On the Compatibility of Privacy and Fairness

Rachel Cummings* Varun Gupta* Dhamma Kimpara* Jamie Morgenstern*

March 13, 2019

Abstract

In this work, we investigate whether privacy and fairness can be simultaneously achieved by a single classifier in several different models. Some of the earliest work on fairness in algorithm design defined fairness as a guarantee of similar outputs for "similar" input data, a notion with tight technical connections to differential privacy. We study whether tensions exist between differential privacy and statistical notions of fairness, namely Equality of False Positives and Equality of False Negatives (EFP/EFN). We show that even under full distributional access, there are cases where the constraint of differential privacy precludes exact EFP/EFN. We then turn to ask whether one can learn a differentially private classifier which approximately satisfies EFP/EFN, and show the existence of a PAC learner which is private and approximately fair with high probability. We conclude by giving an efficient algorithm for classification that maintains utility and satisfies both privacy and approximate fairness with high probability.

1 Introduction

The growing practice of applying machine learning techniques on personal data has raised concerns that too much of some individual's information might be leaked through a model learned on training data. For example, if one learns a model based on historical health data, there is a risk that the algorithm will produce a model containing information about individuals' disease status or other sensitive information. Machine learning algorithms for word prediction have been shown to leak Social Security Numbers and credit card numbers when trained on a corpus that included such data [Carlini et al., 2018]. As a result, academia and industry have spent much effort designing and implementing *differentially private* machine learning methods. Differential privacy gives a strong guarantee to individuals whose data we use to train a model: the model will learn aggregate information about the population, but will not encode information about the individuals.

Recent human-centric uses of machine learning systems for applications such as loan approvals and predictive policing have also raised concerns of equity of the predictive power for a model on different populations. The *fairness* of a model can be thought of as an equitable performance guarantee for individuals who will be evaluated by the model, rather than a guarantee for someone participating in the model's training process. When phrasing privacy and fairness of a model this way, a natural question arises: in what settings can we learn a model which is private in the training data and guarantees equitable performance for multiple populations on which the model will ultimately be deployed? More precisely, will it be possible to guarantee privacy of training data, fairness for the predictions made on two populations, and reasonable accuracy overall? Our work addresses these questions.

1.1 Our Contributions

We present three contributions to the study of the intersection of differential privacy and fairness in classification. First, we show that it is impossible to achieve both differential privacy and exact fairness while maintaining non-trivial accuracy, even in the setting where we have access to the full distribution of the

*Georgia Institute of Technology. Email: {rachelc, varungupta, dkimpara1, jamiemmt.cs}@gatech.edu. R.C. supported in part by a Mozilla Research Grant.

data. We then consider a notion of approximate fairness in the finite sample access setting and show that there exists a private PAC learner that is differentially private and satisfies approximate fairness with high probability. Finally, we give a polynomial time no-regret algorithm that returns an accurate classifier which satisfies both privacy and approximate fairness with high probability.

1.2 Related Work

The focus on fairness in machine learning and its relationship to differential privacy was explored in early work by the privacy community [Dwork et al., 2012]. This work introduced the concept of treating similar people similarly, where "similarity" is defined as a task-specific metric over individuals. The authors then point out that this desideratum can be formulated as a Lipschitz constraint and show how to satisfy it using tools from differential privacy.

Ekstrand et al. [2018] raised questions of whether statistical notions of equitable predictive power, such as equalized odds [Hardt et al., 2016], are compatible with privacy. In a limiting sense, when one talks about feature and model selection, there appears to be some tension here: an additional feature might increase the possible privacy loss of an individual, while the additional feature should only make equalized odds easier to satisfy [Bird et al., 2016]. Recent work at the interface of privacy and fairness has also advocated that these two societal concerns be studied together [Datta et al., 2018]. [VG: However, to the best of the authors' knowledge, no work has heretofore shown whether a machine learning algorithm can guarantee individuals differential privacy and their populations group fairness simultaneously.] [VG: This area remains largely unexplored, but initial work by Jagielski et al. [2019] shows two algorithms that satisfy both differential privacy and equalized odds. To the best of the author's knowledge, [Jagielski et al., 2019] is the only work to date that has considered satisfying both differential privacy and fairness constraints.]

The technical tools we use for this work come from differential privacy, including the exponential mechanism [McSherry and Talwar, 2007] and a differentially private version of Follow the Perturbed Leader [Kalai and Vempala, 2005, Kearns et al., 2014, 2018]. Similar tools were used to satisfy statistical parity across many different group definitions simultaneously in Kearns et al. [2018], although their algorithms were not differentially private. More recently, differentially private tools contributed to the design of algorithms which are well-calibrated across many different group definitions simultaneously [Hebert-Johnson et al., 2018].

2 Preliminaries

Let $\mathcal{X}$ be a data universe consisting of elements of the form $z = (x, a, y)$ where x are the element's features, a is a protected (binary) attribute, and y is a binary label. As a concrete example, consider loan applications, where x may be an applicant's income and credit score, a may indicate whether the applicant is a racial minority, and y may indicate whether the applicant intends to repay her loan. We will assume that data entries are drawn from a joint distribution over $\mathcal{X}$. We allow x to be arbitrarily correlated with a, including containing a copy of a.

We will use the notation $\mathcal{A}(D)$ to denote the probability distribution over outputs of a randomized algorithm $\mathcal{A}$ on input D.

2.1 Differential Privacy

Differential privacy ensures that a randomized algorithm will generate similar distributions over outputs on *neighboring databases*, which differ in a single entry. For our results, we require two different notions of a database, each with an accompanying definition of neighbors:

1. a finite sample, vector $Z = (z_1, ..., z_n)$ with entries drawn i.i.d. from a distribution D over $\mathcal{X}$,
2. a distribution D over $\mathcal{X}$.

The first notion is standard in the privacy literature, where databases are viewed as a finite collection of data points from n individuals. The second notion is standard for statistical notions of fairness, where the goal is to ensure fairness over a large—possibly infinite-sized—population.

For the first notion, we can use the standard definition of neighboring databases, in which one element of the sample is changed.

Definition 1 (Neighboring samples)**.** Samples Z and Z' are neighboring if $z_i \neq z'_i$ for exactly one $i \in [n]$.

For the second database notion, we no longer have a finite size database with discrete entries, so the concept of changing a single entry is no longer well-defined. Instead we use a closeness measure over distributions to define neighboring databases.

Definition 2 (ζ-closeness [McGregor et al., 2010])**.** Random variables D and D' taking values in $\mathcal{X}$ are ζ-close if the statistical distance between their distributions is at most ζ, i.e.,

$$\|D - D'\|_{SD} := \frac{1}{2} \sum_{z \in \mathcal{X}} \left| \Pr[D = z] - \Pr[D' = z] \right| \leq \zeta.$$

When we view databases as distributions over $\mathcal{X}$, we will say that D and D' are neighbors if they are ζ-close for some specified ζ.

Note that we could instead define a finite sample as a multi-set of sampled points, and use symmetric distance instead of Hamming distance to measure finite sample neighbors. In this case, Definition 2 can simply be considered a generalization of Definition 1.

The definition of differential privacy remains the same under both database notions. That is, a randomized algorithm is differentially private if neighboring databases induce close distributions over outputs.

Definition 3 (Differential privacy [Dwork et al., 2006])**.** A randomized algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private if for all pairs of neighboring databases D, D' and for all sets $\mathcal{S} \in \mathit{Range}(\mathcal{A})$ of outputs,

$$\Pr[\mathcal{A}(D) \in \mathcal{S}] \leq \exp(\epsilon) \Pr[\mathcal{A}(D') \in \mathcal{S}] + \delta.$$

If $\delta = 0$, we say that $\mathcal{A}$ is ϵ-differentially private.

A survey of relevant differentially private tools and algorithmic properties is given in Appendix A.

2.2 Exact Fairness

When considering exact fairness, we use the second database notion presented in Section 2.1, where we consider a database to be a distribution D over $\mathcal{X}$. This corresponds to the full distributional access setting studied in Hardt et al. [2016], in which an algorithm has access to the joint probability distribution for $D = (X, A, Y)$ taking values in $\mathcal{X}$, where X, A, and Y are random variables respectively denoting features, sensitive attributes, and true labels. We will consider binary predictors $h : \mathcal{X} \to \{0, 1\}$, which attempts to predict the true label y of a point, based upon (x, a).[1]

We use *equal opportunity* as our notion of exact fairness in this setting.

Definition 4 (Equal Opportunity [Hardt et al., 2016])**.** A binary predictor h satisfies equal opportunity with respect to A and Y if

$$\Pr_D[h = 1 | Y = 1, A = 1] = \Pr_D[h = 1 | Y = 1, A = 0].$$

[1]Although we write $h : \mathcal{X} \to \{0, 1\}$ or $h(z)$ for ease of notation, it should be understood that $h(z)$ depends only on the observable attributes x and a. If a hypothesis could condition its predicted label on the true label, then perfect prediction would always be possible and the learning problem would be trivial.

