Active Fairness Auditing

Tom Yan [*1] **Chicheng Zhang** [*2]

Abstract

The fast spreading adoption of machine learning (ML) by companies across industries poses significant regulatory challenges. One such challenge is scalability: how can regulatory bodies efficiently *audit* these ML models, ensuring that they are fair? In this paper, we initiate the study of query-based auditing algorithms that can estimate the demographic parity of ML models in a query-efficient manner. We propose an optimal deterministic algorithm, as well as a practical randomized, oracle-efficient algorithm with comparable guarantees. Furthermore, we make inroads into understanding the optimal query complexity of randomized active fairness estimation algorithms. Our first exploration of active fairness estimation aims to put AI governance on firmer theoretical foundations.

1. Introduction

With the growing usage of artificial intelligence across industries, governance efforts are increasingly ramping up. A key challenge in regulatory efforts is the problem of scalability. Even for well-resourced countries like Norway, which is a pioneer in AI governance, regulators are only able to monitor and engage with a "small fraction of the companies" (McCarthy, 2021). This growing issue calls for a better understanding of *efficient* algorithms that can audit machine learning (ML) models, which we now formalize.

Problem Formulation: A regulatory institution is interested in auditing an unknown model $h^* : \mathcal{X} \to \{-1, 1\}$ held by a company (e.g. a lending company in the finance sector), where $\mathcal{X}$ is the feature space (e.g. of all information supplied by users). We assume that the regulatory institution only has knowledge of the hypothesis class $\mathcal{H}$ where h^* comes from (e.g. the family of linear classifiers), and it would like to estimate $\mu(h^*)$ for a function μ that measures the model property of interest. To this end, the institution is allowed to send black-box queries to the model h^*, i.e. send the company a query example x and receive $h^*(x)$. The regulatory institution's goal is to *efficiently* estimate $\mu(h^*)$ to within an error of at most $\epsilon > 0$.

We measure an algorithm's *efficiency* in terms of both its *query complexity* and *computational complexity*. Having an auditing algorithm with low query and computational complexity naturally helps to address the scalability challenge: greater efficiency means that each audit may be processed faster and more audits may be processed at a time.

Property of Interest: While which properties μ to assess is still heavily debated by regulators, we initiate the study of algorithms that audit fairness, a mainstay in regulatory efforts. In particular, we will take μ to be Demographic Parity (DP)[1]: given distribution D_X over $\mathcal{X} \times \{0, 1\}$ (where feature x and sensitive attribute x_A are jointly drawn from), $\mu_{D_X}(h) = \Pr_{(x,x_A)\sim D_X}(h(x) = 1 | x_A = 1) - \Pr_{(x,x_A)\sim D_X}(h(x) = 1 | x_A = 0)$. For brevity, when it is clear from context, we abbreviate $\Pr_{D_X}, \mu_{D_X}$ as $\Pr, \mu$, respectively. DP measures the degree of disparate treatment of model h on the two sub-populations $x \mid x_A = 0$ and $x \mid x_A = 1$, which we assume are non-negligible: $\underline{p} := \min(\Pr(x_A = 1), \Pr(x_A = 0)) = \Omega(1)$. Achieving a small Demographic Parity may be thought of as a stronger version of the US Equal Employment Opportunity Commission's "four-fifths rule".[2]

To focus on query complexity, we will abstract away the difficulty of evaluating μ by assuming that D_X is known, which means that for any h we may evaluate $\mu(h)$ to arbitrary precision; for instance, this may be achieved with the availability of an arbitrarily large number of (unlabeled) samples randomly drawn from $x \mid x_A = 0$ and $x \mid x_A = 1$. Our main challenge is that we do not know h^* and only want to query h^* *insofar as to be able to accurately estimate* $\mu(h^*)$.

[*]Equal contribution [1]Carnegie Mellon University [2]University of Arizona. Correspondence to: Tom Yan <tyan2@andrew.cmu.edu>, Chicheng Zhang <chichengz@cs.arizona.edu>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

[1]While fairness is the focus of our work, our algorithm may be adapted to any μ which is a function of $\mathcal{X}$ and h^*.

[2]The "selection rate for any race, sex, or ethnic group [must be at least] four-fifths (4/5) (or eighty percent) of the rate for the group with the highest rate."

Guarantees of the Audit: In this paper, we investigate algorithms that can provide two types of guarantees. The first is the natural, *direct estimation accuracy*: the estimate returned by the algorithm should be within ϵ of $\mu(h^*)$.

