions for the players jointly has low *regret*, the uniform distribution over each player's actions forms an approximate equilibrium:

▶ **Theorem 1** (No-Regret Dynamics [14]). *Let S_1 and S_2 be convex, and suppose $W(\cdot, s_2) : S_1 \to \mathbb{R}_{\geq 0}$ is convex for all $s_2 \in S_2$ and $W(s_1, \cdot) : S_2 \to \mathbb{R}_{\geq 0}$ is concave for all $s_1 \in S_1$. Let $(s_1^1, s_1^2, \ldots, s_1^T)$ and $(s_2^1, s_2^2, \ldots, s_2^T)$ be sequences of actions for each player. If for $\nu_1, \nu_2 \geq 0$, the regret of the players jointly satisfies*

$$\sum_{t=1}^{T} W(s_1^t, s_2^t) - \min_{s_1 \in S_1} \sum_{t=1}^{T} W(s_1, s_2^t) \leq \nu_1 T \quad \max_{s_2 \in S_2} \sum_{t=1}^{T} W(s_1^t, s_2) - \sum_{t=1}^{T} W(s_1^t, s_2^t) \leq \nu_2 T$$

then the pair $(\bar{s}_1, \bar{s}_2)$ is a $(\nu_1 + \nu_2)$-approximate equilibrium, where $\bar{s}_1 = \frac{1}{T} \sum_{t=1}^{T} s_1^t \in S_1$ and $\bar{s}_2 = \frac{1}{T} \sum_{t=1}^{T} s_2^t \in S_2$ are the uniform distributions over the action sequences.

Additionally, we define a Cost Sensitive Classification (CSC) oracle over a classification model class $\mathcal{H}$, which we will use as an efficient subroutine in our algorithm.

▶ **Definition 12** (Weighted Cost-Sensitive Classification Oracle for $\mathcal{H}$ [2]). *An instance of a Weighted Cost-Sensitive Classification problem, or a CSC problem, for the class $\mathcal{H}$, is given by a set of n tuples $\{w(x_i), x_i, c_i^0, c_i^1\}_{i=1}^{n}$ such that c_i^1 corresponds to the cost for predicting label 1 on sample x_i and c_i^0 corresponds to the cost for prediction label 0 on sample x_i. The weight of x_i is denoted by $w(x_i)$. Given such an instance as input, a $CSC(\mathcal{H})$ oracle finds a hypothesis $h \in \mathcal{H}$ that minimizes the total cost across all points: $h \in \operatorname{argmin}_{h' \in \mathcal{H}} \sum_{i=1}^{n} w(x_i) \left[h'(x_i) c_i^1 + (1 - h'(x_i)) c_i^0 \right]$.*

3 Computing Sampling Weights from a Proxy (QP Approach)

Now we introduce our first methodological contribution: a selection approach for producing a balanced set *given* a proxy. At a high level, our approach involves mapping each example to an acceptance probability. We construct such a mapping by labeling the range of the proxy $g : \mathcal{X} \to \mathbb{N}$ with acceptance probabilities and then selecting samples for our set by applying the proxy function to a sample and keeping it with probability corresponding to the element of the range of the proxy that the point maps to.

Recall the condition distribution matrix A, where each row represents the distribution of z values mapped to a given proxy value. Our goal is to find acceptance probabilities such that the induced distribution on retained points is uniform over the protected attributes. By Lemma 10, such probabilities exist if A contains the uniform distribution in its convex hull. Consider the system $qA = U$, where $U = (\frac{1}{K}, ..., \frac{1}{K})$ and q must be a length ℓ stochastic vector. If there is a solution for q, we consider this a valid acceptance rate scheme and use it to derive the selection probabilities for our filtering problem. If there is not an exact solution (which will happen frequently) we take $\text{argmin}_q \|qA - U\|_2$ as our best acceptance rate scheme. *Because this involves solving a quadratic program, we refer to the proxies and accompanying selection schemes produced by this approach as QP (Quadratic Program) proxies.*

Algorithm 1 Finding Acceptance Probabilities ρ.

Input: proxy g, $D = \{(x_i, z_i)\}_{i=1}^n$, number of sensitive groups K
  for j in Range(g) **do**
    For k in $[K]$, let $a_k = \Pr_{z|x\sim D}[z = k|g(x) = j]$
    Let $\hat{r}_j = \Pr_{x\sim D}[g(x) = j]$
  Let A be the matrix with k^{th} row a_k and let $U = (\frac{1}{K}, ..., \frac{1}{K})$
  $q = \text{argmin}_q \|qA - U\|_2$ s.t. $q_i \geq 0$ and $\sum q_i = 1$
  For j in Range(g), set $\rho_j = \frac{q_j}{\hat{r}_j}$
  Let $C = \max_j \rho_j$ and normalize $\rho_j = \rho_j / C$
  return ρ, A

Algorithm 2 Filtering with ρ.

Input: g, ρ, $\mathcal{P}_x$
  Draw $x \sim \mathcal{P}_x$ and compute $g(x)$
  With probability $\rho_{g(x)}$, accept x into sample

▶ **Lemma 13** (Filtering According to ρ). *Consider acceptance probabilities ρ and conditional distribution matrix A returned by Algorithm 1. Then, if $U \in C(A)$, filtering according to ρ as in Algorithm 2 induces a uniform distribution over protected attributes.*

Proof. We want to show that the distribution over sensitive attributes in the *filtered set* is uniform. We begin by expressing the distribution over sensitive attributes in the filtered set constructively, as the distribution obtained from sampling according to ρ. From there, we plug in our definitions of $a_{k,j}$ as the j^{th} element in the k^{th} row of the conditional distribution matrix of z values given proxy values and as well as our definition of $\hat{r}_j$ as the marginal

probability that a proxy value is j. Finally, we use the result that $qA = U = (\frac{1}{K} ... \frac{1}{K})$

$$\sum_{j \in Range(g)} \rho_j \Pr[z = k, g(x) = j] = \sum_{j \in Range(g)} \rho_j \Pr[z = k | g(x) = j] \Pr[g(x) = j]$$
$$= \sum_{j \in Range(g)} a_{k,j} \hat{r}_j \rho_j = \sum_{j \in Range(g)} a_{k,j} q_j = \frac{1}{K}$$ ◀

4 Learning an (α, β) Proxy

We have discussed a proxy function $g : \mathcal{X} \to \mathbb{N}$ that maps samples to proxy groups and described the conditional distribution matrix A indicating the distribution of sensitive attributes *within* each proxy group. In Section 3, we showed how A can be used to derive acceptance probabilities for each group, such that under appropriate conditions, selecting according to these probabilities induces a uniform distribution over the protected attributes. Up until now, however, we have referenced A as fixed – we have used it to derive retention probabilities but have not described how it and the proxy can be generated. Recall that our proxy function $g \in \mathcal{G}$ takes the form of a decision tree, where each leaf is a *proxy group*. Therefore, each row in A, corresponding to the distribution over sensitive attributes in a given *proxy group*, also corresponds to the distribution over these attributes in a given *leaf*.

We grow our decision tree by sequentially making *splits* over the feature space – our tree will start as a stump and our matrix will have just one row, then we will split the tree into two leaves and the matrix will have two rows, and we will continue in this manner, splitting a leaf (and adding a row to the matrix) at each iteration. We will make these splits by employing a classification function from the pre-specified model class $\mathcal{H} \subseteq \{h : X \to \{0, 1\}\}$ assigned to each leaf. Because the two representations, as a matrix or a tree, afford different analytical advantages, we will continue to refer to both as we derive our algorithm. One advantage of the matrix representation is that it allows us to reason about the convex hull of a set of conditional distributions. Lemma 10 showed that there is a solution to $qA = U$ for a stochastic vector q if and only if U lies in the convex hull of A. Our goal will be to grow our tree (and the matrix A) so that the $\inf_{U' \in C(A)} \|U' - U\|_2$ shrinks at each iteration – until finally U is contained within (or sufficiently close to) $C(A)$.[6]

We begin with a geometric interpretation of $C(A)$ and describe how it changes as our tree and conditional distribution matrix expand. In particular, we grow a tree that has leaves V and keep track of the corresponding matrix A of sensitive attribute distributions conditional on their classification by the tree. We can always label the leaves of a tree with a binary sequence, so from now on we will identify each V with a binary sequence. Using this description, we derive sufficient conditions to decrease the Euclidean distance between $C(A)$ and U. We begin with several definitions that we will use to characterize $C(A)$.

▶ **Definition 14** (Vertex). *Let R be a bijective mapping of vertices to a rows in A. Then $V \in \{0, 1\}^{\mathbb{N}}$ is a vertex of $C(A)$ if $R(V)$ corresponds to a row a_i such that $a_i \notin C(\{a_j\}_{j<i})$.*

Note that in our context this means that each *row* of A corresponds to a *vertex* of $C(A)$ as long as it cannot be represented as a convex combination of the other rows. Next, we introduce the function that is used at a node of the decision tree to partition samples into the left or right child. It will also be convenient in our algorithm to make use of randomized splitting functions, so we handle both cases.

[6] Algorithms 1 and 2 and Lemma 13 extend easily to distributions other than the uniform.