This fairness definition requires equality of *group-conditional true positive classification rates*, denoted

$$\gamma_{ya}(h) := \Pr[h = 1 | Y = y, A = a].$$

To ensure $\gamma_{ya}(h)$ is always well-defined, we introduce *group membership probabilities* $P_{ya} = \Pr[Y = y, A = a]$ and assume throughout the paper that $P_{ya} > 0$ for $a, y \in \{0, 1\}$.

Equal opportunity is also known as Equality of False Negatives (EFN), because it requires that the false positive rates be equal on groups with $a = 0$ and $a = 1$. A strictly stronger condition would be to additionally require Equality of False Positives (EFP). The requirement of both EFN and EFP is also known as *equalized odds* [Hardt et al., 2016], which is a strictly stronger fairness requirement than equal opportunity. Since we will prove impossibility results for exact fairness, it only strengthens our results to consider this weaker fairness constraint. These measures of fairness belong to a broader class of fairness constraints that can be represented as constraints on functions of the confusion matrix of a classifier [Narasimhan, 2018].

2.3 Approximate Fairness

In our setting for approximate fairness, we use the first database notion presented in Section 2.1, where we consider a database to be a finite sample Z consisting of entries drawn i.i.d. from a distribution D over $\mathcal{X}$.

Considering approximate fairness is appropriate for two reasons. First, when learning a classifier from a finite sample, achieving exact fairness is impossible: a finite sample necessarily implies some error in estimating any statistic. Second, as we will show in Section 3, exact fairness is incompatible with differential privacy. Thus approximate notions of fairness are *necessary* relaxations if one hopes to satisfy any fairness and privacy criteria from a finite sample.

For ease of notation, we define subgroups in the database based upon group membership and true labeling as follows:

$$Z_{ya} := \{z_i \in Z | y_i = y, a_i = a\}.$$

We define group-conditional true positive rates analogously to the distributional setting in Section 2.2:

$$\gamma^Z_{ya}(h) := \frac{1}{|Z_{ya}|} \sum_{z_i \in Z_{ya}} h(z_i).$$

We use α-*discrimination* as our notion of approximate fairness, which requires that group-conditional true positive rates are not different by more that α.

Definition 5 (α-discrimination [Woodworth et al., 2017])**.** A binary predictor h is α-discriminatory with respect to a binary protected attribute A on a sample Z if,[2]

$$\Gamma^Z(h) := \max_{y \in \{0,1\}} |\gamma^Z_{y0}(h) - \gamma^Z_{y1}(h)| \leq \alpha.$$

A discrimination parameter of $\alpha = 0$ corresponds to exact fairness. As an analog of Definition 4, we only consider true positive rates, and approximate non-discrimination reduces to the condition:

$$\Gamma^Z(h) := |\gamma^Z_{10}(h) - \gamma^Z_{11}(h)| \leq \alpha.$$

For the remainder of this paper, the notation Γ refers to this fairness measure.

2.4 Agnostic PAC Learning

Our learning models consider the agnostic setting that removes the *realizability assumption* that a perfect hypothesis exists in the class $\mathcal{H}$. If no such perfect hypothesis exists, then the goal of a learning algorithm

[2] α-discrimination can also be defined on a population, defined by the condition $\Gamma(h) := \max_{y \in \{0,1\}} |\gamma_{y0}(h) - \gamma_{y1}(h)| \leq \alpha$. By Woodworth et al. [2017], the sample fairness measure Γ^Z converges to the population measure Γ when n is large.

should be to minimize prediction error. Formally, the goal of a learner in the agnostic setting is to output a (possibly randomized) hypothesis $h \in \mathcal{H}$ whose error with respect to the distribution is close to the optimal possible by any function from $\mathcal{H}$. The misclassification error of h on D is defined as:

$$err(h) = \Pr_{z \sim D}[h(z) \neq y].$$

In our setting we consider only binary labels, and therefore consider randomized hypotheses $h : \mathcal{X} \to [0, 1]$, where $h(z)$ is the probability that h predicts 1 on example z.

Definition 6 (Agnostic PAC Learning [Chaudhuri et al., 2011])**.** A hypothesis class $\mathcal{H}$ is agnostically PAC learnable if there exists a polynomial $poly(\cdot, \cdot, \cdot)$ and a learning algorithm $\mathcal{A}$ with the following property: for every $\alpha, \beta \in (0, 1)$ and for every distribution D on $\mathcal{X}$, when running the learning algorithm on $n \geq poly(1/\alpha, \log(1/\beta), |\mathcal{H}|)$ i.i.d. examples generated by D, the algorithm $\mathcal{A}$ returns a hypothesis h such that,

$$\Pr[err(h) \leq OPT + \alpha] \geq 1 - \beta,$$

where $OPT = \min_{h \in \mathcal{H}} err(h)$ and the probability is over the choice of n training examples. The learning algorithm $\mathcal{A}$ is said to be (α, β)-accurate.

We define private and approximately fair PAC learners as algorithms that satisfy the definitions of differential privacy, approximate fairness (with high probability), and PAC learning.

Definition 7 (Private and Approximately Fair Agnostic PAC Learning)**.** A hypothesis class $\mathcal{H}$ is privately and approximately fair agnostically PAC learnable if there exists a polynomial $poly(\cdot, \cdot, \cdot, \cdot, \cdot)$ and a learning algorithm $\mathcal{A}$ with the following property: for every $\alpha, \beta \in (0, 1)$, $\epsilon > 0$, $P_{min} = \min_{a \in \{0,1\}} P_{1a}$, and for every distribution D on $\mathcal{X}$, when running the learning algorithm on $n \geq poly(1/\epsilon, 1/P_{min}, 1/\alpha, \log(1/\beta), |\mathcal{H}|)$ i.i.d. examples generated by D:

1. [Fairness and Accuracy] Algorithm $\mathcal{A}$ satisfies $\Pr[\Gamma(h) + err(h) \leq OPT + \alpha] \geq 1 - \beta$;
2. [Privacy] [DK: changed to (ϵ, δ)-DP to reflect change in approx alg] Algorithm $\mathcal{A}$ is (ϵ, δ)-differentially private;

where $OPT = \min_{h \in \mathcal{H}} \Gamma(h) + err(h)$ and the probability is over the choice of n training examples. The learning algorithm $\mathcal{A}$ is both (α, β)-accurate and is at most α-discriminatory with probability at least $1 - \beta$.

By Woodworth et al. [2017], the dependence of the sample complexity on P_{1a} is unavoidable for our definition of approximate fairness. If there is a group with low prevalence in the sample, we still need enough samples from that group to ensure that the sample group-conditional true positive rates γ_{1a}^{Z} generalize to the population.

3 Exact Fairness with Differential Privacy is Impossible

In this section, we see that exact fairness and differential privacy are strong guarantees that together prove to be incompatible. Since we will provide impossibility results, we consider the simplest possible task. We grant our learning algorithm full distributional access to the underlying population, and ask only for non-trivial classification accuracy. That is, better than any *constant classifier* which predicts the same label for all points: $h(z_0) = h(z_1)$ for all $z_0, z_1 \in \mathcal{X}$.

Our main result of this section shows that it is impossible for a classifier with non-trivial accuracy to simultaneously achieve exact fairness and differential privacy.

Theorem 1. For any hypothesis class $\mathcal{H}$, no algorithm can simultaneously satisfy $(\epsilon, 0)$-differential privacy for $\epsilon < \infty$ and guarantee to output a hypothesis $h \in \mathcal{H}$ that satisfies equal opportunity and has error less than that of any constant classifier.

Proof. We prove the theorem by first constructing a simple distribution D and a neighboring (i.e., ζ-close for arbitrarily small $\zeta > 0$) distribution D'. We then show that any non-trivial hypothesis h which is fair with respect to D is not fair with respect to D'. Since D and D' are neighboring, any differentially private algorithm must output h with approximately the same probability under D and D'. Therefore, no differentially private algorithm can produce an exactly fair hypothesis with non-trivial accuracy.

We begin by constructing D as the uniform distribution over the following four elements:

$$z_0 = (x_0, 0, 0); z_1 = (x_1, 0, 1);$$
$$z_2 = (x_2, 1, 0); z_3 = (x_3, 1, 1).$$

That is, under distribution D, protected attributes and labels are equally likely to be 0 or 1, and x, a, and y are all independent. Thus any constant classifier will have error 1/2.

Next consider an arbitrary $h \in \mathcal{H}$ such that $h(z_1) + h(z_3) > h(z_0) + h(z_2)$ and $h(z_1) = h(z_3)$. The first condition ensures that $err(h) < 1/2$, and the second condition ensures that h satisfies equal opportunity. Note that there must exist such an $h \in \mathcal{H}$. Otherwise no algorithm can produce a fair hypothesis with accuracy better than a constant classifier, because no such hypothesis exists, and we are done. Consider an algorithm $\mathcal{A}$ that always outputs a hypothesis that satisfies equal opportunity and has non-trivial accuracy, and assume without loss of generality that $\Pr[\mathcal{A}(D) = h] > 0$.