The second is that of *manipulation-proof* (MP) estimation. Audits can be very consequential to companies as they may be subject to hefty penalties if caught with violations. Not surprisingly, there have been effortful attempts in the past to avoid being caught with violations (e.g. Hotten, 2015) by "gaming" the audit. We formulate our notion of manipulation-proofness in light of one way the audit may be gamed, which we now describe. Note that all the auditor knows about the model used by the company is that it is consistent with the queried labels in the audit. So, while our algorithm may have estimated $\mu(h^*)$ accurately during audit-time, nothing stops the company from changing its model *post-audit* from h^* to a different model $h_{\text{new}} \in \mathcal{H}$ (e.g to improve profit), so long as h_{new} is still consistent with the queries seen during the audit. With this, we also look to understand: given this post-hoc possibility of manipulation, can we devise an algorithm that nonetheless ensures the algorithm's estimate is within ϵ of $\mu(h_{\text{new}})$?

Indeed, a robust set of audit queries would serve as a *certificate* that no matter which model the company changes to after the audit, its μ-estimation would remain accurate. Given a set of classifiers V, a classifier h, and a unlabeled dataset S, define the version space (Mitchell, 1982) induced by S to be $V(h,S) := \left\{h' \in V : h'(S) = h(S)\right\}$. An auditing algorithm is ϵ-manipulation-proof if, for any h^*, it outputs a set of queries S and estimate $\hat{\mu}$ that guarantees that $\max_{h \in \mathcal{H}(h^*,S)} \left|\mu(h) - \hat{\mu}\right| \leq \epsilon$.

Baseline: i.i.d Sampling: One natural baseline that comes to mind for the direct estimation is i.i.d sampling. We sample $O(1/\epsilon^2)$ examples i.i.d from the distribution $x \mid x_A = i$ for $i \in \{0,1\}$, query h^* on these examples and take the average to obtain an estimate of $\Pr(h^*(x) = +1 \mid x_A = i)$. Finally, we take the difference of these two estimates as our final DP estimate. By Hoeffding's Inequality, with high probability, this estimate is ϵ-accurate, and this estimation procedure makes $O(1/\epsilon^2)$ queries.

However, i.i.d sampling is not necessarily MP. To see an example, let there be $2n$ points in group $x_A = 1$ with $n = 1/\epsilon^2$ that are shattered by $\mathcal{H}$ and D_X is uniform over these points. Suppose that all points in group $x_A = 0$ are labeled the same: $\Pr_{D_X}(h(x) = 1 | x_A = 0) = 0, \forall h \in \mathcal{H}$. Then, μ-estimation reduces to estimating the proportion of positives in group $x_A = 1$. i.i.d sampling will randomly choose n of these data points to see, and it will produce an ϵ-accurate estimate of $\mu(h^*)$. However, we do not see the other n points. Since the $2n$ points are shattered by $\mathcal{H}$, *after* the queried points are determined, we see that the company can increase or decrease DP by up to $1/2$ by switching to a different model.

To perform both direct and MP estimation, it seems promising then to examine algorithms that make use of *non-iid* sampling. Moreover, for MP, we observe that the auditing algorithm should leverage knowledge of the hypothesis class as well, which i.i.d sampling is agnostic to.

Baseline: Active Learning: *PAC active learning* (Hanneke, 2014) (where PAC stands for Probably Approximately Correct (Valiant, 1984)) algorithms are a set of algorithms that can achieve both direct and MP estimation accuracy. PAC active learning algorithms guarantee that, with high probability, $\hat{h}$ in the resultant version space is such that $\mathbb{P}(\hat{h}(x) \neq h^*(x)) \leq \underline{p}\epsilon = O(\epsilon)$. With this, we have $\left|\mu(\hat{h}) - \mu(h^*)\right| \leq \epsilon$ (see Lemma C.1 in Appendix C for a formal proof).

To mention a setting where learning is favored over i.i.d sampling, learning homogeneous linear classifiers under certain well-behaved unlabeled data distributions requires only $O(d \log 1/\epsilon)$ queries (e.g. Dasgupta, 2005b; Balcan & Long, 2013) and would thus be far more efficient than $O(1/\epsilon^2)$ for low-dimensional learning settings with high auditing precision requirements.

Still, as our goal is only to estimate the μ values of the induced version space, it is unclear if we always need to go as far as to learn the model itself. In this paper, we investigate whether, and if so when, it may be possible to design adaptive approaches to efficiently directly and MP estimate $\mu(h^*)$ using knowledge of $\mathcal{H}$.

To the best of our knowledge, we are the first to theoretically investigate active approaches for direct and MP estimation of $\mu(h^*)$. Our first exploration of active fairness estimation seeks to provide a more complete picture of the theory of auditing machine learning models. Our hope is that our theoretical results can pave the way for subsequent development of practical and efficient algorithms.