▶ **Definition 15** (Splitting Function). *We call $h_V \in \mathcal{H}$ a deterministic splitting function at vertex V. A randomized splitting function $\tilde{h}_V \in \Delta\mathcal{H}$ is a distribution supported on a finite set of deterministic splitting functions $\{h_V^i\}_{i=1}^n$ such that $\tilde{h}_V(x) = h_V^i(x)$ with probability $\frac{1}{n}$ for all i.*

Each vertex V is paired with a splitting function $\tilde{h}_V$ operating on samples mapped to V. To model the *expected* action of a randomized splitting function, we introduce the notion of *sample weights*, where the weight of a sample x at V is the probability that x reaches V in its random walk down the tree (as determined by the randomized splitting function). Here, $V\backslash 0$ indicates the parent of V if V ends in 0, and $V\backslash 1$ indicates the parent if V ends in 1. Note that because V is a binary sequence, we can apply the modulo operator with the binary representation of 2 to isolate the last digit.

▶ **Definition 16** (Sample Weights). *The weight of a sample x at vertex V is defined as follows: $w_0(x) = 1$ and for $V \neq 0$, $w_V(x) = \left\{ \begin{array}{ll} w_{V\backslash 0}(x) \cdot \mathbb{E}[\tilde{h}_{V\backslash 0}(x)] & \text{if } V \mod 2 = 0 \\ w_{V\backslash 1}(x) \cdot \mathbb{E}[1 - \tilde{h}_{V\backslash 1}(x)] & \text{if } V \mod 2 = 1 \end{array} \right\}$*

We distinguish between V and the collection of weighted samples represented by V, l_V.

▶ **Definition 17** (Collection of Weighted Samples at V). *Given randomized splitting functions $\{\tilde{h}_i\}_{i=0}^V$, the collection of weighted samples at V is denoted by $l_V = \{w_V(x), (x, z) : (x, z) \in S\}$.*

▶ **Definition 18** (Vertex Split). *A vertex split results from applying $\tilde{h}_V$ to $x \in l_V$, where $l_{V0} = \{w_{V0}(x), (x, z) : (x, z) \in S\}$ and $l_{V1} = \{w_{V1}(x), (x, z) : (x, z) \in S\}$.*

After V is split into V0 and V1, V is no longer a vertex, whereas V0 and V1 may be. So, the number of leaves in the tree, and therefore the number of rows in A, increased by at most 1.

4.1 Growing the Convex Hull and Learning a Splitting Function

Imagine that we have started to grow our proxy tree, but U is not in $C(A)$. We would like to expand $C(A)$ to contain U, and intuitively, we might like to expand $C(A)$ in the direction of U. One way to do so is to choose a vertex V to split into two vertices, $V1$ and $V0$. We assume that $V1$ is the split such that $R(V1) - R(V)$ is most in the direction of $U - U'$, where U' is the closest point in Euclidean distance to U in $C(A)$.

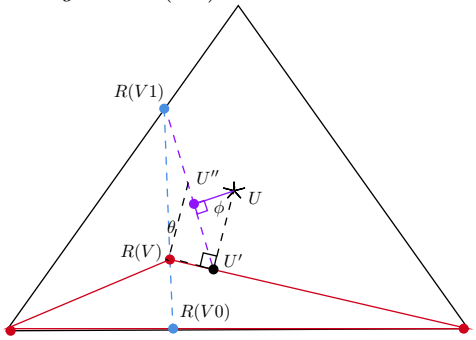
▶ **Definition 19** (Convex Hull Notation). *Let θ be the angle between $R(V1) - R(V)$ and $U - U'$, and U'' be the closest point to U on the line segment $R(V1) - U'$:*

$$U' = \arg \min_{U^* \in C(A)} \|U - U^*\|_2$$

$$\cos\theta = \frac{\langle R(V1) - R(V), U - U' \rangle}{\|R(V1) - R(V)\|_2 \|U - U'\|_2}$$

$$U'' = tU' + (1-t)R(V1) \text{ where}$$

$$t = \underset{0<t^*<1}{\operatorname{argmin}} \|U - (t^*U' + (1-t^*)R(V1))\|_2$$



We show that, given certain assumptions, we can lower bound how much this splitting process will decrease the distance from $C(A)$ to U. The first condition in Lemma 20 will be used to derive an objective function over which we can optimize to find a splitting function. The

second and third conditions limit the theory to the case where we can prove our progress lemma. The second condition says that the distance between $R(V)$ and $R(V1)$ has to be sufficiently large compared to the existing distance between the uniform distribution and its projection onto $C(A)$. The third condition is needed for the proof, allowing us to make arguments based on right triangles – it is satisfied when the second condition is met and the angle between $R(V1) - R(V)$ and $U - U'$ is not too large. As these conditions are potentially limiting theoretically, we verify that they are indeed frequently satisfied in the experiments.

▶ **Lemma 20** (Progress via Vertex Split). *When a vertex V is split, forming new vertices $V0$ and $V1$, the distance from the convex hull to U decreases by at least a factor of $1-\gamma$ if*

$$\langle R(V1) - R(V), U - U' \rangle / \|U - U'\|_2 \geq f(\gamma) \text{ and} \tag{1}$$
$$\|R(V1) - R(V)\|_2 \geq (1-\gamma)^{-1}\sqrt{2\gamma - \gamma^2}\|U - U'\|_2, \quad R(V1) - U' \perp U - U'' \text{ where}$$

$$f(\gamma) := \sqrt{(2\gamma - \gamma^2)\left(2 - \|R(V) - U'\|_2^2 + 2\|R(V) - U'\|_2(1-\gamma)\sqrt{(\gamma^2 - 2\gamma)\|R(V) - U'\|_2^2 + 2}\right)}$$

▶ Remark 21. These are sufficient, but not necessary, conditions for a split to make sufficient progress. Empirically, we simply require that each split decreases the distance from the convex hull to U by at least a factor of $1-\gamma$ for the algorithm to continue.

To summarize, these conditions ask that we split a vertex of the convex hull (equivalently a leaf of the proxy tree), so that the convex hull expands in the direction of the target vector. In other words, we want to split a leaf into the over-represented groups in one child and the under-represented groups in the other child, without violating the disclosure constraints. Lemma 1 also involves conditions that make sure that this split is sufficiently large to move the convex hull closer to the uniform rather than making minute progress. Having identified a sufficient condition for a split to make suitable progress toward containing the uniform distribution within the convex hull, we present a subroutine to find an α-proxy. We first express Equation (1) in a form amenable to use in a linear program:

▶ **Lemma 22** (Objective Function). *Let m_V be the number of samples in l_V and let h_V be the splitting function for vertex V. The condition $\frac{\langle R(V1)-R(V), U-U'\rangle}{\|U-U'\|_2} \geq f(\gamma)$ is equivalent to*

$$\sum_{i=1}^{m_V} w_V(x_i)h_V(x_i)(-Q + \sum_{k=1}^{K} \mathbb{1}_{z_i=k}(U'_k - U_k)) \leq 0$$
$$\text{for } Q_{V,U',\gamma} := \|U' - U\|f(\gamma) + \frac{\sum_{i=1}^{m_V} w_V(x_i)\sum_{k=1}^{K}\mathbb{1}_{z_i=k}(U'_k - U_k)}{\sum_{i=1}^{m_V} w_V(x_i)}$$

Proof. We begin by expanding the scaled dot product between $R(V1) - R(V)$ and $U - U'$:

$$\frac{\langle R(V1) - R(V), U - U'\rangle}{\|U - U'\|} = \sum_{j=1}^{m_V} w_V(x_j)\sum_{k=1}^{K}\mathbb{1}_{z_j=k}\left(\frac{h_V(x_j)}{\sum_{j=1}^{m_V} w_V(x_j)h_V(x_j)} - \frac{1}{\sum_{j=1}^{m_V} w_V(x_j)}\right)\frac{U_k - U'_k}{\|U - U'\|}$$

Asking $\frac{\langle R(V1)-R(V), U-U'\rangle}{\|U-U'\|} \geq f(\gamma)$ is equivalent to asking $\frac{\langle R(V1)-R(V), U'-U\rangle}{\|U'-U\|} \leq f(\gamma)$ or:

$$\frac{\sum_{j=1}^{m_V} w_V(x_j)h_V(x_j)\sum_{k=1}^{K}\mathbb{1}_{z_j=k}(U'_k - U_k)}{\sum_{i=1}^{m_V} w_V(x_j)h_V(x_j)} \leq \|U' - U\|f(\gamma) + \frac{\sum_{j=1}^{m_V} w_V(x_j)\sum_{k=1}^{K}\mathbb{1}_{z_j=k}(U'_k - U_k)}{\sum_{j=1}^{m_V} w_V(x_j)} \tag{2}$$

Finally, the right-hand side is constant given V, U', and γ. Therefore, we represent it by a constant $Q_{V,U',\gamma} := \|U' - U\| f(\gamma) + \frac{\sum_{j=1}^{m_V} w_V(x_j) \sum_{k=1}^{K} \mathbb{1}_{z_j=k} (U'_k - U_k)}{\sum_{j=1}^{m_V} w_V(x_j)}$. This allows us to rewrite Equation (2) as

$$\sum_{j=1}^{m_V} w_V(x_j) h_V(x_j) \left(-Q_{V,U',\gamma} + \sum_{k=1}^{K} \mathbb{1}_{z_j=k} (U'_k - U_k) \right) \leq 0 \quad \blacktriangleleft$$

We use Lemma 22 to form a cost-sensitive classification problem for vertex V, where the constraints make sure that any candidate proxy is no more than α-disclosive:

$$\min_{h_V \in \mathcal{H}} \sum_{i=1}^{m_V} w_V(x_i) h_V(x_i) \left(-Q_{V,U',\gamma} + \sum_{k=1}^{K} \mathbb{1}_{z_i=k} (U'_k - U_k) \right) \text{ s.t. } \forall k \tag{3}$$
$$\left| \frac{\sum_{i=1}^{m_V} w_V(x_i) h_V(x_i) \mathbb{1}_{z_i=k}}{\sum_{i=1}^{m_V} w_V(x_i) h_V(x_i)} - r_k \right| \leq \alpha \text{ and } \left| \frac{\sum_{i=1}^{m_V} w_V(x_i)(1 - h_V(x_i)) \mathbb{1}_{z_i=k}}{\sum_{i=1}^{m_V} w_V(x_i)(1 - h_V(x_i))} - r_k \right| \leq \alpha$$

Next, we will appeal to strong duality to derive the corresponding Lagrangian. We note that computing an approximately optimal solution to the linear program corresponds to finding approximate equilibrium strategies for both players in the game in which one player, the "Learner," controls the primal variables and aims to minimize the Lagrangian value. The other player, the "Auditor," controls the dual variables and seeks to maximize the Lagrangian value. If we construct our algorithm in such a way that it simulates repeated play of the Lagrangian game such that both players have sufficiently small regret, we can apply Theorem 1 to conclude that our empirical play converges to an approximate equilibrium of the game. Furthermore, our algorithm will be *oracle efficient*: it will make polynomially many calls to oracles that solve weighted cost-sensitive classification problems over $\mathcal{H}$.