We next construct a neighboring distribution D' that is ζ-close to D for arbitrarily small $\zeta > 0$, and show that $\Pr[\mathcal{A}(D') = h] = 0$. Define D' to place an additional ζ probability mass on z_3 relative to D, and ζ less mass on z_1. Then D and D' are ζ-close under Definition 2, and are neighbors for any arbitrarily small $\zeta > 0$. However, h does not satisfy equal opportunity with respect to D' because:

$$\gamma_{10}^{D'} = \frac{1}{4} - \zeta \neq \frac{1}{4} + \zeta = \gamma_{11}^{D'}.$$

Therefore algorithm $\mathcal{A}$ cannot output h with positive probability on input D'.

Finally, there is no $\epsilon < \infty$ for which,

$$0 < \Pr[\mathcal{A}(D) = h] \leq \exp(\epsilon) \Pr[\mathcal{A}(D') = h] = 0,$$

so $\mathcal{A}$ cannot be differentially private for any finite ϵ.

□

Our result also has implications for more general notions of statistical fairness. Most immediately, Theorem 1 implies that no algorithm can simultaneously satisfy privacy, accuracy, and equalized odds [Hardt et al., 2016], which is a strictly stronger fairness condition than equal opportunity.

More generally, the same proof technique can be used to show impossibility of achieving privacy with other notions of approximate fairness such as disparate impact and mean difference scores [Narasimhan, 2018]. Under these definitions, approximate fairness means that the resulting classifier (or anything released by the randomized algorithm) satisfies some hard threshold on the 'level of discrimination' on the distribution. One simply needs to construct neighboring distributions where most or all of the fair hypotheses h which satisfy the hard threshold on one distribution will fail to satisfy the threshold on a neighboring distribution.

At the crux of these impossibility results is the constraint that some hypothesis h which is fair on database D and may be output with positive probability under $\mathcal{A}(D)$, but cannot be output under a neighboring database D'. The differential privacy constraint then implies that h cannot be output under D either. One might consider relaxing to (ϵ, δ)-differential privacy, which allows for some failure probability δ of the privacy guarantee, and would allow h to be output with probability $\delta > 0$ under distribution D. However, in most privacy applications, δ is typically required to be cryptographically small which would still not qualitatively circumvent these impossibility results.

In the next section, we instead relax the requirement that the hypothesis is always fair on the input database, and consider algorithms that are both approximately fair with high probability and differentially private.

4 Approximate Fairness with Differential Privacy

In the previous section we sought impossibility results, so we considered the simplest possible learning setting: the algorithm was allowed full distributional access, and the goal was simply non-trivial classification accuracy. We showed that even in that simple setting, privacy and exact fairness are not compatible. Here we relax our fairness constraint to α-*discrimination* (Definition 5) and give positive results for learning under the constraints of privacy and this notion of approximate fairness. As we shift our focus to positive results, we correspondingly consider a more challenging learning environment. In this section, we only allow the algorithm sample access to the distribution, and we ask the algorithm to agnostically PAC learn a hypothesis class.

Recall from Section 2.1 that in the finite sample setting, a database is a vector $Z = (z_1, ..., z_n)$ with entries drawn i.i.d. from a distribution D over $\mathcal{X}$. In this setting, our goal is to release a hypothesis that minimizes error with respect to D and is approximately fair. Our algorithm for this task (Algorithm 1) is an instantiation of the $(\epsilon, 0)$-differentially private Exponential Mechanism [McSherry and Talwar, 2007] to select a hypothesis $h \in \mathcal{H}$. In the following analysis, we assume that our databases contain at least two positively labled instances for each protected attribute. This proves useful in our analysis of differential privacy and is a very reasonable assumption in practice if one hopes to learn an accurate classifier.

The Exponential Mechanism relies on a *utility loss score* $u : \mathcal{X}^n \times \mathcal{H} \to \mathbb{R}$, where $u(Z, h)$ is the utility loss from producing hypothesis h on input database Z. The mechanism then samples an output h with probability exponentially biased by negative loss score, which ensures that a hypothesis with small loss is sampled with high probability.[3]

We wish to optimize both fairness and accuracy, so we incorporate both objectives into our utility function. We use α-discrimination as our in-sample fairness measure, quantified by $\Gamma^Z(h) = |\gamma_{10}^Z(h) - \gamma_{11}^Z(h)|$ for group-conditional true positive rates $\gamma_{10}^Z(h), \gamma_{11}^Z(h)$. We would ideally like to measure accuracy with respect to the underlying distribution D, using accuracy measure $err(h)$, but the algorithm does not have access to D. Instead, we will use the empirical misclassification error $err^Z(h) = \frac{1}{n}\sum_{z_i \in Z} \Pr[h(x_i, a_i) \neq y_i]$, and will later have to reason that $err^Z(h)$ is close to $err(h)$. Therefore Algorithm 1 uses the utility score[4]

$$u(Z, h) = \Gamma^Z(h) + err^Z(h).$$

To instantiate the Exponential Mechanism, we also need to know the *sensitivity* of the utility score, defined as

$$\Delta u = \max_{h \in \mathcal{H}} \max_{Z, Z' \text{neighbors}} |u(Z, h) - u(Z', h)|.$$

The sensitivity of a function is the maximum change in its value from changing one entry in the database. We can analogously define $\Delta\Gamma$ and Δerr as the respective sensitivities of the discrimination level $\Gamma^Z(h)$ and the empirical misclassification error $err^Z(h)$. By Triangle Inequality, it suffices to set $\Delta u = \Delta\Gamma + \Delta err$. Both terms will be bounded in our analysis of Algorithm 1.

Our main result of this section shows that our Approximately Fair Private Learner of Algorithm 1 computes an approximately fair hypothesis with good accuracy in a differentially private manner.

Theorem 2. [DK: added that $\epsilon > \Omega(1/\sqrt{n})$] Any hypothesis class $\mathcal{H}$ is privately and approximately fairly agnostically learnable with (α, β)-accuracy by Algorithm 1 $\mathcal{A}(\mathcal{H}, \cdot, \epsilon)$ with $\epsilon > \Omega(1/\sqrt{n})$ and

$$n \geq 144(\ln|\mathcal{H}| + \ln 1/\beta) \cdot \max_{a \in \{0,1\}} \left(\frac{1}{\alpha^2 P_{1a}}, \frac{1}{\epsilon \alpha P_{1a}} \right)$$

labeled examples drawn i.i.d. from distribution D.

Though our guarantee holds for only the simple sum of the fairness and accuracy scores, one may weight the terms in the sum differently to achieve more preference on either fairness or accuracy. The sample size will still hold up to a constant that depends on the weighting.

[3] The standard Exponential Mechanism of McSherry and Talwar [2007] is designed to sample an output with high utility. We change signs because we wish to minimize loss rather than maximize utility. The two versions are equivalent.

[4] Any weighted combination of the fairness and accuracy terms would suffice. We use the unweighted sum for simplicity.

Algorithm 1 Approximately Fair Private Learner $\mathcal{A}(\mathcal{H}, Z, \epsilon)$

Input:[DK: we need to assume that alg knows n, and P_a, not sure how to include this] hypothesis class $\mathcal{H}$, sample Z, privacy parameter ϵ
if $PTR(Z, \epsilon, \delta) = \perp$ **then**
  Output: $\perp$ and **Halt.**
else
  Set $u(Z, h) = \Gamma^Z(h) + err^Z(h)$ and $\Delta u = \Delta\Gamma + \Delta err$
  Sample hypothesis $h \in \mathcal{H}$ with probability proportional to

$$\exp(-\frac{\epsilon \cdot u(Z, h)}{2\Delta u})$$

  Output: sampled hypothesis h
end if

Algorithm 2 Propose-Test-Release $PTR(Z, \epsilon, \delta)$

Input: sample Z, privacy parameter ϵ
Query $M = \min_{a\in\{0,1\}} |Z_{1a}|/n + \text{Lap}(\frac{1}{n\epsilon})$
if $M \geq \min_{a\in\{0,1\}} P_{1a} + \ln(\frac{1}{\delta})(\frac{1}{n\epsilon})$ **then**
  Output: $\top$
else
  Output: $\perp$
end if

Proof. [DK: edited this to reflect that we now have a δ failure probability of being not ϵ-DP] Algorithm 1 first tests that all sample base rates, $|Z_a|/n$, are greater than the respective true base rates P_a. We then use the true base rate as an upper bound for our sensitivity. Which we can do since

Lemma 1. $\Gamma^Z(h)$ has sensitivity $\Delta\Gamma = \min_{a\in\{0,1\}} 2/(|Z_{1a}| - 1)$, and $err^Z(h)$ has sensitivity $\Delta err = 1/n$.