Our Contributions: Our main contributions are on two fronts, MP and direct estimation of $\mu(h^*)$:

- For the newly introduced notion of manipulation-proofness, we identify a statistically optimal, but computationally intractable deterministic algorithm. We gain insights into its query complexity through comparisons to the two baselines, i.i.d sampling and PAC active learning.
- In light of the computational intractability of the optimal deterministic algorithm, we design a randomized algorithm that enjoys *oracle efficiency* (e.g. Dasgupta et al., 2007): it has an efficient implementation given access to a mistake-bounded online learning oracle, and an constrained empirical risk minimization oracle for the hypothesis class $\mathcal{H}$. Furthermore, its query

performance matches that of the optimal deterministic algorithm up to $\text{polylog}|\mathcal{H}|$ factors.

- Finally, on the direct estimation front, we obtain bounds on information-theoretic query complexity. We establish that MP estimation may be more expensive than direct estimation, thus highlighting the need to develop separate algorithms for the two guarantees. Then, we establish the usefulness of randomization in algorithm design and develop an optimal, randomized algorithm for linear classification under Gaussian subpopulations. Finally, to shed insight on auditing in general settings, we develop distribution-free lower bounds for direction estimation under general VC classes. This lower bound charts the query complexity that any optimal randomized auditing algorithms must attain.

1.1. Additional Notations

We now introduce some additional useful notation used throughout the paper. Let $[m]$ denote $\{1, ..., m\}$. For an unlabeled dataset S, and two classifiers h, h', we say $h(S) = h'(S)$ if for all $x \in S$, $h(x) = h'(x)$. Given a set of classifiers V and a labeled dataset T, define $V[T] := \left\{h \in V : \forall (x, y) \in T, h(x) = y\right\}$. Furthermore, denote by $V_x^y = V\left[\left\{(x, y)\right\}\right]$ for notational simplicity. Given a set of classifiers V and fairness measure μ, denote by $\text{diam}_\mu(V) := \max_{h,h' \in V} \mu(h) - \mu(h')$ the *μ-diameter* of V. Given a set of labeled examples T, denote by $\Pr_T(\cdot)$ the probability over the uniform distribution on T; given a classifier h, denote by $\text{err}(h, T) = \Pr_T(h(x) \neq y)$ the empirical error of h on T.

Throughout this paper, we will consider active fairness auditing under the membership query model, similar to membership query-based active learning (Angluin, 1988). Specifically, a deterministic active auditing algorithm $\mathcal{A}$ with label budget N is formally defined as a collection of $N + 1$ (computable) functions $f_1, f_2, \ldots, f_N, g$ such that:

1. For every $i \in [N]$, $f_i : (\mathcal{X} \times \mathcal{Y})^{i-1} \to \mathcal{X}$ is the label querying function used at step i, that takes into input the first $(i - 1)$ labeled examples $\langle (x_1, y_1), \ldots, (x_{i-1}, y_{i-1})\rangle$ obtained so far, and chooses the i-th example x_i for label query.

2. $g : (\mathcal{X} \times \mathcal{Y})^N \to \mathbb{R}$ is the estimator function that takes into input all N labeled examples $\langle (x_1, y_1), \ldots, (x_N, y_N)\rangle$ obtained throughout the interaction process, and outputs $\hat{\mu}$, the estimate of $\mu(h^*)$.

When $\mathcal{A}$ interacts with a target classifier h, let the resultant queried unlabeled dataset be $S_{\mathcal{A},h} = \langle x_1, \ldots, x_N\rangle$, and the final μ estimate be $\hat{\mu}_{\mathcal{A},h}$.

Similar to deterministic algorithms, a randomized active auditing algorithm $\mathcal{A}$ with label budget N and B bits of random seed is formally defined as a collection of $N + 1$ (computable) functions $f_1, \ldots, f_N, g$, where $f_i : (\mathcal{X} \times \mathcal{Y})^{i-1} \times \{0, 1\}^B \to \mathcal{X}$ and $g : (\mathcal{X} \times \mathcal{Y})^N \times \{0, 1\}^B \to \mathbb{R}$. Note that each function now take as input a B-bit random seed; as a result, when $\mathcal{A}$ interacts with a fixed h^*, its output $\hat{\mu}$ is now a random variable. Note also that under the above definition, a randomized active auditing algorithm $\mathcal{A}$ that uses a fixed seed b may be viewed as a deterministic active auditing algorithm $\mathcal{A}_b$.