To turn Program (3) into a form amenable to our two-player zero-sum game formulation, we expand $\mathcal{H}$ to $\Delta\mathcal{H}$, allow our splitting function to be *randomized*, and take expectations over the objective and constraints with respect to deterministic splitting functions drawn according to $\tilde{h}_V$. Doing so yields the following CSC problem to be solved for vertex V:

$$\begin{aligned}
\min_{\tilde{h}_V \in \Delta\mathcal{H}} \quad & \mathbb{E}_{h_V \sim \tilde{h}_V} \sum_{i=1}^{m_V} w_V(x_i) h_V(x_i) \left(-Q_{V,U',\gamma} + \sum_{k=1}^{K} \mathbb{1}_{z_i=k} (U'_k - U_k)) \right) \\
\text{s.t.} \quad & \mathbb{E}_{h_V \sim \tilde{h}_V} \sum_{i=1}^{m_V} w_V(x_i) \tilde{h}_V(x_i) (\mathbb{1}_{z_i=k} - r_k - \alpha) \leq 0 \ \forall k, \\
& \mathbb{E}_{h_V \sim \tilde{h}_V} \sum_{i=1}^{m_V} w_V(x_i)(1 - h_V(x_i)) (\mathbb{1}_{z_i=k} - r_k - \alpha) \leq 0 \ \forall k, \\
& \mathbb{E}_{h_V \sim \tilde{h}_V} \sum_{i=1}^{m_V} w_V(x_i) h_V(x_i) (r_k - \mathbb{1}_{z_i=k} - \alpha) \leq 0 \ \forall k, \\
& \mathbb{E}_{h_V \sim \tilde{h}_V} \sum_{i=1}^{m_V} w_V(x_i)(1 - h_V(x_i)) (r_k - \mathbb{1}_{z_i=k} - \alpha) \leq 0 \ \forall k
\end{aligned} \tag{4}$$

We solve this constrained optimization problem by simulation a zero-sum two-player game on the Lagrangian dual. Given dual variables $\lambda \in \mathbb{R}^{4K}_{\geq 0}$ such that $\|\lambda\|_2 \leq \lambda_{max}$ for some constant λ_{max}, the Lagrangian of Program (4) is:

$$L(\lambda, \tilde{h}_V) = \mathbb{E}_{h_V \sim \tilde{h}_V} \sum_{i=1}^{m_V} w_V(x_i) \Bigg(-Q_{V,U',\gamma} h_V(x_i) + \sum_{k=1}^{K} h_V(x_i) \mathbb{1}_{z_i=k} \left(U'_k - U_k\right) + \\ \left(\lambda_{k,1} h_V(x_i) + \lambda_{k,0}(1 - h_V(x_i))\right) \left(\mathbb{1}_{z_i=k} - r_k - \alpha\right) + \\ \left(\lambda_{k,3} h_V(x_i) + \lambda_{k,2}(1 - h_V(x_i))\right) \left(r_k - \mathbb{1}_{z_i=k} - \alpha\right) \Bigg)$$

Given the Lagrangian, solving Program (4) is equivalent to solving the minimax problem $\min_{\tilde{h}_V \in \Delta\mathcal{H}} \max_{\lambda \in \mathbb{R}^{4K}_{\geq 0}} L(\lambda, \tilde{h}_V) = \max_{\lambda \in \mathbb{R}^{4K}_{\geq 0}} \min_{\tilde{h}_V \in \Delta\mathcal{H}} L(\lambda, \tilde{h}_V)$, where the minimax theorem holds because the range of the primal variable, i.e., $\Delta\mathcal{H}$ is convex and compact, the range of the dual variable, i.e., $\mathbb{R}^{4K}_{\geq 0}$ is convex, and the Lagrangian function L is linear in both primal and dual variables. Therefore, we focus on solving the minimax problem, which can be seen as a two-player zero-sum game between the primal player (the Learner) who is controlling $\tilde{h}_V$ and the dual player (the Auditor) who is controlling λ. Using no-regret dynamics, we will have the Learner deploy its best response strategy in every round, which will be reduced to a call to $CSC(\mathcal{H})$ and let the Auditor with strategies in $\Lambda = \{\lambda : 0 \leq \lambda \leq \lambda_{max}\}$ play according to Online Projected Gradient Descent [45].

Our local algorithm for splitting a vertex is described in Algorithm 3, and its guarantee is given in Theorem 2. We note that the algorithm returns a distribution over $\mathcal{H}$. Given an action λ of the Auditor, we write $LC(\lambda)$ for the vector of costs for labeling each data point as 1. We view our costs as the inner product of the outputs of a deterministic splitting function h_V on the m_V points and corresponding cost vector. We define the cost for labeling an example 0 to be 0 for all x ($c^0(x) = 0$), and the cost for labeling an example 1 as:

$$c^1(x) = w_V(x)(-Q_{V,U',\gamma} + \sum_{k=1}^{K} \mathbb{1}_{z_j=k} \left(U'_k - U_k\right) + \left(\lambda_{k,1} - \lambda_{k,0}\right) \left(\mathbb{1}_{z=k} - r_k - \alpha\right) \\ + \left(\lambda_{k,3} - \lambda_{k,2}\right) \left(r_k - \mathbb{1}_{z=k} - \alpha\right))$$

Algorithm 3 Learning a Splitting Function.

Input: $\{w_V(x_i), (x_i, z_i)\}_{i=1}^{m_V}$, model class $\mathcal{H}$, $CSC(\mathcal{H})$, α, ϵ, γ
Set $\lambda_{max} = m(K-1)/K\epsilon + 2$ and $T = \lceil (2Km\left(1+\alpha\right)\lambda_{\max}/\epsilon)^2 \rceil$
Initialize $\lambda_k = 0\ \forall k$
for $t = 1 \ldots T$ **do**
  $h_t = \text{argmin}_{h \in \mathcal{H}} \langle LC(\lambda), h \rangle$
  $\lambda_t = \lambda_{t-1} + t^{-1/2} \left(\nabla_\lambda L\right)^+$; If $\|\lambda_t\| > \lambda_{\max}$, set $\lambda_t = \lambda_{\max} \frac{\lambda_t}{\|\lambda_t\|}$
return $\tilde{h}_V :=$ uniform distribution over h_V^t

▶ **Theorem 2** (Learning an $(\alpha + \epsilon)$-Disclosive Proxy). *Fix α, ϵ, suppose $\mathcal{H}$ has finite VC dimension, and suppose $\exists\ \tilde{h}_V^* \in \Delta\mathcal{H}$ that is a feasible solution to Program (4). Then, Algorithm 3 returns a distribution $\tilde{h}_V$ that is an ϵ-optimal solution to Program (4).*

Theorem 2 says that with appropriate conditions on the model class $\mathcal{H}$ and access to $CSC(\mathcal{H})$, Algorithm 3 returns a model satisfying the conditions of Program 4 (i.e. produces an acceptable split) up to an additive factor of ϵ. A few requirements of this theorem may not hold in practice and thus motivate our experiments. The choice of a base model class $\mathcal{H}$ impacts whether a feasible solution exists – typically more complex model classes will be more likely to contain a feasible solution, but this complexity will impact the generalization bounds. Also, the guarantee relies on Algorithm 3 having access to a cost sensitive classification oracle. In practice, we typically do not have such an oracle so must use a heuristic.

4.2 Decision Tree Meta-Algorithm

Finally, we use these results to greedily construct a proxy $g : \mathcal{X} \to \mathbb{N}$. We do this iteratively using a decision tree, where leaves correspond to proxy groups. We split the data into these leaves in such a way that when we consider the distribution of groups in each leaf, the uniform vector is contained in their convex hull. This allows us to select a balanced set in expectation. In addition, we require that the proxy be α-disclosive at every step. We grow the tree as follows, for some tolerance β: (1) If $\|U - C(A)\|_2 \leq \beta$, output the tree. (2) Otherwise, look for a leaf to split. If we find a suitable split, make it, and continue. If not, output the tree. To determine if a split is suitable, we use the results from Section 4.1: for fixed approximation factor ϵ, disclosivity budget $\alpha - \epsilon$, and progress parameter γ, a splitting function $\tilde{h}_V$ must be an ϵ-approximate solution to Program 4 (and therefore no more than α-disclosive) at vertex V. If we can find such an $\tilde{h}_V$ for at least $\frac{\ln\beta - \ln\sqrt{2}}{\ln 1-\gamma}$ rounds, the decision tree will be an (α, β) proxy.