Specifically, if $\min_{a\in\{0,1\}} 1/|Z_{1a}| \geq \min_{a\in\{0,1\}} nP_{1a}$ then $\min_{a\in\{0,1\}} 2/(|Z_{1a}| - 1) \leq \min_{a\in\{0,1\}} 2/(nP_{1a} - 1)$

If the test fails, then we halt. If it passes then the rest of the algorithm is an instantiation of the Exponential Mechanism, which is known to be ϵ-differentially private [McSherry and Talwar, 2007]. However, our algorithm is not always guaranteed to be ϵ-differentially private since if our test $PTR(Z, \epsilon, \delta)$ outputs a false-positive (i.e. the test passes but $\min_{a\in\{0,1\}} 1/|Z_{1a}| < \min_{a\in\{0,1\}} nP_{1a}$), then the exponential mechanism may not have run with enough noise to ensure ϵ-differential privacy. But, by Lemma 2,

Lemma 2. $PTR(\cdot, \cdot, \delta)$ outputs a false-positive with probability less than or equal to δ. [DK: pf in appendix]

$PTR(\cdot, \cdot, \cdot)$ outputs a false-positive with probability δ and therefore our algorithm is (ϵ, δ)-differentially private. [DK: not sure how to formalize following] Usually, δ should be exponential in n. However in the Propose-Test-Release algorithm, we add $\ln(\frac{1}{\delta})(\frac{1}{n\epsilon})$ to our threshold in order to give us a small false-positive probability. If $\delta = O(2^{-\sqrt{n}})$ then $\ln(\frac{1}{\delta})(\frac{1}{n\epsilon}) = \frac{\ln 2}{\sqrt{n}\epsilon}$. But since $P_{1a} \in (0, 1)$, if $\frac{\ln 2}{\sqrt{n}\epsilon}$ is too large (near 1 or larger than 1) then Propose-Test-Release will always fail with high probability. Hence for our algorithm to output some hypothesis with nontrivial probability, we need that $\epsilon > \Omega(1/\sqrt{n})$.

Now we will show that if the algorithm outputs a hypothesis (and not $\perp$), the utility condition is also satisfied. If the utility loss function $u(Z, h) = \Gamma^Z(h) + err^Z(h)$ used in Algorithm 1 was our desired objective, then we could immediately apply accuracy guarantees of the Exponential Mechanism. However, we actually wish to minimize a slightly different objective: $\Gamma^Z(h) + err(h)$, which we will denote $u(D, h)$. Let the event $E = \{\mathcal{A}(\mathcal{H}, Z, \epsilon) = h \text{ with } u(D, h) > OPT + \alpha\}$. We will show that event E happens with low probability.

We will start by showing that $u(D,h)$ is close to $u(Z,h)$ with high probability. We require Lemma 3 (stated below and proved in Appendix B), which relies on Chernoff-Hoeffding bounds (Theorem 7) and Lemma 4 from Woodworth et al. [2017], both also stated in Appendix B.

Lemma 3 (Concentration of Utility)**.** For any sample Z of size n drawn i.i.d. from distribution D and for any binary predictor h,

$$\Pr\big[\big|u(Z,h) - u(D,h)\big| > \rho\big] \leq 18\exp\left(-\min_a \frac{\rho^2 n P_{1a}}{16}\right).$$

Applying a union bound over all $h \in \mathcal{H}$, Lemma 3 implies that

$$\Pr[|u(Z,h) - u(D,h)| \geq \rho \text{ for some } h \in \mathcal{H}] \leq 18|\mathcal{H}|\exp(-\min_a \frac{\rho^2 n P_{1a}}{16}).$$

Now we analyze $\mathcal{A}(\mathcal{H}, Z, \epsilon)$ conditioned on the event that for all $h \in \mathcal{H}$, $|u(Z,h) - u(D,h)| < \rho$. For every $h \in \mathcal{H}$,

$$\begin{aligned}
\Pr[\mathcal{A}(\mathcal{H}, Z, \epsilon) = h] &= \frac{\exp(-\frac{\epsilon}{2\Delta u} \cdot u(Z,h))}{\sum_{h' \in \mathcal{H}} \exp(-\frac{\epsilon}{2\Delta u} \cdot u(Z,h'))} \\
&\leq \frac{\exp(-\frac{\epsilon}{2\Delta u} \cdot u(Z,h))}{\max_{h' \in \mathcal{H}} \exp(-\frac{\epsilon}{2\Delta u} \cdot u(Z,h'))} \\
&= \exp(-\frac{\epsilon}{2\Delta u}(u(Z,h) - \min_{h' \in \mathcal{H}} u(Z,h'))) \\
&\leq \exp(-\frac{\epsilon}{2\Delta u}(u(Z,h) - (OPT + \rho))).
\end{aligned}$$

Hence the probability that $\mathcal{A}(\mathcal{H}, Z, \epsilon)$ outputs a hypothesis $h \in \mathcal{H}$ such that $u(Z,h) > OPT + 2\rho$ is at most $|\mathcal{H}|\exp(-\frac{\epsilon \cdot \rho}{2\Delta u})$.

Setting $\rho = \alpha/3$, we get the following bound on $\Pr[E]$:

$$\begin{aligned}
\Pr[E] &= \Pr[\mathcal{A}_\epsilon = h \text{ with } u(Z,h) > OPT + \alpha] \\
&\leq \Pr[|u(D,h) - u(Z,h)| \geq \alpha/3] + \Pr[u(Z,h) \geq OPT + 2\alpha/3] \\
&\leq |\mathcal{H}|(18\exp(-\min_a \frac{\alpha^2 n P_{1a}}{144}) + \exp(-\frac{\epsilon \cdot \alpha}{6\Delta u}))
\end{aligned}$$

Then our desired utility guarantee holds for any β satisfying:

$$\beta \geq |\mathcal{H}|(18\exp(-\min_a \frac{\alpha^2 n P_{1a}}{144}) + \exp(-\frac{\epsilon\alpha}{6\Delta u})).$$

To complete the proof and translate the bound on β to a bound on n, we plug in the sensitivity bound

of Lemma 1. Thus the utility guarantee holds for

$$\begin{aligned}
\beta &\geq |\mathcal{H}|(18\exp(-\min_a \frac{\alpha^2 n P_{1a}}{144}) + \exp(-\frac{\epsilon\alpha}{6\Delta u})) \\
\beta &\geq |\mathcal{H}|(19\exp(\max_a(-\frac{\alpha^2 n P_{1a}}{144}, -\frac{\epsilon\alpha}{6\Delta u})) \\
-\ln 1/\beta &\geq \ln|\mathcal{H}| + \ln 19 + \max_a(\frac{\alpha^2 n P_{1a}}{144}, \frac{\epsilon\alpha}{6\Delta u}) \\
-\ln 1/\beta &\geq \ln|\mathcal{H}| + \ln 19 + \max_a(-\frac{\alpha^2 n P_{1a}}{144}, -\frac{\epsilon\alpha}{6(1/|Z_{1a}| + 1/n)}) \\
\max_a(\frac{\alpha^2 n P_{1a}}{144}, \frac{\epsilon\alpha n}{6(n/|Z_{1a}|+1)}) &\geq \ln|\mathcal{H}| + \ln 1/\beta + \ln 19 \\
n &\geq 144(\ln|\mathcal{H}| + \ln 1/\beta) \cdot \max_{a\in\{0,1\}} \left(\frac{1}{\alpha^2 P_{1a}}, \frac{n/|Z_{1a}|+1}{\epsilon\alpha}\right).
\end{aligned}$$

□

5 An Efficient Algorithm for Approximately Fair and Private Classification

In this section, we construct a private version of the Fair No-Regret Dynamics (FairNR) algorithm originally given by Kearns et al. [2018]. This is a polynomial time algorithm that returns an approximately fair and accurate randomized classifier with high probability. Their results depend on a polynomial-time equivalence between cost-sensitive classification (CSC) and agnostic learning, and an equivalence between weak agnostic learning and auditing for fairness. This equivalence is given in Appendix C.

The cost-sensitive classification problem takes as input a hypothesis class $\mathcal{H}$, a finite sample $Z = (z_1, \ldots, z_n)$, and costs for predicting positive c_i^1 and negative c_i^0 labels on each point z_i. The goal is to output a hypothesis $\hat{h} \in \mathcal{H}$ that satisfies

$$\hat{h} \in \arg\min_{h\in\mathcal{H}} \sum_{i=1}^{n} [h(z_i)c_i^1 + (1 - h(z_i))c_i^0].$$

Kearns et al. [2018] show these equivalences to subsequently assume access to a cost-sensitive classification oracle, which they utilize in the FairNR algorithm. This is motivated by many practical heuristics for agnostic learning, which cannot be polynomial time in the worst case but work well in practice. These heuristics can be employed to achieve subgroup fairness using the reduction, and in practice should converge quickly.