We will be comparing our algorithms' query complexities with those of disagreement-based active learning algorithms (Cohn et al., 1994; Hanneke, 2014). Given a classifier h and $r > 0$, define $B(h, r) = \left\{h' \in \mathcal{H} : \Pr_{D_X}\left(h'(x) \neq h(x)\right) \leq r\right\}$ as the disagreement ball centered at h with radius r. Given a set of classifiers V, define its disagreement region $\text{DIS}(V) = \left\{x \in \mathcal{X} : \exists h, h' \in V : h(x) \neq h'(x)\right\}$. For a hypothesis class $\mathcal{H}$ and an unlabeled data distribution D_X, an important quantity that characterizes the query complexity of disagreement-based active learning algorithm is the *disagreement coefficient* $\theta(r)$, defined as

$$\theta(r) = \sup_{h \in \mathcal{H}, r' \geq r} \frac{\Pr_{D_X}(x \in \text{DIS}(B(h, r')))}{r'}.$$

2. Related Work

Our work is most related to the following two lines of work, both of which are concerned with estimating some property of a model without having to learn the model itself.

Sample-Efficient Optimal Loss Estimation: Dicker (2014); Kong & Valiant (2018) propose U-statistics-based estimators that estimate the optimal population mean square error in d-dimensional linear regression, with a sample complexity of $O(\sqrt{d})$ (much lower than $O(d)$, the sample complexity of learning optimal linear regressor). Kong & Valiant (2018) also extend the results to a well-specified logisitic regression setting, where the goal is to estimate the optimal zero-one loss. Our work is similar in focusing on the question of efficient $\mu(h^*)$ estimation without having to learn h^*. Our work differs in focusing on fairness property instead of the optimal MSE or zero-one loss. Moreover, our results apply to arbitrary $\mathcal{H}$, and not just to linear models.

Interactive Verification: Goldwasser et al. (2021) studies verification of whether a model h's loss is near-optimal with respect to a hypothesis class $\mathcal{H}$ and looks to understand when verification is cheaper than learning. They prove that verification is cheaper than learning for specific hypothesis classes and is just as expensive for other hypothesis classes. Again, our work differs in focusing on a different property of the model, fairness.

Our algorithm also utilizes tools from active learning and machine teaching, which we review below.

Active Learning and Teaching: The task of learning h^* approximately through membership queries has been well-studied (e.g. Angluin, 1988; Hegedűs, 1995; Dasgupta, 2005a; Hanneke, 2006; 2007). Our computationally efficient algorithm for active fairness auditing is built upon the connection between active learning and machine teaching (Goldman & Kearns, 1995), as first noted in Hegedűs (1995); Hanneke (2007). To achieve computational efficiency, our work builds on recent work on black-box teaching (Dasgupta et al., 2019), which implicitly gives an efficient procedure for computing an approximate-minimum specifying set; we adapt Dasgupta et al. (2019)'s algorithm to give a similar procedure for approximating the minimum specifying set that specifies the μ value.

In the interest of space, please see discussion of additional related work in Appendix A.

3. Manipulation-Proof Algorithms

3.1. Optimal Deterministic Algorithm

We begin our study of the MP estimation of $\mu(h^*)$ by identifying an optimal deterministic algorithm based on dynamic programming. Inspired by a minimax analysis of exact active learning with membership queries (Hanneke, 2006), we recursively define the following value function for any version space $V \subseteq \mathcal{H}$:

$$\mathrm{Cost}(V) = \begin{cases} 0, & \mathrm{diam}_\mu(V) \leq 2\epsilon \\ 1 + \min_x \max_y \mathrm{Cost}(V[(x,y)]), & \text{otherwise} \end{cases}$$

Note that $\mathrm{Cost}(V)$ is similar to the minimax query complexity of exact active learning (Hanneke, 2006), except that the induction base case is different – here the base case is $\mathrm{diam}_\mu(V) \leq 2\epsilon$, which implies that subject to $h^* \in V$, we have identified $\mu(h^*)$ up to error ϵ. In contrast, in exact active learning, Hanneke (2006)'s induction base case is $|V| = 1$, where we identify h^* through V.

The value function Cost also has a game-theoretic interpretation. Imagine that a learner plays a multi-round game with an adversary. The learner makes sequential queries of examples to obtain their labels, and the adversary reveals the labels of the examples, subject to the constraint that all labeled examples shown agree with some classifier in $\mathcal{H}$. The version space V encodes the state of the game: it is the set of classifiers that agrees with all the labeled examples shown so far in the game. The interaction between the learner and the adversary ends when all classifiers in V has μ values 2ϵ-close to each other. The learner would like to minimize its total cost, which is the number of rounds. $\mathrm{Cost}(V)$ can be viewed as the minimax-optimal future cost, subject to the game's current state being represented by version space V.