Algorithm 4 Learning an (α, β) Proxy.

Input: $D = \{x_i, z_i\}_{i=1}^n$, $CSC(\mathcal{H})$, α, ϵ, γ, β
  while $\inf_{U' \in C(A)} \|U' - U\|_2 > \beta$ **do**
    Apply Algorithm 3 to find feasible split (if no feasible split, terminate)
    Expand tree T and re-calculate A, $C(A)$
  return T, A

▶ **Theorem 3** (Learning an (α, β) Proxy). *If the conditions of Lemma 20 are satisfied at every split, Algorithm 4 produces an (α, β) proxy in-sample within $\frac{\ln\beta - \ln\sqrt{2}}{\ln 1-\gamma}$ rounds.*

Proof. Let A_{i^*} be the conditional distribution matrix returned by Algorithm 4 after i^* rounds. Our goal is to produce A_{i^*} such that $\|U - C(A_{i^*})\|_2 \leq \beta$. Let A_0 be the initial conditional distribution matrix, and observe that if we decrease the distance from the current conditional distribution matrix to U by a factor of $1 - \gamma$ each round, at round i, $\|U - C(A_i)\|_2 \leq (1-\gamma)^i \|U - C(A_0)\|_2$. Further, recall that $\|U - C(A_0)\|_2 \leq \sqrt{2}$ because both U and $C(A_0)$ must lie in the unit simplex. Setting $\|U - C(A_{i^*})\|_2 \leq \beta$, we have $\beta \leq (1-\gamma)^{i^*}\sqrt{2} \implies \frac{\beta}{\sqrt{2}} \leq (1-\gamma)^{i^*} \implies i* \geq \frac{\ln\beta - \ln\sqrt{2}}{\ln(1-\gamma)}$. Then, after $i^* = \frac{\ln\beta - \ln\sqrt{2}}{\ln(1-\gamma)}$ rounds, $\|U - C(A_{i^*})\|_2 \leq \gamma$. Finally, because our linear program constrains splits to only those that guarantee α-disclosiveness, the final proxy must be α-disclosive in-sample. ◀

Theorem 3 allows us to upper bound the number of times that Algorithm 4 performs a split and, therefore, the number of unique proxy groups generated. The theorem's hypothesis states, informally, that it must be possible to find a splitting function at each round that makes both a sufficiently *large* split (i.e. the new vertex is sufficiently far from the old vertex compared to the current distance from the convex hull to the target uniform) and the split is sufficiently in the direction of the target. This theorem, in turn, allows us to state generalization bounds depending on both the number and size of each proxy group.

▶ **Theorem 4** (Generalization). *Let $\epsilon, \delta, \gamma > 0$ and G be the proxy class. Let there be K sensitive groups. If each proxy group has at least $\frac{1}{2\epsilon^2} \ln \frac{8K \cdot VC(\mathcal{G})\left(\ln\beta - \ln\sqrt{2}\right)}{\delta \ln(1-\gamma)}$ samples, with probability $1 - \delta$, an (α, β) proxy in-sample will be an $(\alpha + 2\epsilon, \beta + K\epsilon\sqrt{\frac{\ln\beta - \ln\sqrt{2}}{\ln(1-\gamma)}})$ proxy out-of-sample.*

Theorem 4 presents the number of samples needed in each proxy group to obtain a sufficiently small generalization gap in both the disclosure level and imbalance – it is based on the size of the *smallest proxy group in-sample*, which might get quite small in practice.

Furthermore, the generalization gap for β scales by an additive factor of the number of sensitive groups, K. Therefore, as our problem becomes more challenging, more samples are required to achieve a proxy that performs similarly out-of-sample compared to in-sample.

5 Experiments

Here, we test our two main methodological contributions. The first is to use Algorithms 1 and 2 to solve $\min_q \|qA - U\|_2$ subject to $q_i \geq 0 \ \forall i$ and $\sum_i q_i = 1$ and derive the corresponding acceptance probabilities ρ for the given proxy. The second is to additionally use Algorithms 3 and 4 to learn a decision-tree proxy *guaranteed* not to exceed a specified level of disclosure.

1. *QP Regression and Decision Tree* Proxies: We train a multinomial logistic regression model or decision tree to *directly predict* the sensitive attribute but select our acceptance probabilities by employing Algorithms 1 and 2.
2. (α, β) Proxy: We use Algorithm 4 to develop a proxy for a specified disclosure budget.

We compare the performances of these proxy functions against those of two baselines:[7]

1. *Naive Regression and Decision Tree* proxies: We train models to *directly predict* sensitive attributes then sample the same number of points from each predicted group, inducing a conditional distribution matrix of the distribution of sensitive attributes in each proxy group. We then calculate the degree of disclosure and imbalance of the sampled set.
2. SMOTE [8]: We train a decision tree to *directly predict* groups and then, using these predictions as input for SMOTE, balance the data by synthesizing minority examples.

5.1 Data, Hyperparameters, and Compute Time

We evaluate the disclosure, α, and imbalance, β, obtained by each proxy filtering scheme on the Bank Marketing [25, 12], Adult [12], and Communities and Crime data sets [12, 35, 37, 38, 39, 30], for which we have 5, 4, and 12 sensitive attribute values, respectively. The Marketing data set consists of 45211 labeled samples with 48 non-sensitive attributes and a sensitive attribute of job type. The downstream classification goal is to predict whether a client will subscribe a term deposit based on a phone call marketing campaign of a Portuguese banking institution. The Adult data set consists of 48842 labeled samples with 14 non-sensitive attributes, and we select race as the sensitive attribute. The associated classification task is to determine whether individuals make over \$50K dollars per year. The Communities and Crime data set consists of 1594 samples with 132 non-sensitive features, race as the sensitive group, and the number of violent crimes per population as the prediction task.

For each experiment, we run trials with 20 different seeds, and for each seed, we input a grid of values with increments of 0.1 for the disclosure parameter, α, evenly spaced between 0 and 1. We then average over the seeds for each α and calculate empirical 95% confidence intervals (which are displayed as the shaded region around each line in the plots). Each data set is split into three parts of sizes 50%, 30%, and 20%. The first is used to train the proxy. The second is used first to test the filtering effects of the proxy out-of-sample and then to train a classification model on to study downstream performance. The third is the set upon which we apply these classifiers trained on filtered and unfiltered data to see how

[7] For the Naive Proxies, QP Proxies, and SMOTE, we interpolate between a uniform and proxy-specific sampling strategy by post-processing: We predict z with the proxy and then, with probability $\eta \in [0, 1]$, uniformly re-assign the prediction. Finally, we apply Algorithm 2 to sample according to the post-processed proxy labels and plot the balance and disclosure of the corresponding data set *with respect to the post-processed proxy values*. We use a large point marker for the results without post-processing

the group-wise accuracy levels are affected. For brevity, we will refer to these three splits as the "Train" set, "Test" set, and "Post-Test" set, respectively. See Appendix B for an analysis of downstream fairness effects induced by our strategy.

On the Adult and Communities and Crime data sets, one run over the grid of α values typically took between 20 minutes and two hours for the (α, β) proxy. On the Marketing data set, running one full experiment over the grid of α values took about three hours. The parameter γ was set to 0.0001, the maximum height of the proxy tree was set to 15, and the learning process was stopped once the distance between the convex hull of the conditional distribution matrix and the uniform distribution fell below 0.05. As we used publicly available tabular data sets that has already been cleaned, there were no missing values.

Finally, the choice of oracle (the base model class for the (α, β) proxy) is heuristic – as we do not have a true cost-sensitive classification oracle for Algorithm 3, we choose two models that allow us to predict the cost of each example and then classify based on the cost's sign. We experiment with a linear threshold function – the paired regression classifier (PRC) used in [19] and defined below – as well as the XGBoost Regressor model. We found that the PRC was simpler and seemed to perform at least as well as the XGBoost Regressor, so we relegate the analysis for the latter to Appendix B.

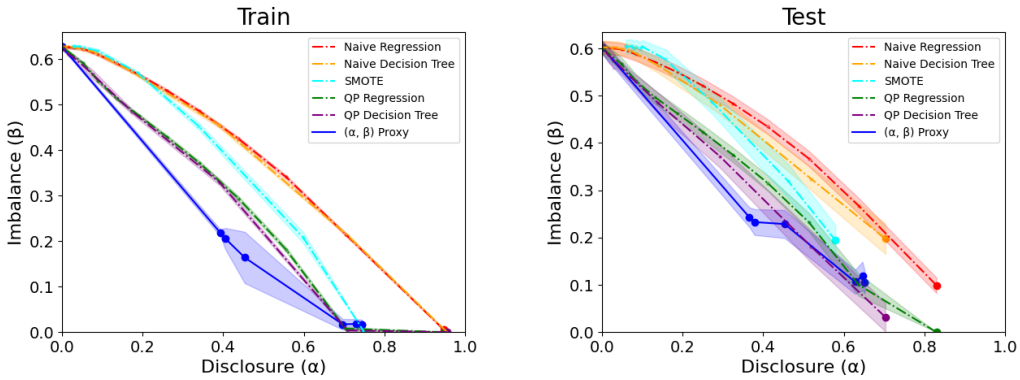
▶ **Definition 23** (Paired Regression Classifier [19]). *The paired regression classifier operates as follows: We form two weight vectors, z^0 and z^1, where z_i^k corresponds to the penalty assigned to sample i in the event that it is labeled k. For the correct labeling of x_i, the penalty is 0. For the incorrect labeling, the penalty is the current sample weight of the point, w_V. We fit two linear regression models h^0 and h^1 to predict z^0 and z^1, respectively, on all samples. Then, given a new point x, we calculate $h^0(x)$ and $h^1(x)$ and output $h(x) = \text{argmin}_{k \in \{0,1\}} h^k(x)$.*