The FairNR algorithm satisfies a generalized version of α-discrimination (Definition 5) known as *False Positive Subgroup Fairness* (Definition 8 below) which ensures approximate fairness for a large number of subgroups—such as combinations of protected groups or, more generally, large structured subsets of individuals. In particular, this fairness notion allows a class $\mathcal{G}$ of indicator functions defined over a set of protected attributes. $\mathcal{G}$ defines a set of protected subgroups and each function $g \in G$ corresponds to one protected subgroup. Until now, our fairness notions had considered the special case of a single (binary) protected attribute and a single protected subgroup, where $\mathcal{G}$ was a singleton containing the only the function $g(a) = a$. Here we allow x to contain multiple protected and non-protected attributes.

As its name suggests, the fairness definition used in Kearns et al. [2018] (Definition 8) considered *false positive fairness*, whereas we have been considering *false negative fairness* until now. However, as the authors note, ensuring equality of false positives is symmetric to considering fairness in true positives. One can think of this as simply transforming the cost vectors in a way that penalizes false negatives rather than false positives, or equivalently, flipping the labels of the training samples and ensuring equality of false positives

on these modified entries. In this section, we proceed with false positive fairness for consistency with the previous literature.

Definition 8 (False Positive (FP) Subgroup Fairness [Kearns et al., 2018])**.** Fix any classifier h, distribution D, collection of group indicators $\mathcal{G}$, and parameter $\alpha \in [0, 1]$. For each $g \in \mathcal{G}$, define,

$$A_{\text{FP}}(g, D) = \Pr_D[g(x) = 1, y = 0],$$

and

$$B_{\text{FP}}(g, h, D) = |\text{FP}(h) - \text{FP}(h, g)|,$$

where $\text{FP}(h) = \Pr_{h,D}[h(z) = 1|y = 0]$ denotes the overall false-positive rate of h and $\text{FP}(h, g) = \Pr_{h,D}[h(z) = 1|g(x) = 1, y = 0]$ denotes the false-positive rate of h on group g. We say h satisfies α-False Positive (FP) Fairness with respect to D and $\mathcal{G}$ if for every $g \in \mathcal{G}$,

$$A_{\text{FP}}(g, D)B_{\text{FP}}(g, h, D) \leq \alpha.$$

To complete the analogy to Definition 5, $A_{\text{FP}}(g, D)$ corresponds to P_{0a} and $\text{FP}(h, g)$ corresponds to $\gamma_{0a}(h)$.

Given this definition, the Fair ERM Linear Program is defined as:

$$\begin{aligned} &\min_{h \in \mathcal{H}} \mathbb{E}_{z \sim D}[err(h, D)] \\ \text{s.t. } \forall g \in \mathcal{G}: \quad &A_{\text{FP}}(g, D)(\text{FP}(h) - \text{FP}(h, g)) \leq \alpha \\ &A_{\text{FP}}(g, D)(\text{FP}(h, g) - \text{FP}(h)) \leq \alpha. \end{aligned}$$

The FairNR algorithm uses the fact that the Fair ERM Linear Program (LP) can be cast as a two-player zero-sum game and solved approximately with no-regret dynamics. Kearns et al. [2018] derive the partial Lagrangian of the LP, since computing an approximate solution to this LP is equivalent to finding an approximate minimax solution for a corresponding zero-sum game [Freund and Schapire, 1996]. In the corresponding game, the Learner attempts to output a distribution over hypotheses that minimizes error and satisfies the fairness constraints, while the Auditor attempts to penalize fairness violations by identifying the subgroup with the largest fairness violation. The rewritten LP reduces the problem of finding the most violated fairness constraint to a CSC problem, which is where the oracles are needed. This is achieved by introducing two dual variables λ_g^+ and λ_g^- as multipliers for each fairness constraint, and also restricting the dual space to $\Lambda = \{\lambda \in \mathbb{R}_+^{2|\mathcal{G}(S)|} | \|\lambda\|_1 \leq C\}$, where C is a parameter of the algorithm. The core of the algorithm comes from the no-regret play in the game described above, computed by the two CSC oracles. At each time step t, the CSC oracles compute a response based on the cumulative losses on the auditor's previous plays, represented as $\text{LC}(\lambda^{(t-1)})$.

We modify the FairNR algorithm to additionally satisfy differential privacy. We call this algorithm *Private-FairNR*, formally presented in Algorithm 4 in Appendix C.2. We introduce differential privacy into our algorithm by using a differentially private subroutine (Private Follow The Perturbed Leader (FTPL*) in Appendix C.1) to private computing the best responses of the players in each round. Privacy of the overall algorithm follows by post-processing and composition guarantees.

In addition to simply using a private subroutine, our algorithm differs from that of Kearns et al. [2018] because we add exponential rather than uniform noise to the Learner's loss in each round. We tune the magnitude of the noise to be large enough to ensure privacy, yet still small enough to ensure fairness and accurate classification. Additional technical challenges arise from the fact that Follow The Perturbed Leader (FTPL) was initially analyzed using one fixed noise vector, not one drawn fresh each round. However to ensure privacy, we need a fresh draw of noise for each round.

We now state our main result of this section: there exists a polynomial time algorithm with provable accuracy, privacy, and fairness guarantees.

Theorem 3. For any accuracy parameters $\alpha, \beta \in (0,1)$, given an input of n data points and access to oracles $CSC(\mathcal{H})$ and $CSC(\mathcal{G})$, there exists an algorithm that runs in polynomial time, is (ϵ, δ)-differentially private, and with probability at least $1-\beta$, outputs a randomized classifier $\hat{h}$ with error $err(\hat{h}) \leq OPT + \alpha$ and for any $g \in \mathcal{G}$, the fairness constraint violation satisfies

$$A_{FP}(g, D)|\mathrm{FP}(\hat{h}) - \mathrm{FP}(\hat{h}, g)| \leq \xi + O(\alpha).$$

Details of the algorithm and the proof of Theorem 3 are deferred to Appendix C.2.

References

Jacob D. Abernethy, Chansoo Lee, Audra McMillan, and Ambuj Tewari. Online learning via differential privacy. arXiv pre-print 1711.10019, 2017.

Sarah Bird, Solon Barocas, Kate Crawford, Fernando Diaz, and Hanna Wallach. Exploring or exploiting? Social and ethical implications of autonomous experimentation in AI. In *Proceedings of the Workshop on Fairness, Accountability, and Transparency in Machine Learning*, 2016.

Nicholas Carlini, Chang Liu, Jernej Kos, Ulfar Erlingsson, and Dawn Song. The secret sharer: Measuring unintended neural network memorization and extracting secrets. arXiv pre-print 1802.08232, 2018.

Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12:1069–1109, 2011.

Anupam Datta, Shayak Sen, and Michael Carl Tschantz. Correspondences between privacy and nondiscrimination: Why they should be studied together. arXiv pre-print 1808.01735, 2018.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, pages 265–284, 2006.

Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and differential privacy. In *Proceedings of the IEEE 51st Annual Symposium on Foundations of Computer Science*, FOCS '10, pages 51–60, 2010.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, pages 214–226, 2012.

Michael D. Ekstrand, Rezvan Joshaghani, and Hoda Mehrpouyan. Privacy for all: Ensuring fair and equitable privacy protections. In *Proceedings of the 1st Conference on Fairness, Accountability and Transparency*, FAT* '18, pages 35–47. PMLR, 2018.

Yoav Freund and Robert E. Schapire. Game theory, on-line prediction and boosting. In *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, COLT '96, pages 325–332, 1996.

Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. In *Advances in Neural Information Processing Systems 29*, NIPS '16, pages 3315–3323, 2016.

Ursula Hebert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. Multicalibration: Calibration for the (computationally-identifiable) masses. In *Proceedings of the 35th International Conference on Machine Learning*, ICML '18, pages 1939–1948, 2018.

Matthew Jagielski, Michael Kearns, Jieming Mao, Aaron Oprea, Alina Roth, Saeed Sharifi-Malvajerdi, and Jonathan Ullman. Differentially private fair learning. *arXiv preprint arXiv:1812.02696*, 2019.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Michael J. Kearns, Mallesh M. Pai, Aaron Roth, and Jonathan Ullman. Mechanism design in large games: Incentives and privacy. In *Proceedings of the 5th Conference on Innovations in Theoretical Computer Scienc*, ITCS '14, pages 403–410, 2014.

Michael J. Kearns, Seth Neel, Aaron Roth, and Zhiwei Steven Wu. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In *Proceedings of the 35th International Conference on Machine Learning*, ICML '18. PMLR, 2018.

Andrew McGregor, Ilya Mironov, Toniann Pitassi, Omer Reingold Reingold, Kunal Talwar, and Salil Vadhan. The limits of two-party differential privacy. In *Proceedings of the IEEE 51st Annual Symposium on Foundations of Computer Science*, FOCS '10, pages 81–90, 2010.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 94–103, 2007.

Harikrishna Narasimhan. Learning with complex loss functions and constraints. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, AISTATS '18, pages 1646–1654. PMLR, 2018.

Blake Woodworth, Suriya Gunasekar, Mesrob I. Ohannessian, and Nathan Srebro. Learning non-discriminatory predictors. In *Proceedings of the 2017 Conference on Learning Theory*, COLT '17, pages 1920–1953. PMLR, 2017.