Algorithm 1 Minimax optimal deterministic auditing

Require: Finite hypothesis class $\mathcal{H}$, target error ϵ, fairness measure μ
Ensure: $\hat{\mu}$, an estimate of $\mu(h^*)$
1: Let $V \leftarrow \mathcal{H}$
2: **while** $\mathrm{diam}_\mu(V) > 2\epsilon$ **do**
3: Query $x \in \mathrm{argmin}_x \max_y \mathrm{Cost}\left(V_x^y\right)$, obtain label $h^*(x)$
4: $V \leftarrow V(h^*, \{x\})$
5: **return** $\frac{1}{2}\left(\max_{h\in V} \mu(h) + \min_{h\in V} \mu(h)\right)$

Based on the notion of Cost, we design an algorithm, Algorithm 1, that has a worst-case label complexity at most $\mathrm{Cost}(\mathcal{H})$. Specifically, it maintains a version space $V \subset \mathcal{H}$, initialized to $\mathcal{H}$ (line 1). At every iteration, if the μ-diameter of V, $\mathrm{diam}_\mu(V) = \max_{h,h'\in V} \mu(h) - \mu(h')$, is at most 2ϵ, then since $\mu(h^*) \in I = [\min_{h\in V} \mu(h), \max_{h\in V} \mu(h)]$ returning the midpoint of I gives us an ϵ-accurate estimate of $\mu(h^*)$ (line 5). Otherwise, Algorithm 1 makes a query by choosing the x that minimizes the worst-case future value functions (line 3). After receiving $h^*(x)$, it updates its version space V (line 4). By construction, the interaction between the learner and the labeler lasts for at most $\mathrm{Cost}(V)$ rounds, which gives the following theorem.

Theorem 3.1. *If Algorithm 1 interacts with some $h^* \in \mathcal{H}$, then it outputs $\hat{\mu}$ such that $\left|\hat{\mu} - \mu(h^*)\right| \leq \epsilon$, and queries at most $\mathrm{Cost}(\mathcal{H})$ labels.*

By the minimax nature of Cost, we also show that among all deterministic algorithms, Algorithm 1 has the optimal worst-case query complexity:

Theorem 3.2. *If $\mathcal{A}$ is a deterministic algorithm with query budget $N \leq \mathrm{Cost}(\mathcal{H}) - 1$, there exists some $h^* \in \mathcal{H}$, such that $\hat{\mu}$, the output of $\mathcal{A}$ after querying h^*, satisfies $\left|\hat{\mu} - \mu(h^*)\right| > \epsilon$.*

The proofs of Theorems 3.1 and 3.2 are deferred to Appendix D.1.

3.1.1. Comparison to Baselines

To gain a better understanding of $\mathrm{Cost}(\mathcal{H})$, we relate it to the label complexity of the two baselines, i.i.d sampling and active learning. To establish the comparison, we prove that we can derandomize existing i.i.d sampling-based and active learning-based auditing algorithms with a small overhead on label complexity.

Our first result is that the label complexity of Algorithm 1 is within a factor of $O(\ln|\mathcal{H}|)$ of the label complexity of i.i.d sampling.

Proposition 3.3. $\mathrm{Cost}(\mathcal{H}) \leq O\left(\frac{1}{\epsilon^2} \ln|\mathcal{H}|\right)$.

Our second result is that the label complexity of Algorithm 1 is always no worse than the distribution-dependent label complexity of CAL (Cohn et al., 1994; Hanneke, 2014), a well-known PAC active learning algorithm. We believe that similar bounds comparing $\mathrm{Cost}(\mathcal{H})$ to the complexity of generic active learning algorithms can also be shown; these algorithms include the Splitting Algorithm (Dasgupta, 2005b) or the confidence-based algorithm of Zhang & Chaudhuri (2014), through suitable derandomization procedures.

Proposition 3.4. $\mathrm{Cost}(\mathcal{H}) \leq O\left(\theta(\epsilon) \cdot \ln|\mathcal{H}| \cdot \ln\frac{1}{\epsilon}\right)$, *where θ is the disagreement coefficient of $\mathcal{H}$ with respect to D_X (recall Section 1.1 for its definition).*

Proof sketch. We present Algorithm 2, which is a derandomized version of the Phased CAL algorithm (Hsu, 2010, Chapter 2). To prove this proposition, using Theorem 3.2, it suffices to show that Algorithm 2 has a deterministic label complexity bound of $O\left(\theta(\epsilon) \cdot \ln|\mathcal{H}| \cdot \ln\frac{1}{\epsilon}\right)$. We only present the main idea here, and defer a precise version of the proof to Appendix D.3.

We first show that for every n, the optimization problem in line 5 is always feasible. To see this, observe that if we draw S_n, a sample of size m_n, drawn i.i.d from D_X, we have:

1. By Bernstein's inequality, with probability $1 - \frac{1}{4}$,

$$\Pr_{S_n}(x \in \mathrm{DIS}(V_n)) \leq 2\Pr_{D_X}(x \in \mathrm{DIS}(V_n)) + \frac{\ln 8}{m_n},$$

2. By Bernstein's inequality and union bound over $h, h' \in \mathcal{H}$, we have with probability $1 - \frac{1}{4}$,

$$\forall h, h' \in \mathcal{H}: \quad \Pr_S(h(x) \neq h'(x)) = 0 \implies \Pr_{D_X}(h(x) \neq h'(x)) \leq \frac{16\ln|\mathcal{H}|}{m_n}.$$

By union bound, with nonzero probability, the above two condition hold simultaneously, showing the feasibility of the optimization problem.