5.2 Results

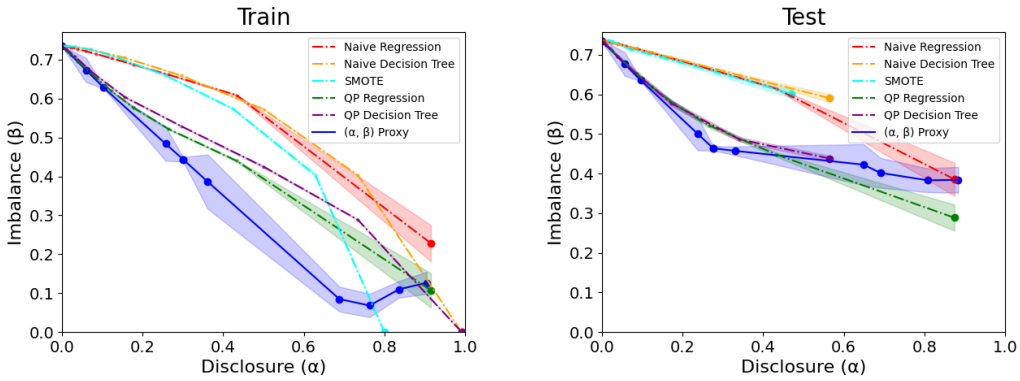
In Figure 1, on the Communities data set, the (α, β) proxy Pareto-dominates the other approaches in sample, while the QP proxies Pareto-dominate SMOTE and the Naive proxies. All methods generalize well. On the Adult data set, the (α, β) proxy primarily dominates the remaining approaches in-sample. The generalization performance for all methods, but particularly the (α, β) proxy, is weaker on the Adult data set. This is likely because there are slightly more sensitive groups than in the Communities data set, and the acceptance probabilities were sparse. On the Marketing data set, the (α, β) and QP Decision Tree proxies exhibit favorable performance in-sample, driving the imbalance to just above zero at higher levels of disclosure. The plot on the test set shows a more modest improvement in balance for all methods. One source of variance in Figure 1 is the generalization performance by the (α, β) proxy. We believe this to be due to the size of the smallest proxy group being quite low (especially for the Marketing data set which has 12 sensitive groups). Recall that the generalization gap depends directly on this quantity. There is also nothing in our method to prevent a sparse sampling scheme. Empirically, we found that in cases where generalization results were weak, the acceptance probabilities were nonzero for only a handful of the final proxy groups. Addressing these weaknesses, if possible, could strengthen our approach.

5.3 Discussion and Future Work

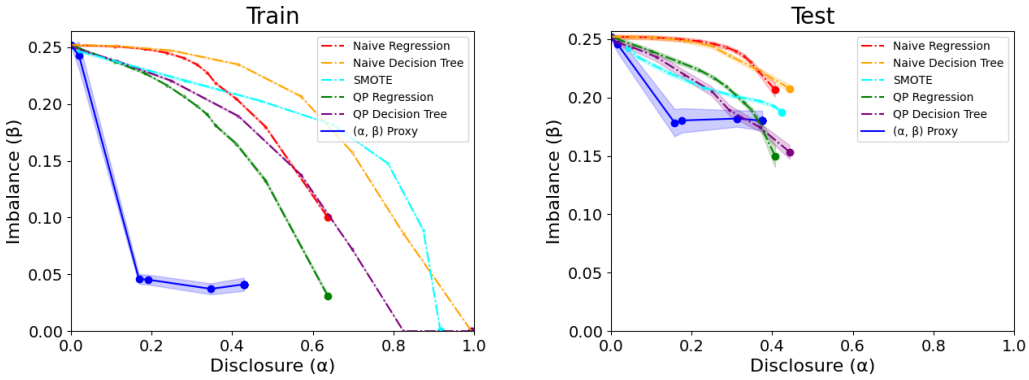
Our primary conceptual point is that even though the final goal (balance) references the protected attributes, it is a condition on the aggregate composition of the final selected set. Therefore, achieving it does not necessarily require finding a predictor strongly correlated with the protected attribute. We emphasize that while the QP proxies (our secondary contribution) are appealingly simple and provide a range of disclosure levels *after* post-processing, they



(a) Communities.



(b) Adult.



(c) Marketing.

Figure 1 Trade-off of Disclosure and Balance of Proxies on Communities, Adult, and Marketing.

still involve explicitly training a classifier for the attribute. In contrast, the (α, β) proxy (our primary contribution) never involves training a classifier at any step of the process that is more disclosive than a pre-specified threshold. While this does not solve the challenging legal and technical problems associated with proxy use in high-stakes selection processes, it takes a step in this direction by permitting controlled trade-offs between balance and disclosure.

References

1 Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna M. Wallach. A reductions approach to fair classification. *CoRR*, abs/1803.02453, 2018. `arXiv:1803.02453`.

2 Alekh Agarwal, Miroslav Dudík, and Zhiwei Steven Wu. Fair regression: Quantitative definitions and reduction-based algorithms. *CoRR*, abs/1905.12843, 2019. arXiv:1905.12843.

3 Larry Alexander. What makes wrongful discrimination wrong? biases, preferences, stereotypes, and proxies. *University of Pennsylvania Law Review*, 141(1):149–219, 1992. URL: http://www.jstor.org/stable/3312397.

4 Gustavo E. A. P. A. Batista, Ana Lúcia Cetertich Bazzan, and Maria Carolina Monard. Balancing training data for automated annotation of keywords: a case study. In *WOB*, 2003.

5 Gustavo E. A. P. A. Batista, Ronaldo C. Prati, and Maria Carolina Monard. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.*, 6(1):20–29, June 2004. doi:10.1145/1007730.1007735.

6 Patrick Billingsley. *Probability and Measure*. John Wiley and Sons, second edition, 1986.

7 Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

8 Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1):321–357, June 2002.

9 Yifan Cui, Hongming Pu, Xu Shi, Wang Miao, and Eric Tchetgen Tchetgen. Semiparametric proximal causal inference. *Journal of the American Statistical Association*, 0(0):1–12, 2023. doi:10.1080/01621459.2023.2191817.

10 Emily Diana, Wesley Gill, Michael Kearns, Krishnaram Kenthapadi, Aaron Roth, and Saeed Sharifi-Malvajerdi. Multiaccurate proxies for downstream fairness. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, pages 1207–1239, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3531146.3533180.

11 Georgios Douzas, Fernando Bacao, and Felix Last. Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Information Sciences*, 465:1–20, October 2018. doi:10.1016/j.ins.2018.06.056.

12 Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL: http://archive.ics.uci.edu/ml.

13 Marc N. Elliott, Peter A. Morrison, Allen M. Fremont, Daniel F. McCaffrey, Philip M Pantoja, and Nicole Lurie. Using the census bureau's surname list to improve estimates of race/ethnicity and associated disparities. *Health Services and Outcomes Research Methodology*, 9:69–83, 2009.

14 Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, 1996.

15 Hui Han, Wenyuan Wang, and Binghuan Mao. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In *International Conference on Intelligent Computing*, 2005.

16 Peter E. Hart. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, pages 515–516, 1968.

17 Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328, 2008.

18 Gabbrielle Johnson. Algorithmic bias: On the implicit biases of social technology, May 2020. URL: http://philsci-archive.pitt.edu/17169/.

19 Michael Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *International Conference on Machine Learning*, pages 2564–2572. PMLR, 2018.

20 David Madras, Elliot Creager, Toniann Pitassi, and Richard S. Zemel. Learning adversarially fair and transferable representations. *CoRR*, abs/1802.06309, 2018. arXiv:1802.06309.

21 Inderjeet Mani and Jianping Zhang. knn approach to unbalanced data distributions: A case study involving information extraction. *Workshop on Learning from Imbalanced Datasets II, ICML*, 126:1–7, 2003.

22 Daniel Mccaffrey, Greg Ridgeway, and Andrew Morral. Propensity score estimation with boosted regression for evaluating causal effects in observational studies. *Psychological methods*, 9:403–25, January 2005. doi:10.1037/1082-989X.9.4.403.

23 Giovanna Menardi and Nicola Torelli. Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, 28:92–122, 2012.

24 Wang Miao, Zhi Geng, and Eric J. Tchetgen Tchetgen. Identifying causal effects with proxy variables of an unmeasured confounder. *Biometrika*, 105 4:987–993, 2016. URL: https://api.semanticscholar.org/CorpusID:88521475.

25 Sérgio Moro, Paulo Cortez, and Paulo Rita. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62:22–31, 2014. doi:10.1016/j.dss.2014.03.001.

26 Hien M. Nguyen, Eric W. Cooper, and Katsuari Kamei. Borderline over-sampling for imbalanced data classification. *Int. J. Knowl. Eng. Soft Data Paradigms*, 3:4–21, 2009.

27 Valerio Perrone, Michele Donini, Muhammad Bilal Zafar, Robin Schmucker, Krishnaram Kenthapadi, and Cédric Archambeau. Fair bayesian optimization. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '21, pages 854–863, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3461702.3462629.

28 Flavien Prost, Hai Qian, Qiuwen Chen, Ed H. Chi, Jilin Chen, and Alex Beutel. Toward a better trade-off between performance and fairness with kernel-based distribution matching. *ArXiv*, abs/1910.11779, 2019. URL: https://api.semanticscholar.org/CorpusID:204900934.