A Differentially private tools

One common tool for achieving differential privacy is the *Laplace Mechanism*, which adds noise that scales with the *sensitivity* of the analysis.

Definition 9 (ℓ_1-sensitivity)**.** The ℓ_1-sensitivity of a function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$ is

$$\Delta f = \max_{D,D' \text{neighbors}} \|f(D) - f(D')\|_1.$$

Definition 10 (Laplace mechanism [Dwork et al., 2006])**.** Given any function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$, the Laplace mechanism is defined as:

$$\mathcal{A}_L(D, f(\cdot), \epsilon) = f(D) + (Y_1, \ldots, Y_k)$$

where Y_i are i.i.d. random variables drawn from $\text{Lap}(\Delta f/\epsilon)$.

Theorem 4 ([Dwork et al., 2006])**.** The Laplace mechanism preserves $(\epsilon, 0)$-differential privacy.

[DK: added following theorem for PTR: lemma 2]

Theorem 5 ([Dwork et al., 2006])**.** Let $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$, and let $y = \mathcal{A}_L(D, f(\cdot), \epsilon)$ be the output of a Laplace mechanism. Then $\forall \beta \in (0, 1]$:

$$\Pr[||f(D) - y||_\infty \geq \ln(\frac{k}{\beta}) \cdot (\frac{\Delta f}{\epsilon})] \leq \beta$$

Differential privacy also has a number of desirable algorithmic properties, including robustness to *post-processing*, and that privacy guarantees *compose adaptively* as additional analysis are performed on the data.

Proposition 1 (Post-processing [Dwork et al., 2006])**.** Let $\mathcal{A} : \mathbb{N}^{|\mathcal{X}|} \to \mathcal{R}$ be an $(\epsilon, \delta)-$differentially private algorithm, and let $g : \mathcal{R} \to \mathcal{R}'$ be an arbitrary randomized mapping. Then $g \circ \mathcal{A} : \mathbb{N}^{|\mathcal{X}|} \to \mathcal{R}'$ is (ϵ, δ)-differentially private.

Theorem 6 (Advanced composition [Dwork et al., 2010])**.** Let $\mathcal{A} : \mathcal{U} \to \mathcal{R}^T$ be a T-fold adaptive composition of (ϵ, δ)-differentially private mechanisms. Then $\mathcal{A}$ satisfies $(\epsilon', T\delta + \delta')$-differential privacy for

$$\epsilon' = \epsilon\sqrt{2T \ln(1/\delta')} + T\epsilon(e^\epsilon - 1).$$

In particular, for any $\epsilon \leq 1$, if $\mathcal{A}$ is a T-fold adaptive composition of $(\epsilon/\sqrt{8T \ln(1/\delta)}, 0)$-differentially private mechanisms, then $\mathcal{A}$ satisfies (ϵ, δ)-differential privacy.

B Approximate Fairness with Differential Privacy

This appendix contains the omitted proofs from Section 4

Theorem 7 (Real-valued Additive Chernoff-Hoeffding Bound)**.** Let $X_1, ..., X_d$ be i.i.d. random variables with $\mathbb{E}[X_i] = \mu$ and $a \leq X_i \leq b$ for all i. Then for every $\rho > 0$,

$$Pr\Big[\Big|\frac{\sum_i X_i}{n} - \mu\Big| > \rho\Big] \leq 2\exp\left(\frac{-2\rho^2 n}{(b-a)^2}\right).$$

Lemma 4 (Woodworth et al. [2017])**.** For any binary predictor h,

$$\Pr\big[\big|\Gamma(h) - \Gamma^Z(h)\big| > \rho\big] \leq 16\exp\left(\frac{1}{4}\rho^2 n \min_{ya} P_{ya}\right) = \delta,$$

where Γ is as defined in as Definition 5 for $y, a \in \{0, 1\}$.

Lemma 3 (Concentration of Utility)**.** For any sample Z of size n drawn i.i.d. from distribution D and for any binary predictor h,

$$\Pr\big[\big|u(Z,h) - u(D,h)\big| > \rho\big] \leq 18\exp\left(-\min_a \frac{\rho^2 n P_{1a}}{16}\right).$$

Proof. Fix any $h \in \mathcal{H}$. Recall that $P_{1a} = \Pr[Y = 1, A = a]$.

$$\begin{aligned}
&\Pr[|u(Z,h) - u(D,h)| > \rho] \\
&= \Pr[|\Gamma^Z(h) + err^Z(h) - (\Gamma(h) + err(h))| > \rho] \\
&\leq \Pr[|\Gamma^Z(h) - \Gamma(h)| + |err^Z(h) - err(h)| > \rho] \\
&\leq \Pr\left[|\Gamma^Z(h) - \Gamma(h)| > \frac{\rho}{2}\right] + \Pr\left[|err^Z(h) - err(h)| > \frac{\rho}{2}\right] \\
&\leq 16\exp\left(-\min_a \frac{\rho^2 n P_{1a}}{16}\right) + 2\exp\left(\frac{-2\rho^2 n}{4}\right) \\
&\leq 18\exp\left(-\min_a \frac{\rho^2 n P_{1a}}{16}\right)
\end{aligned}$$

where the first inequality follows from the triangle inequality, the second from union bound, and the third from Theorem 7 and Lemma 4. The final inequality is due the the fact $P_{1a} \leq 1$ for any a. □

Lemma 2. *$PTR(\cdot,\cdot,\delta)$ outputs a false-positive with probability less than or equal to δ.* [DK: pf in appendix]

Proof. *$PTR(Z,\epsilon,\delta)$ returns a false positive if and only if it outputs $\top$ and*

$$\min_{a\in\{0,1\}} \frac{|Z_{1a}|}{n} < \min_{a\in\{0,1\}} P_{1a}$$

given the above setting, the following inequality (the test) holds with probability less than or equal to δ by Lemma 4:

$$\min_{a\in\{0,1\}} \frac{|Z_{1a}|}{n} + \text{Lap}(\frac{1}{n\epsilon}) \geq \min_{a\in\{0,1\}} P_{1a} + \ln(\frac{1}{\delta})(\frac{1}{n\epsilon}).$$

[DK: need more detail?] This inequality holds if and only if the test outputs $\top$. Thus a false positive is returned with probability less than δ.

□

Lemma 1. *$\Gamma^Z(h)$ has sensitivity $\Delta\Gamma = \min_{a\in\{0,1\}} 2/(|Z_{1a}| - 1)$, and $err^Z(h)$ has sensitivity $\Delta err = 1/n$.*

Proof. We start by showing that $\Delta err = 1/n$. The empirical error function for any hypothesis h is $err^Z(h) = \frac{1}{n}\sum_{z_i\in Z} \Pr[h(x_i,a_i) \neq y_i]$. Changing a single entry in Z can change at most one z_i, which can change $\Pr[h(x_i,a_i) \neq y_i]$ by at most 1. Since $err^Z(h)$ averages these probabilities, this can change the value of $err^Z(h)$ by at most $1/n$.

We next bound the sensitivity of $\Gamma^Z(h)$, and we examine two cases for neighboring databases Z, Z'. Case 1: The difference in databases we consider is changing an entry within a subgroup (i.e., within Z_{1a} and Z'_{1a} for $a \in \{0,1\}$). Assume without loss of generality that $a = 0$. Let the differing entries be called $z \in Z_{10}$ and $z' \in Z'_{10}$.

Then, we have $\gamma^{Z'}_{11} = \gamma^Z_{11}$, but,

$$\begin{aligned}
\gamma^Z_{10}(h) &= \frac{1}{|Z_{10}|}\sum_{Z_{10}} h(x,0) \\
\gamma^{Z'}_{10}(h) &= \frac{1}{|Z_{10}|}\Big(\sum_{z_i\in Z_{10}\cap Z'_{10}} h(x_i,0) + h(x',0)\Big).
\end{aligned}$$

Then,

$$\begin{aligned}\Gamma^{Z'}(h) &= |\gamma_{10}^{Z'}(h) - \gamma_{11}^{Z'}(h)| \\ &= |\gamma_{10}^{Z'}(h) - \gamma_{11}^{Z}(h)| \\ &\leq \Gamma^{Z}(h) + \frac{1}{|Z_{10}|}.\end{aligned}$$

Case 2: The neighboring databases are different in that an entry moves from one subgroup and to another subgroup. Without loss of generality, let $\gamma_{11}^{Z}(h) < \gamma_{10}^{Z}(h)$ and $Z' = (Z_{11} \setminus \{z_1\}) \cup (Z_{10} \cup \{z_1\})$, where z_1 is a positively labeled example. Then,

$$\begin{aligned}\gamma_{11}^{Z'}(h) &= \frac{1}{|Z'_{11}|} \sum_{z_i \in Z'_{11}} h(x_i) \\ &= \frac{1}{|Z'_{11}|}\Big(\sum_{z_i \in Z_{11}} h(x_i) - h(x_1)\Big) \\ &= \frac{1}{|Z_{11}| - 1}\Big(\sum_{z_i \in Z_{11}} h(x_i) - h(x_1)\Big).\end{aligned}$$