We then argue that for all n, $V_{n+1} \subseteq B(h^*, \frac{16\ln|\mathcal{H}|}{m_n})$. This is because for each $h \in V_{n+1}$, h and h^* are both in V_n and therefore they agree on $S_n \setminus T_n$; on the other hand, h and h^* agree on T_n by the definition of of V_{n+1}. As a consequence, $\Pr_{S_n}(h(x) \neq h^*(x)) = 0$, which implies that $\Pr_{D_X}(h(x) \neq h^*(x)) \leq \frac{16\ln|\mathcal{H}|}{m_n}$. As a consequence, for all $h \in V_{N+1}$, $\Pr(h(x) \neq h^*(x)) \leq \underline{p}\epsilon$, which, combined with Lemma C.1, implies that $\left|\mu(h) - \mu(h^*)\right| \leq \epsilon$.

Algorithm 2 Derandomized Phased CAL for Auditing

Require: Hypothesis class $\mathcal{H}$, target error ϵ, minority population proportion $\underline{p}$, fairness measure μ
Ensure: $\hat{\mu}$, an estimate of $\mu(h^*)$
1: Let $N = \lceil \log_2 \frac{16\ln|\mathcal{H}|}{\underline{p}\epsilon} \rceil$.
2: Let $V_1 \leftarrow \mathcal{H}$
3: **for** $n = 1, \ldots, N$ **do**
4: Let $m_n = 2^n$
5: Find (the lexicographically smallest) $S_n \in \mathcal{X}^{m_n}$ such that:

$$\Pr_{S_n}(x \in \mathrm{DIS}(V_n)) \leq 2\Pr_{D_X}(x \in \mathrm{DIS}(V_n)) + \frac{\ln 8}{m_n},$$

and

$$\forall h, h' \in \mathcal{H}: \quad \Pr_{S_n}(h(x) \neq h'(x)) = 0 \implies \Pr_{D_X}(h(x) \neq h'(x)) \leq \frac{16\ln|\mathcal{H}|}{m_n}.$$

6: Query h^* for the labels of examples in $T_n := S_n \cap \mathrm{DIS}(V_n)$
7: $V_{n+1} \leftarrow V_n(h^*, T_n)$.
8: **return** $\mu(h)$ for an arbitrary $h \in V_{N+1}$.

Finally, to upper bound Algorithm 2's label complexity:

$$\begin{aligned}\sum_{n=1}^{N}|T_n| &= \sum_{n=1}^{N} m_n \cdot (2\Pr_{D_X}(x \in \mathrm{DIS}(V_n)) + \frac{\ln 8}{m_n}) \\ &\leq \sum_{n=1}^{N} m_n \cdot (2\theta(\epsilon)\frac{16\ln|\mathcal{H}|}{m_n} + \frac{\ln 8}{m_n}) \\ &\leq O\left(\theta(\epsilon)\cdot\ln|\mathcal{H}|\cdot\ln\frac{1}{\epsilon}\right).\end{aligned}$$

□

3.1.2. Computational Hardness of Implementing Algorithm 1

Although Algorithm 1 attains the optimal label complexity of deterministic algorithms, we show in the following proposition that, under standard complexity-theoretic assumptions ($\mathrm{NP} \not\subseteq \mathrm{TIME}(n^{O(\log\log n)})$), even approximating $\mathrm{Cost}(\mathcal{H})$ is computationally intractable.

Proposition 3.5. *If there is an algorithm that can approximate $\mathrm{Cost}(\mathcal{H})$ to within $0.3\ln|\mathcal{H}|$ factor in $\mathrm{poly}(|\mathcal{H}|, |\mathcal{X}|, 1/\epsilon)$ time, then $\mathrm{NP} \subseteq \mathrm{TIME}(n^{O(\log\log n)})$.*

We remark that the constant 0.3 can be improved to a constant arbitrarily smaller than 1. The main insight behind this proposition is a connection between $\mathrm{Cost}(\mathcal{H})$ and optimal-depth decision trees (see Theorem D.4). Using the hardness of computing an approximately-optimal-depth decision

tree (Laber & Nogueira, 2004) and taking into account the structure of μ, we establish the intractability of approximating $\mathrm{Cost}(\mathcal{H})$.

Owing to the intractability of Algorithm 1, in the next section, we turn to the design of a computationally efficient algorithm whose label complexity nears that of Algorithm 1 (i.e. $\mathrm{Cost}(\mathcal{H})$).