29 Hongxiang Qiu, Xu Shi, Wang Miao, Edgar Dobriban, and Eric Tchetgen Tchetgen. Doubly robust proximal synthetic controls, 2023. arXiv:2210.02014.

30 M. A. Redmond and A. Baveja. A data-driven software tool for enabling cooperative information sharing among police departments, 2002.

31 Xu Shi, Kendrick Li, Wang Miao, Mengtong Hu, and Eric Tchetgen Tchetgen. Theory for identification and inference with synthetic controls: A proximal causal inference framework, 2023. arXiv:2108.13935.

32 Eric J Tchetgen Tchetgen, Andrew Ying, Yifan Cui, Xu Shi, and Wang Miao. An introduction to proximal causal learning, 2020. arXiv:2009.10982.

33 I. Tomek. An experiment with the edited nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-6(6):448–452, 1976. doi:10.1109/TSMC.1976.4309523.

34 I. Tomek. Two modifications of cnn. *IEEE Transactions on Systems, Man, and Cybernetics*, 6:769–772, 1976.

35 Bureau of the Census U. S. Department of Commerce. Census of population and housing 1990 united states: Summary tape file 1a & 3a (computer files).

36 U.S. Students for fair admissions, inc. v. president and fellows of harvard college, 2023.

37 Bureau Of The Census Producer U.S. Department Of Commerce, 1992.

38 Bureau Of The Census Producer U.S. Department Of Commerce. U.s. department of justice, bureau of justice statistics, law enforcement management and administrative statistics (computer file), 1992.

39 Federal Bureau of Investigation U.S. Department of Justice. Crime in the united states (computer file), 1995.

40 Ioan Voicu. Using first name information to improve race and ethnicity classification. *Statistics and Public Policy*, 5:1–13, 2016.

41 Michael R. Wickens. A note on the use of proxy variables. *Econometrica*, 40(4):759–761, 1972. URL: http://www.jstor.org/stable/1912971.

42 Dennis L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Trans. Syst. Man Cybern.*, 2:408–421, 1972.

43 Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2018. arXiv:1710.09412.

44 Yan Zhang. Assessing fair lending risks using race/ethnicity proxies. *Comparative Political Economy: Regulation eJournal*, 2016.

45 Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the Twentieth International Conference on Machine Learning*. Washington, DC, 2003.

A Omitted Proofs

▶ **Lemma 20** (Progress via Vertex Split). *When a vertex V is split, forming new vertices $V0$ and $V1$, the distance from the convex hull to U decreases by at least a factor of $1-\gamma$ if*

$$\langle R(V1) - R(V), U - U'\rangle / \|U - U'\|_2 \geq f(\gamma) \text{ and} \tag{1}$$
$$\|R(V1) - R(V)\|_2 \geq (1-\gamma)^{-1}\sqrt{2\gamma-\gamma^2}\|U-U'\|_2, \quad R(V1) - U' \perp U - U'' \text{ where}$$

$$f(\gamma) := \sqrt{(2\gamma-\gamma^2)\left(2 - \|R(V)-U'\|_2^2 + 2\|R(V)-U'\|_2(1-\gamma)\sqrt{(\gamma^2-2\gamma)\|R(V)-U'\|_2^2+2}\right)}$$

Proof. We want to find sufficient conditions for $\|U-U''\| \leq (1-\gamma)\|U-U'\|$. Let ϕ be the angle between the vectors $U-U''$ and $U-U'$. Then $\|U-U''\| = \|U-U'\|\cos\phi$. So, we would like to find conditions for which $\cos\phi \leq (1-\gamma)$. By the law of cosines, $\|V1-U'\|^2 = \|V-U'\|^2 + \|V1-V\|^2 - 2\|V-U'\|\|V1-V\|\cos(90+\theta)$ and

$$\begin{aligned}
\cos(\phi) &= \frac{\|V-U'\|^2 + \|V1-U'\|^2 - \|R(V1)-R(V)\|^2}{2\|V-U'\|\|V1-U'\|} \\
&= \frac{\|V-U'\| - \|R(V1)-R(V)\|\cos(90+\theta)}{\sqrt{\|V-U'\|^2 + \|R(V1)-R(V)\|^2 - 2\|V-U'\|\|R(V1)-R(V)\|\cos(90+\theta)}} \\
&= \frac{\|V-U'\| + \|R(V1)-R(V)\|\sin\theta}{\sqrt{\|V-U'\|^2 + \|R(V1)-R(V)\|^2 + 2\|V-U'\|\|R(V1)-R(V)\|\sin\theta}}
\end{aligned}$$

Setting $-(1-\gamma) \leq \cos\phi \leq 1-\gamma$ and solving for $\sin\theta$, we see that this is satisfied by

$$\sin\theta \in \frac{(\gamma^2-2\gamma)\|R(V)-U'\|}{\|R(V1)-R(V)\|} \pm \frac{(1-\gamma)\sqrt{(\gamma^2-2\gamma)\|R(V)-U'\|^2 + \|R(V1)-R(V)\|^2}}{\|R(V1)-R(V)\|}$$

To find a set of values for $\cos\theta$ that make the above expression always true, we will consider only γ for which the set of values of $\sin\theta$ includes the origin. This is true for $\gamma \in \left[0, 1-\sqrt{\frac{\|V-U'\|^2}{\|V-U'\|^2+\|R(V1)-R(V)\|^2}}\right]$. Then,

$$\cos^2\theta \geq 1 - \left(\frac{(\gamma^2-2\gamma)\|R(V)-U'\|}{\|R(V1)-R(V)\|} + \frac{(1-\gamma)\sqrt{(\gamma^2-2\gamma)\|R(V)-U'\|^2 + \|R(V1)-R(V)\|^2}}{\|R(V1)-R(V)\|}\right)^2$$

Rearranging, we have

$$\begin{aligned}
&\|R(V1)-R(V)\|^2\cos^2\theta \geq \\
&\|R(V1)-R(V)\|^2 - \left((\gamma^2-2\gamma)\|R(V)-U'\| + (1-\gamma)\sqrt{(\gamma^2-2\gamma)\|R(V)-U'\|^2 + \|R(V1)-R(V)\|^2}\right)^2 \\
&= (2\gamma-\gamma^2)\cdot\left(\|R(V1)-R(V)\|^2 - \|R(V)-U'\|^2 + \right. \\
&\left. 2\|R(V)-U'\|(1-\gamma)\sqrt{(\gamma^2-2\gamma)\|R(V)-U'\|^2 + \|R(V1)-R(V)\|^2}\right)
\end{aligned}$$

Using the fact that $\|R(V1)-R(V)\|^2 \leq 2$, we upper bound the right-hand side to say that a split satisfying the following condition will guarantee that we decrease the distance from U to the convex hull by $(1-\gamma)$:

$$\begin{aligned}
&\|R(V1)-R(V)\|\cos\theta \geq \\
&(2\gamma-\gamma^2)^{\frac{1}{2}}\left(2 - \|R(V)-U'\|^2 + 2\|R(V)-U'\|(1-\gamma)\sqrt{(\gamma^2-2\gamma)\|R(V)-U'\|^2+2}\right)^{\frac{1}{2}} := f(\gamma)
\end{aligned}$$

◀

▶ **Theorem 2** (Learning an $(\alpha+\epsilon)$-Disclosive Proxy)**.** *Fix α, ϵ, suppose $\mathcal{H}$ has finite VC dimension, and suppose $\exists\ \tilde{h}_V^* \in \Delta\mathcal{H}$ that is a feasible solution to Program* (4)*. Then, Algorithm 3 returns a distribution $\tilde{h}_V$ that is an ϵ-optimal solution to Program* (4)*.*

Proof. We begin by upper bounding the L_2 norm of the gradient:

$$\begin{aligned}\|\nabla_\lambda L(\lambda,\tilde{h}_V)\|^2 = &\left(\sum_{k=1}^{K}\sum_{i=1}^{m}\mathbb{E}_{h_V\sim\tilde{h}_V} w(x_i)h_V(x_i)\left(\mathbb{1}_{z_i=k}-r_k-\alpha\right)\right)^2 + \\ &\left(\sum_{k=1}^{K}\sum_{i=1}^{m}\mathbb{E}_{h_V\sim\tilde{h}_V} w(x_i)(1-h_V(x_i))\left(\mathbb{1}_{z_i=k}-r_k-\alpha\right)\right)^2 + \\ &\left(\sum_{k=1}^{K}\sum_{i=1}^{m}\mathbb{E}_{h_V\sim\tilde{h}_V} w(x_i)h_V(x_i)\left(r_k-\mathbb{1}_{z_i=k}-\alpha\right)\right)^2 + \\ &\left(\sum_{k=1}^{K}\sum_{i=1}^{m}\mathbb{E}_{h_V\sim\tilde{h}_V} w(x_i)(1-h_V(x_i))\left(r_k-\mathbb{1}_{z_i=k}-\alpha\right)\right)^2 \\ &\leq 4K^2m^2\left(1+\alpha\right)^2\end{aligned}$$

We now apply the regret bound for Online Gradient Descent from [45]. With an appropriate choice of η (derived below), we bound the Auditor's average regret over T rounds:

$$\frac{R_T}{T} \leq \frac{\sup_{\lambda,\lambda'\in\Lambda}\|\lambda-\lambda'\|\|\nabla_\lambda L(\tilde{h},\lambda)\|\sqrt{T}}{T} \leq \frac{2Km\left(1+\alpha\right)\sup_{\lambda,\lambda'\in\Lambda}\|\lambda-\lambda'\|}{\sqrt{T}}$$