Similarly,

$$\begin{aligned}\gamma_{10}^{Z'}(h) &= \frac{1}{|Z'_{10}|} \sum_{z_i \in Z'_{10}} h(x_i) \\ &= \frac{1}{|Z'_{10}|}\Big(\sum_{z_i \in Z_{10}} h(x_i) + h(x_1)\Big) \\ &= \frac{1}{|Z_{10}| + 1}\Big(\sum_{z_i \in Z_{10}} h(x_i) + h(x_1)\Big).\end{aligned}$$

Thus, we see that

$$\begin{aligned}|\Gamma^{Z'}(h) - \Gamma^{Z}(h)| &= (\gamma_{10}^{Z'} - \gamma_{11}^{Z'}) - (\gamma_{10}^{Z} - \gamma_{11}^{Z}) \\ &= (\gamma_{10}^{Z'} - \gamma_{10}^{Z}) + (\gamma_{11}^{Z} - \gamma_{11}^{Z'})\end{aligned}$$

Let $\phi_{10} = \gamma_{10}^{Z'} - \gamma_{10}^{Z}$ and $\phi_{11} = \gamma_{11}^{Z} - \gamma_{11}^{Z'}$

$$\begin{aligned}\phi_{10} &= \frac{1}{|Z_{10}| + 1}\Big(\sum_{z_i \in Z_{10}} h(x_i) + h(z_1)\Big) - \gamma_{10}^{Z} \\ &= \frac{1 - \gamma_{10}^{Z}}{|Z_{10}| + 1}.\end{aligned}$$

and

$$\begin{aligned}\phi_{11} &= \gamma_{11}^{Z} - \frac{1}{|Z_{11}| - 1}\Big(\sum_{z_i \in Z_{11}} h(x_i) - h(x_1)\Big) \\ &= \frac{1 - \gamma_{11}^{Z}}{|Z_{11}| - 1}.\end{aligned}$$

Thus, we have

$$|\Gamma^{Z'}(h) - \Gamma^{Z}(h)| = \phi_{10} + \phi_{11} = \frac{1 - \gamma_{10}^{Z}}{|Z_{10}| + 1} + \frac{1 - \gamma_{11}^{Z}}{|Z_{11}| - 1}.$$

Similar arguments can be used to show the sensitivity of $\Gamma(h)^Z$ when moving a negatively labeled example between groups and when switching a label for an example as it moves between groups. Thus we have $\Delta\Gamma = \max_{a\in\{0,1\}} \left(\frac{1}{|Z_{1a}|}, \frac{\gamma^Z_{1a}}{|Z_{1a}|-1} + \frac{\gamma^Z_{1\neg a}}{|Z_{1\neg a}|+1}, \frac{1-\gamma^Z_{1a}}{|Z_{1a}|+1} + \frac{1-\gamma^Z_{1\neg a}}{|Z_{1\neg a}|-1}, \frac{1-\gamma^Z_{1a}}{|Z_{1a}|+1} + \frac{\gamma^Z_{1\neg a}}{|Z_{1\neg a}|-1}\right)$, where $\neg a = 1 - a$.

We can bound this sensitivity using our assumption that we have at least two positively labeled examples for each protected attribute: $|Z_{1a}| \geq 2$ and $|Z_{1\neg a}| \geq 2$. First we note that $\gamma^Z_{1a}, 1-\gamma^Z_{1a}, \gamma^Z_{1\neg a}, 1-\gamma^Z_{1\neg a} \leq 1$. Next, if $|Z_{1a}| \leq |Z_{1\neg a}|$, then

$$\frac{1}{|Z_{1a}|-1} + \frac{1}{|Z_{1\neg a}|+1} \leq \frac{2}{|Z_{1a}|-1}.$$

Similarly, if $|Z_{1\neg a}| \leq |Z_{1a}|$, then $|Z_{1\neg a}| - 1 \leq |Z_{1a}| - 1$,

$$\frac{1}{|Z_{1a}|-1} + \frac{1}{|Z_{1\neg a}|+1} \leq \frac{2}{Z_{1\neg a}-1}.$$

Thus, we have that $\Delta\Gamma \leq \max_{a\in\{0,1\}} \frac{2}{|Z_{1a}|-1}$. [DK: deleted assumption for old approximation] □

C Algorithm via Reduction to Cost-Sensitive Classification

In this Appendix, we provide all the omitted details from Section 5.

C.1 Differentially Private Follow the Perturbed Leader

We first introduce a fundamental tool, the differentially private version of Follow the Perturbed Leader [Abernethy et al., 2017]. This algorithm will be used as a subroutine in our algorithm Private-FairNR, given in Appendix C.2.

Algorithm 3 Follow the Perturbed Leader* (FTPL*(ϵ))

Input: Action set $\mathcal{D} \subseteq \{0,1\}^n$, $\epsilon \in (0,1)$
Initialize: $d^1 \in \mathcal{D}$ be arbitrary.
for t = 1,...,T **do**
Play action d^t; Observe loss vector ℓ^t and suffer loss $\langle \ell^t, d^t \rangle$.
Update:

$$d^{t+1} = \text{argmin}_{d\in\mathcal{D}} \left[\sum_{r\leq t} \langle \ell^r, d\rangle + \langle \xi^t, d\rangle \right],$$

where $\xi^t \sim \text{Lap}(1/\epsilon)$ independently for each t and for each coordinate.
end for

Theorem 8 ([Kalai and Vempala, 2005])**.** For nonnegative $\mathcal{D}, \mathcal{S} \subset \mathbb{R}^n$, where $\mathcal{D}$ is the decision space and $\mathcal{S}$ is the state space, FTPL gives

$$\mathbb{E}[\text{cost of FTPL}(\epsilon)] \leq (1 + 2A\epsilon)\text{OPT} + \frac{2M(1+\ln(n))}{\epsilon},$$

where for all $d, d' \in \mathcal{D}$ and $s \in \mathcal{S}$, $M \geq \|d - d'\|_1$, and $A \geq \|s\|_1$.

One can interpret the decision space as the set of all actions that could be taken by an agent, and the state space as the set of all possible loss vectors.

Lemma 5. Suppose that in each round $t \in [T]$, the loss vector ℓ^t is computed via a function that has access to the database x, ie. $\ell^t = f^t(x)$. Then FTPL*(ϵ') is (ϵ, δ)-differentially private for

$$\epsilon' = \frac{\epsilon}{\max_t(\Delta f^t)\sqrt{T \ln 1/\delta}}.$$

Proof. For each round t, we compute a new loss vector and also add i.i.d. noise from $\text{Lap}(1/\epsilon')$ to each coordinate. This is equivalent to privately computing the loss vector with the Laplace Mechanism. By Theorem 4, in each round, the algorithm is $(\epsilon', 0)$-differentially private. Since we do this for T rounds, by Theorem 6 and Proposition 1 the overall algorithm is (ϵ, δ)-differentially private. □

C.2 Efficient Private and Fair Algorithm

In our game, we define the payoff function for any pair of actions $(h, \lambda) \in \mathcal{H} \times \Lambda_{pure}$ as:

$$U(h, \lambda) = err(h, D) + \sum_{g \in \mathcal{G}(S)} (\lambda_g^+ \Phi_+(h, g) + \lambda_g^- \Phi_-(h, g)) \text{ where}$$

$$\Phi_+(h, g) = A_{\text{FP}}(g, D)(\text{FP}(h) - \text{FP}(h, g)) \text{ and}$$
$$\Phi_-(h, g) = A_{\text{FP}}(g, D)(\text{FP}(h, g) - \text{FP}(h))$$

Algorithm 4 Private-FairNR: Private and Fair No-Regret Dynamics

Input: distribution $\mathcal{D}$ over n labeled data points, CSC oracles CSC($\mathcal{H}$) and CSC($\mathcal{G}$), dual bound C, target accuracy parameters α, β, privacy parameters ϵ, δ, absolute constant c_0, $d_2 = \text{VCDIM}(\mathcal{G})$.