3.2. Efficient Randomized Algorithm with Competitive Guarantees

We present our efficient algorithm in this section, which also serves as a first upper bound on the statistical complexity of computationally tractable algorithms. Our algorithm, Algorithm 3, is inspired by the exact active learning literature (Hegedűs, 1995; Hanneke, 2007), based on a connection between machine teaching (Goldman & Kearns, 1995) and active learning.

Algorithm 3 takes into input two oracles, a mistake-bounded online learning oracle $\mathcal{O}$ and an constrained empirical risk minimization (ERM) oracle C-ERM, defined below.

Definition 3.6. An *online-learning oracle* $\mathcal{O}$ is said to have a mistake bound of M for hypothesis class $\mathcal{H}$, if for any classifier $h^* \in \mathcal{H}$, and any sequence of examples $x_1, x_2, \ldots$, at every round $t \in \mathbb{N}$, given historical examples $(x_s, h^*(x_s))_{s=1}^{t-1}$, outputs classifier $\hat{h}_t$ such that $\sum_{t=1}^{\infty} I(\hat{h}_t(x_t) \neq h^*(x_t)) \leq M$.

Well-known implementations of mistake bounded online learning oracle include the halving algorithm and its efficient sampling-based approximations (Bertsimas & Vempala, 2004) as well as the Perceptron / Winnow algorithm (Littlestone, 1988; Ben-David et al., 2009). For instance, if $\mathcal{O}$ is the halving algorithm, a mistake bound of $M = \log_2 |\mathcal{H}|$ may be achieved.

We next define the constrained ERM oracle, which has been previously used in a number of works on oracle-efficient active learning (Dasgupta et al., 2007; Hanneke, 2011; Huang et al., 2015).

Definition 3.7. An *constrained ERM oracle* for hypothesis class $\mathcal{H}$, C-ERM, is one that takes as input labeled datasets A and B, and outputs a classifier $\hat{h} \in \operatorname{argmin} \{\mathrm{err}(h, A) : h \in \mathcal{H}, \mathrm{err}(h, B) = 0\}$.

The high-level idea of Algorithm 3 is as follows: at every iteration, it uses the mistake-bounded online learning oracle to generate some classifier $\hat{h}$ (line 3); then, it aims to construct a dataset T of small size, such that after querying h^* for the labels of examples in T, one of the following two happens: (1) $\hat{h}$ disagrees with h^* on some example in T; (2) for all classifiers in the version space $V = \{h \in \mathcal{H} : \forall x \in T, h(x) = h^*(x)\}$, we have $\mathrm{diam}_\mu(V) \leq 2\epsilon$. In case (1), we have found a counterexample for $\hat{h}$, which can be fed to the online learning oracle to learn a new model, and this can happen at most M times; in case (2), we are done: our queried labeled examples ensure that our auditing estimate is ϵ-accurate, and satisfies manipulation-proofness. Dataset T of such property is called a (μ, ϵ)-*specifying set* for $\hat{h}$, as formally defined in Defintion D.7 in Appendix D.5.

Another view of the μ-specifying set is a set T such that for all h, h' with $\mu(h) - \mu(h') > 2\epsilon$, there exists some $x \in T$, such that $h(x) \neq \hat{h}(x)$ or $h'(x) \neq \hat{h}(x)$. The requirements on T can be viewed as a set cover problem, where the universe U is $\{(h, h') \in \mathcal{H}^2 : \mu(h) - \mu(h') > 2\epsilon\}$, and the set system is $\mathcal{C} = \{C_x : x \in \mathcal{X}\}$, where (h, h') is in C_x if $h(x) \neq \hat{h}(x)$ or $h'(x) \neq \hat{h}(x)$.

This motivates us to design efficient set cover algorithms in this context. A key challenge of applying standard offline set cover algorithms (such as the greedy set cover algorithm) to construct approximate minimum (μ, ϵ)-specifying set is that we cannot afford to enumerate all elements in the universe U as U can be exponential in size.

In face of this challenge, we draw inspiration from online set cover literature (Alon et al., 2009; Dasgupta et al., 2019) to design an oracle-efficient algorithm that computes $O(\log |\mathcal{H}| \log |\mathcal{X}|)$-*approximate* minimum (μ, ϵ)-specifying sets, which avoids enumeration over U.

Our key idea is to simulate an online set cover process. We build the cover set[3] T iteratively, starting from $T = \emptyset$ (line 4). At every inner iteration, we first try to find a pair (h_1, h_2) in U not yet covered by the current T. As we shall see next, this step (line 7) can be implemented efficiently given the constrained ERM oracle C-ERM. If such a pair (h_1, h_2) can be found, we use the online set cover algorithm implicit in (Dasgupta et al., 2019) to find a new example that covers this pair, add it to T, and move onto the next iteration (lines 11 to 14). Otherwise, T has successfully covered all the elements in U, in which case we break the inner loop (line 9).