Setting $T \geq \left(\frac{2Km(1+\alpha)\sup_{\lambda,\lambda'\in\Lambda}\|\lambda-\lambda'\|}{\epsilon}\right)^2$ and $\eta = \frac{\sup_{\lambda,\lambda'\in\Lambda}\|\lambda-\lambda'\|}{2Km(1+\alpha)\sqrt{T}}$, we have that $\frac{R_T}{T} \leq \epsilon$. Because the Learner plays a no-regret strategy, we can apply Theorem (1) to assert that the mixed strategy of the Auditor and Learner together form an ϵ-approximate equilibrium. Next, we must show that an approximate solution to the game corresponds to an approximate solution to Program (4). We will show this using two cases. In the first case, we consider some $\tilde{h}_V^*$ that is a feasible solution to Program (4) at vertex V and a $\hat{\lambda}$ that is an $\epsilon-$approximate minimax solution to the Lagrangian game specified in the Lagrangian above. Now we will analyze the case in which we have a solution $\tilde{h}_V$ that is an ϵ-approximate solution to the Lagrangian game but is not a feasible solution for Program (4) – we will show that this is impossible. To illustrate this, assume that we *do* have such a $\tilde{h}_V$. Because it is not a feasible solution for Program (4), some constraints must be violated. Let ξ be the magnitude of the violated constraint, and let λ be such that the dual variable for the violated constraint is set to $\lambda_{max} := \sup_{\lambda,\lambda'\in\Lambda}\|\lambda-\lambda'\|$. By definition of an ϵ-approximate minimax solution, we know that $L(\hat{\lambda},\tilde{h}_V) \geq L(\lambda,\tilde{h}_V) \geq \mathbb{E}_{h_V\sim\tilde{h}_V}\sum_{i=1}^{m} w_V h_V(x_i)\sum_{k=1}^{K}\mathbb{1}_{z_i=k}\left(U_k'-U_k\right) + \lambda_{max}\xi - \epsilon$. Then,

$$\begin{aligned}&\mathbb{E}_{h_V\sim\tilde{h}_V}\sum_{i=1}^{m} w_V h_V(x_i)\sum_{k=1}^{K}\mathbb{1}_{z_i=k}\left(U_k'-U_k\right) + \lambda_{max}\xi \\ &\leq L(\tilde{h}_V,\hat{\lambda}) + \epsilon \leq L(\tilde{h}_V^*,\hat{\lambda}) + 2\epsilon \leq \mathbb{E}_{h_V\sim\tilde{h}_V^*}\sum_{i=1}^{m} w_V h_V(x_i)\sum_{k=1}^{K}\mathbb{1}_{z_i=k}\left(U_k'-U_k\right) + 2\epsilon\end{aligned}$$

Finally, because $\mathbb{E}_{h_V\sim\tilde{h}_V^*}\sum_{i=1}^{m} w_V h_V(x_i)\sum_{k=1}^{K}\mathbb{1}_{z_i=k}\left(U_k'-U_k\right) \leq \frac{m(K-1)}{K}$, we have that $\lambda_{max}\xi \geq \frac{m(K-1)}{K} + 2\epsilon$. Therefore, the maximum constraint violation is no more than $\frac{\frac{m(K-1)}{K}+2\epsilon}{\lambda_{max}}$. Setting $\lambda_{max} = \frac{m(K-1)}{K\epsilon_\alpha} + 2$, $\tilde{h}_V$ does not violate any constraint by more than ϵ. ◀

▶ **Theorem 4** (Generalization). *Let $\epsilon, \delta, \gamma > 0$ and G be the proxy class. Let there be K sensitive groups. If each proxy group has at least $\frac{1}{2\epsilon^2} \ln \frac{8K \cdot VC(\mathcal{G})\left(\ln \beta - \ln \sqrt{2}\right)}{\delta \ln (1-\gamma)}$ samples, with probability $1-\delta$, an (α, β) proxy in-sample will be an $(\alpha + 2\epsilon, \beta + K\epsilon\sqrt{\frac{\ln \beta - \ln \sqrt{2}}{\ln (1-\gamma)}})$ proxy out-of-sample.*

Proof. Let $\tilde{A}_{k,j} = \Pr_{(x,y,z)\sim\mathcal{P}}[z = k, g(x) = j]$ and $A_{i,j} = \frac{1}{n}\sum_{i=1}^{n} w_j(x_i)\mathbb{1}_{z_i=k, g(x_i)=j}$. Then, Hoeffding's inequality gives us that, for fixed k, j, g

$$\Pr_{D\sim\Omega}\left[\left|\frac{1}{n}\sum_{i=1}^{n} w_j(x_i)\mathbb{1}_{z_i=k,g(x_i)=j} - \mathbb{E}_{(x,y,z)\sim\mathcal{P}}[\mathbb{1}_{z_i=k,g(x_i)=j}]\right| > \epsilon\right] \leq 2e^{-2\epsilon^2 n}$$

Recall from Theorem 3, our decision tree proxy will contain at most $\frac{\ln \gamma - \ln \sqrt{2}}{\ln (1-\gamma)}$ splits, and therefore there will be at most $\frac{\ln \gamma - \ln \sqrt{2}}{\ln (1-\gamma)}$ unique proxy groups. Applying a union bound over all k, j pairs and fixed g, we see that

$$\Pr_{D\sim\Omega}\left[\cap_{k,j}\mathbb{1}_{\left|\frac{1}{n}\sum_{i=1}^{n} w_j(x_i)\mathbb{1}_{z_i=k,g(x_i)=j} - \mathbb{E}_{(x,y,z)\sim\mathcal{P}}[\mathbb{1}_{z_i=k,g(x_i)=j}]\right| > \epsilon}\right] \leq 2K\frac{\ln \beta - \ln \sqrt{2}}{\ln (1-\gamma)}e^{-2\epsilon^2 n}$$

Again applying a union bound, this time over the model class g – with VC dimension d – as well as k, j pairs, we see that for all k, j, g,

$$\Pr_{D\sim\Omega}\left[\cap_{k,j}\mathbb{1}_{\left|\frac{1}{n}\sum_{i=1}^{n} w_j(x_i)\mathbb{1}_{z_i=k,g(x_i)=j} - \mathbb{E}_{(x,y,z)\sim\mathcal{P}}[\mathbb{1}_{z_i=k,g(x_i)=j}]\right| > \epsilon}\right] \leq 2dK\frac{\ln \beta - \ln \sqrt{2}}{\ln (1-\gamma)}e^{-2\epsilon^2 n}$$

Setting this to be less than $\frac{\delta}{3}$, we obtain $2dK\frac{\ln \beta - \ln \sqrt{2}}{\ln (1-\gamma)}e^{-2\epsilon^2 n} \leq \frac{\delta}{3}$, which implies $n \geq \frac{1}{2\epsilon^2}\ln\frac{6dk\left(\ln \beta - \ln \sqrt{2}\right)}{\delta \ln (1-\gamma)}$. Then, with probability $1 - \frac{\delta}{3}$

$$\left\|\left(A - \tilde{A}\right)\rho\right\|_2 \leq \sqrt{\sum_{j=1}^{J}\left(\sum_{k=1}^{K}\left(A_{k,j} - \tilde{A}_{k,j}\right)\cdot\rho_j\right)^2} < \sqrt{\sum_{j=1}^{J}(K\epsilon\rho_j)^2} \leq K\epsilon\sqrt{\frac{\ln \beta - \ln \sqrt{2}}{\ln (1-\gamma)}}$$

This bounds the degradation we expect in balance when we apply the proxy out of sample. Next, we consider the degradation in disclosiveness, which will depend on our estimates of $\Pr_{z\sim\mathcal{P}_z}[z = k]$ and $\Pr_{z|x\sim\mathcal{P}_{z|x}}[z = k|g(x) = j]$. First, we bound the empirical estimate of $\Pr_{z\sim\mathcal{P}_z}[z = k]$. Applying Hoeffding's inequality gives $\Pr_{D\sim\Omega}\left[\left|\frac{1}{n}\sum_{i=1}^{n}\mathbb{1}_{z_i=k} - \mathbb{E}_{z\sim\mathcal{P}_z}[\mathbb{1}_{z_i=k}]\right| > \epsilon\right] \leq 2e^{-2\epsilon^2 n}$. Applying a union bound over the range of Z gives $\Pr_{D\sim\Omega}\left[\cap_{i=1}^{k}\left|\frac{1}{n}\sum_{l=1}^{n}\mathbb{1}_{z_i=k} - \mathbb{E}_{z\sim\mathcal{P}_z}[\mathbb{1}_{z_i=k}]\right| > \epsilon\right] \leq 2Ke^{-2\epsilon^2 n}$. Setting this to be less than $\frac{\delta}{3}$ gives us: $2Ke^{-2\epsilon^2 n} \leq \frac{\delta}{3} \implies n \geq \frac{1}{2\epsilon^2}\ln\frac{6K}{\delta}$.