Initialize: Let $C = 1/\alpha$, $\overline{\lambda}^{(0)} = \mathbf{0}$,

$$T = \frac{4\sqrt{n}\ln(2/\beta)}{\alpha^4}, \quad m = \frac{\ln(2T/\beta) d_2 \ln(n) C^2 c_0 T}{\sqrt{n}(1+C)^2 \ln(2/\beta)}, \text{ and } \epsilon' = \frac{\epsilon}{\frac{2+C}{n}\sqrt{mT \ln 1/\delta}}.$$

for t = 1,. . . ,T **do**
  for s = 1,. . . ,m **do**
    (Sample from the learner's FTPL* Distribution)
    Draw a random vector ξ^s by independently for each coordinate drawing from $\text{Lap}(1/\epsilon')$
    Use oracle CSC($\mathcal{H}$) to compute:
      $h^{(s,t)} = \text{argmin}_{h \in \mathcal{H}} \left\langle \text{LC}\left(\overline{\lambda}^{(t-1)}\right) + \xi^s, h \right\rangle$
  end for
  Let $\hat{h}^t$ be the empirical distribution over $\{h^{s,t}\}$
  (Auditor best responds to $\hat{h}^t$)
  Use oracle CSC($\mathcal{G}$) to compute:
    $\lambda^t = \text{argmax}_\lambda \mathbb{E}_{h \sim \hat{h}^t}[U(h, \lambda)]$
  Update: Let $\overline{\lambda}^{(t)} = \sum_{t' \le t} \lambda^{t'}$
end for
Sample from the average distribution $\hat{h} = \sum_{t=1}^T \hat{h}^t$
Output: $\hat{h}$ the empirical distribution over the samples.

Using Lemma 5 and the parameters appropriate to Private-FairNR, we obtain a new bound for the Learner's regret.

Lemma 6. Let T be the time horizon for the no-regret dynamics. Let $D^1, ..., D^T$ be the sequence of distributions maintained by the Learner's FTPL*(ϵ') algorithm, and $\lambda^1, ..., \lambda^T$ be the sequence of plays by the auditor. Then

$$\sum_{t=1}^T \mathbb{E}_{h \sim D^t}[U(h, \lambda^t)] - \min_{h \in \mathcal{H}} \sum_{t=1}^T U(h, \lambda^t) \le \frac{\epsilon \text{OPT}}{\sqrt{mT \ln(1/\delta)}} + \frac{2(1 + \ln(n))(2 + C)\sqrt{mT \ln(1/\delta)}}{\epsilon}$$

where $\text{OPT} = \min_{h\in\mathcal{H}} \sum_{t=1}^{T} U(h, \lambda^t)$.

Proof. The bound follows from applying Theorem 8 with $A = (1 + C)/n$, $M = n$ and

$$\epsilon' = \frac{\epsilon}{\frac{2+C}{n}\sqrt{mT \ln 1/\delta}}.$$

Note that $\max_t(\Delta f^t) = \max_t(\Delta \text{LC}(\bar{\lambda}^{(t-1)})) = \frac{2+C}{n}$ since the loss vector is created by assigning, for each example (x_i, y_i), a cost with minimum value of $-1/n$ and maximum value of $(1 + C)/n$. □

We will the following theorem from Freund and Schapire [1996] to show that the no-regret dynamics of the learner and auditor in our algorithm converges to an approximate equilibrium of the game.

Theorem 9. [Freund and Schapire, 1996] Let $D^1, D^2, \ldots, D^T \in \Delta_{\mathcal{H}(S)}$ be a sequence of distributions played by the Learner, and let $\lambda^1, \lambda^2, \ldots, \lambda^T \in \Lambda_{pure}$ be the Auditor's sequence of approximate best responses against these distributions respectively. Let $\bar{D} = \frac{1}{T}\sum_{t=1}^{T} D^t$ and $\bar{\lambda} = \frac{1}{T}\sum_{t=1}^{T} \lambda^t$ be the two players' empirical distributions over their strategies, and γ_L and γ_A be the average regret of the learner and auditor. Suppose that the regret of the learner satisfies

$$\sum_{t=1}^{T} \mathbb{E}_{h\sim D^t}\big[U(h, \lambda^t)\big] - \min_{h\in\mathcal{H}(S)} \sum_{t=1}^{T} U(h, \lambda^t) \le \gamma_L T, \text{ and}$$
$$\max_{\lambda\in\Lambda} \sum_{t=1}^{T} \mathbb{E}_{h\sim D^t}\big[U(h, \lambda)\big] - \sum_{t=1}^{T} \mathbb{E}_{h\sim D^t}\big[U(h, \lambda^t)\big] \le \gamma_A T.$$

Then $(\bar{D}, \bar{\lambda})$ is an $(\gamma_L + \gamma_A)$-approximate minimax equilibrium of the game.

We now restate and prove our main result, which analyzes the new private version of Private-FairNR that uses FTPL*(ϵ) instead of FTPL.

Theorem 3. For any accuracy parameters $\alpha, \beta \in (0, 1)$, given an input of n data points and access to oracles $CSC(\mathcal{H})$ and $CSC(\mathcal{G})$, there exists an algorithm that runs in polynomial time, is (ϵ, δ)-differentially private, and with probability at least $1 - \beta$, outputs a randomized classifier $\hat{h}$ with error $err(\hat{h}) \le OPT + \alpha$ and for any $g \in \mathcal{G}$, the fairness constraint violation satisfies

$$A_{FP}(g, D)|\text{FP}(\hat{h}) - \text{FP}(\hat{h}, g)| \le \xi + O(\alpha).$$

Proof. The only step where Private-FairNR interacts with the data is through the FTPL* subroutine. By Lemma 5, this subroutine is (ϵ, δ)-differentially private, and by Proposition 1, all of Private-FairNR is (ϵ, δ)-differentially private as well. We now just need to prove that the utility guarantees hold.

By Kearns et al. [2018], it suffices to show that with probability at least $1 - \beta$, $(\hat{h}, \bar{\lambda})$ is a α-approximate equilibrium in the zero-sum game.

By Lemma 6, the regret of the sequence $D^1, \ldots, D^T$ implies that:

$$\begin{aligned}\gamma_L &= \frac{1}{T}\big[\sum_{t=1}^{T} \mathbb{E}_{h\sim D^t}[U(h, \lambda^t)] - \min_{h\in\mathcal{H}} \sum_{t=1}^{T} U(h, \lambda^t)\big] \\ &\le \frac{\epsilon \cdot \text{OPT}}{T\sqrt{mT\ln(1/\delta)}} + \frac{2(1 + \ln(n))(2 + C)\sqrt{m\ln(1/\delta)}}{\epsilon\sqrt{T}}.\end{aligned}$$

By Kearns et al. [2018], with probability $1 - \beta/2$ we have

$$\gamma_A \le \sqrt{\frac{c_0 C^2(\ln(2T/\delta) + d_2 \ln(n))}{m}},$$

and except with probability $\beta/2$, by Freund and Schapire [1996], the pair $(\hat{h}, \bar{\lambda})$ form an η-approximate equilibrium for

$$\eta = \gamma_A + \gamma_L + \sqrt{\frac{c_0 C^2(\ln(2/\delta) + d_2 \ln(n))}{m}}.$$

Note that $\eta \leq \alpha$ as long as we have $C = 1/\alpha$,

$$m = \frac{\ln(2T/\beta) d_2 \ln(n) C^2 c_0 T}{\sqrt{n}(1+C)^2 \ln(2/\beta)},\ T = \frac{4\sqrt{n}\ln(2/\beta)}{\alpha^4}, \text{ and}$$

$$\frac{\epsilon}{\sqrt{\ln(1/\delta)}} \geq \frac{24\sqrt{\ln(2T/\beta) d_2 \ln(n) c_0}(1+\ln(n))}{\alpha^2 \sqrt{\sqrt{n}\ln(2/\beta)}}. \tag{1}$$

The conditions on m and T are satisfied by the initialization of the parameters in Algorithm 4. To see why the condition of Equation (1) is required, we need that the regret $\gamma_L \leq c\alpha$ for some constant c.

$$\frac{\epsilon \mathrm{OPT}}{T\sqrt{mT\ln(1/\delta_\epsilon)}} + \frac{(1+\ln(n))(2+C)\sqrt{m\ln(1/\delta_\epsilon)}}{\epsilon\sqrt{T}} \leq c\alpha.$$

Let $m^* = m/T$. Then since the multiplicative term of the regret goes to 0 as T grows, we only need to consider the growth of the additive term.

$$\begin{aligned}
\frac{2(1+\ln(n))(2+C)\sqrt{m\ln(1/\delta)}}{\epsilon\sqrt{T}} &\leq c\alpha \\
\frac{2(1+\ln(n))(2+C)\sqrt{m^* T\ln(1/\delta)}}{\epsilon\sqrt{T}} &\leq c\alpha \\
\frac{2\sqrt{m^*}(1+\ln(n))(2+C)}{c\alpha} &\leq \frac{\epsilon}{\sqrt{\ln(1/\delta)}} \\
\frac{2C(2+C)\sqrt{\ln(2T/\beta) d_2 \ln(n) c_0}(1+\ln(n))}{c\alpha(1+C)\sqrt{\sqrt{n}\ln(2/\beta)}} &\leq \frac{\epsilon}{\sqrt{\ln(1/\delta)}}
\end{aligned}$$

Setting $C = 1/\alpha$,

$$\frac{6\sqrt{\ln(2T/\beta) d_2 \ln(n) c_0}(1+\ln(n))}{c\alpha^2\sqrt{\sqrt{n}\ln(2/\beta)}} \leq \frac{\epsilon}{\sqrt{\ln(1/\delta)}}$$

Setting $c = 1/4$, we have our bound.

□