Finally, repeating the exercise for $\Pr_{z|x\sim\mathcal{P}_{z|x}}[z|g(x)]$, we have that for fixed $g \in \mathcal{G}$ and $z \in Z$,

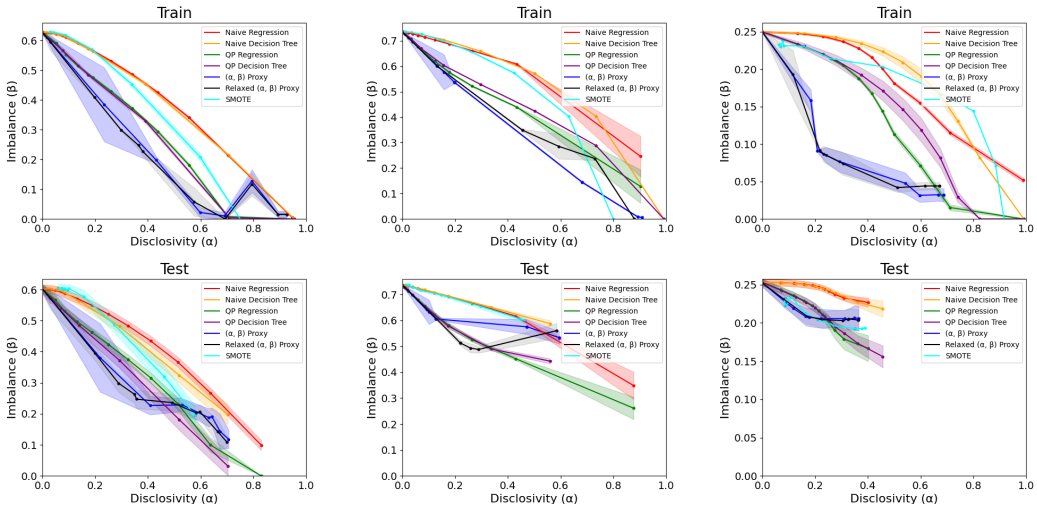
$$\Pr_{D\sim\Omega}\left[\left|\sum_{i=1}^{n}\frac{w_j(x_i)}{\sum_{i=1}^{n} w_j(x_i)}\mathbb{1}_{z_i=k|g(x_i)=j} - \sum_{i=1}^{n} w_j(x_i)\mathbb{E}_{z|x\sim\mathcal{P}_{z|x}}[\mathbb{1}_{z_i=k|g(x_i)=j}]\right| > \epsilon\right] \leq 2e^{-2\epsilon^2\sum_{i=1}^{n} w_j(x_i)}$$

Applying a union bound over z, g, and the VC dimension of $\mathcal{G}$, and setting the probability to be less than $\frac{\delta}{3}$ gives us $2dK\frac{\ln \beta - \ln \sqrt{2}}{\ln (1-\gamma)}e^{-2\epsilon^2\sum_{i=1}^{n} w_j(x_i)} \leq \frac{\delta}{3}$, which implies $\sum_{i=1}^{n} w_j(x_i) \geq \frac{1}{2\epsilon^2}\ln\frac{6dK\left(\ln \beta - \ln \sqrt{2}\right)}{\delta \ln (1-\gamma)}$. Then, with probability $1-\delta$ both of our estimates for $\Pr_{z|x\sim\mathcal{P}_{z|x}}[z|g(x)]$ and $\Pr_{z\sim\mathcal{P}_z}[z]$ must be within ϵ of the true parameters if we have sample

count $n \geq \frac{1}{2\epsilon^2} \max\{\ln \frac{6K}{\delta}, \ln \frac{6dK\left(\ln\gamma - \ln\sqrt{2}\right)}{\delta \ln(1-\gamma)}\}$. Note that $\max\{\ln \frac{6K}{\delta}, \ln \frac{6dK\left(\ln\gamma - \ln\sqrt{2}\right)}{\delta \ln(1-\gamma)}\} \leq \ln \frac{6dK\left(\ln\gamma - \ln\sqrt{2}\right)}{\delta \ln(1-\gamma)}$. Then, taking $n \geq \frac{1}{2\epsilon^2} \ln \frac{6dK\left(\ln\gamma - \ln\sqrt{2}\right)}{\delta \ln(1-\gamma)}$ suffices. Finally, we can apply our concentration bounds to the expression for disclosure level. If we obtain an α-disclosive proxy in-sample, this is equivalent to satisfying, for all $z \in Z$ and $g \in \mathcal{G}$, $|\Pr_{z|x\sim D}[z|g(x)] - \Pr_{z\sim D}[z]| \leq \alpha \implies |\Pr_{z|x\sim \mathcal{P}_{z|x}}[z|g(x)] - \Pr_{z\sim \mathcal{P}_z}[z]| \leq \alpha + 2\epsilon$ ◀

B Additional Experimental Details

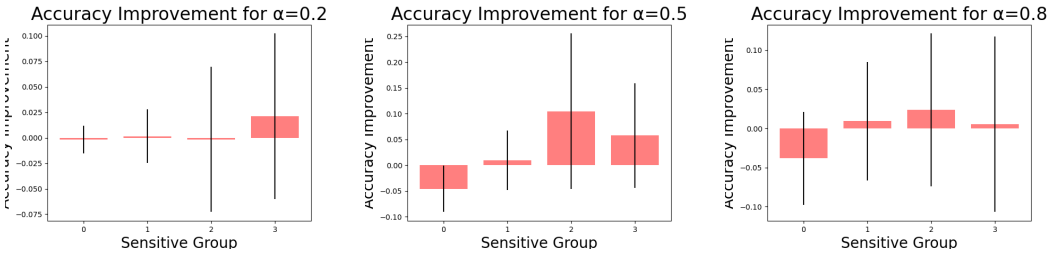
In Figure 2, we show trade-off curves for balance and disclosure when using XGB as the base model. In these plots we also explore a slight relaxation of our (α, β) Proxy, in which we remove the constraint $\sum_i q_i = 1$ when solving $\min_q \|qA - U\|_2$ subject to $q_i \geq 0 \; \forall i$. We find that both the original and relaxed version perform similarly. On Communities, there is less stability displayed by the proxies trained with the XGB base model compared to those trained with the PRC base model. On Adult, the proxies trained with XGB as a base model exhibit a smoother trade-off curve. On the Marketing data set, our proxy approach dominates in sample for smaller levels of α but struggles to generalize.



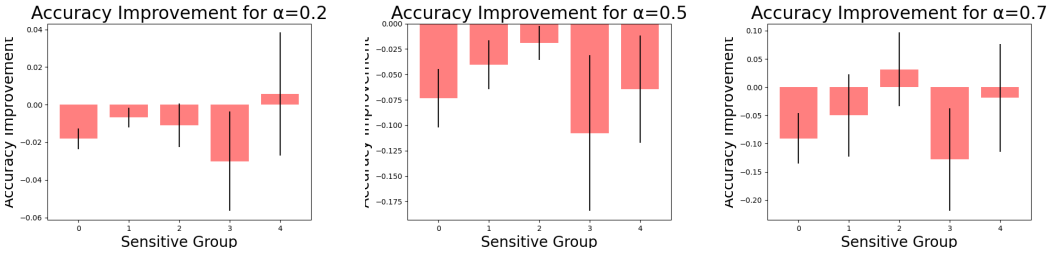
(a) Communities. **(b)** Adult. **(c)** Marketing.

Figure 2 Trade-off of Disclosure and Balance for Proxy Models on the Communities, Adult, and Marketing data sets with XGBoost Base Model.

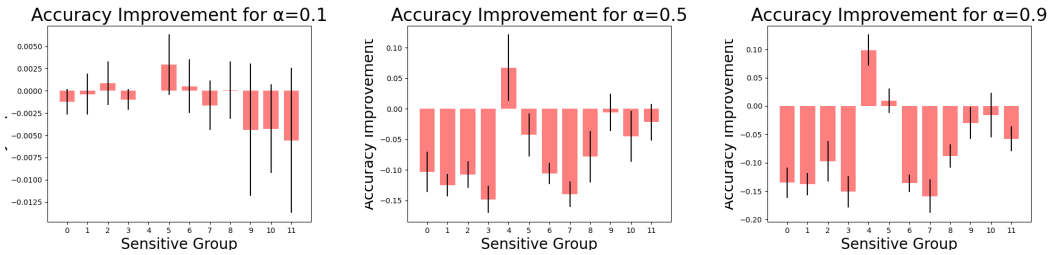
Next, we analyze the downstream fairness impact resulting from using our proxy filtering approach to prepare machine learning training datasets. Here, we train a model for the data-specific classification or regression task on an unfiltered sample and a filtered sample of the same size, and we compare the differences in gropup-wise accuracy obtained by each model. We consider the downstream fairness impact of training a model on data that has been filtered by our (α, β) proxy function but find our results are inconclusive. While we are able to theoretically guarantee a certain level of balance in the filtered data set, we cannot guarantee that the distribution over features and labels will not be skewed in the filtered set, nor can we guarantee that the distribution over features and labels given sensitive attributes will not be distorted. To test this, we first use an (α, β) proxy with a specified α budget to filter the Test set into a balanced sub-sample. Then, we train two model for the dataset specific classification task, one on the filtered data, and the other on a down-sampled version



(a) Communities.



(b) Adult.



(c) Marketing.

Figure 3 Difference in accuracy between models trained on filtered and unfiltered data on the Communities, Adult, and Marketing data sets with PRC base model.

of the original Test set of the same size. We calculate the accuracy of the models on each sensitive group and then plot the *difference* in accuracy between the two models, calculated as the group accuracy on the filtered data minus the group accuracy on the unfiltered data. Thus, positive values indicate an improvement in group accuracy from training on the filtered data, while negative values indicate a decrease. Between the three data sets, we see mixed results, displayed in Figure 3. On the Communities data set, we broadly see improvement on lower accuracy groups when using the model trained on the filtered data. However, results from the Adult data set in show a decrease in performance across all groups, and results from the Marketing data set show improvement for one of the least represented groups, but a decrease in performance for most others.