To see how line 7 finds an uncovered pair in U, we note that it can be also written as:

$$(h_1, h_2) = \operatorname*{argmax}_{h, h' \in \mathcal{H}} \left\{ \mu(h) - \mu(h') : h(T) = h'(T) = \hat{h}(T) \right\}$$

Thus, if $\mu(h_1) - \mu(h_2) > 2\epsilon$, then the returned pair (h_1, h_2) corresponds to a pair in universe U that is not covered by T. Otherwise, by the optimality of (h_1, h_2), T covers all elements in U.

Furthermore, we note that optimization problems (1) and (2) can be implemented with access to C-ERM. We show this for program (1) and the reasoning for program (2) is

[3] When it is clear from context, we slightly abuse notations and say "x covers (h, h')" if $(h, h') \in C_x$.

analogous. Observe that maximizing $\mu(h)$ from $h \in \mathcal{H}$ subject to constraint $h(T) = \hat{h}(T)$ is equivalent to minimizing (a weighted) empirical error of $h \in \mathcal{H}$ on dataset $\{(x, +1) : x \in \mathcal{X}, x_A = 0\} \cup \{(x, -1) : x \in \mathcal{X}, x_A = 1\}$, subject to h having zero error on $\{(x, \hat{h}(x)) : x \in T\}$.

We are now ready to present the label complexity guarantee of Algorithm 3.

Algorithm 3 Oracle-efficient Active Fairness Auditing

Require: Hypothesis class $\mathcal{H}$, online learning oracle $\mathcal{O}$ with mistake bound M, constrained ERM oracle C-ERM, target error ϵ, fairness measure μ.

Ensure: $\hat{\mu}$, an estimate of $\mu(h^*)$

Initialize $S \leftarrow \emptyset$
while true do
  $\hat{h} \leftarrow \mathcal{O}(S)$
  Let $T \leftarrow \emptyset$
  //Computing an approximate minimum (μ, ϵ)-specifying set for $\hat{h}$
  Initialize weights $w(x) = \frac{1}{|\mathcal{X}|}$ and threshold $\tau_x \sim \text{Exponential}(\ln(|\mathcal{H}|^2 M/\delta))$ //random initialization of thresholds
  while true do
    Use C-ERM to solve separate programs:

$$h_1 \leftarrow \text{find } \max_{h \in \mathcal{H}} \mu(h), \text{s.t. } h(T) = \hat{h}(T) \quad (1)$$

    and

$$h_2 \leftarrow \text{find } \min_{h \in \mathcal{H}} \mu(h), \text{s.t. } h(T) = \hat{h}(T) \quad (2)$$

    //T is an (μ, ϵ)-specifying set for $\hat{h}$
    if $\mu(h_1) - \mu(h_2) \leq 2\epsilon$ **then**
      break
    else
      //Add examples to T to cover (h_1, h_2), using the online set cover algorithm implicit in (Dasgupta et al., 2019)
      Determine $\Delta(h_1, h_2) = \{x \in \mathcal{X} : h_1(x) \neq \hat{h}(x) \text{ or } h_2(x) \neq \hat{h}(x)\}$
      while $\sum_{x \in \Delta(h_1, h_2)} w(x) \leq 1$ **do**
        Double weights $w(x)$ for all x in $\Delta(h_1, h_2)$
        Update $T \leftarrow \{x \in \mathcal{X} : w(x) \geq \tau_x\}$
  Query h^* on T
  $S \leftarrow S \cup T$
  if $\hat{h}(T) = h^*(T)$ **then**
    return $\frac{1}{2}(\mu(h_1) + \mu(h_2))$

Theorem 3.8. *If the online learning oracle $\mathcal{O}$ makes a total of M mistakes, then with probability $1 - \delta$, Algorithm 3 outputs $\hat{\mu}$ such that $\left|\hat{\mu} - \mu(h^*)\right| \leq \epsilon$, with its number of label queries bounded by:*

$$O\left(\text{Cost}(\mathcal{H}) M \log \frac{|\mathcal{H}|M}{\delta} \log |\mathcal{X}|\right).$$

The proof of Theorem 3.8 is deferred to Appendix D.5. In a nutshell, it combines the following observations. First, Algorithm 3 has at most M outer iterations using the mistake bound guarantee of oracle $\mathcal{O}$. Second, for each $\hat{h}$ in each inner iteration, its minimum (μ, ϵ)-specifying set has size at most $\text{Cost}(\mathcal{H})$; this is based on a nontrivial connection between the optimal deterministic query complexity and (μ, ϵ)-extended teaching dimension (see Definition D.9), which we present in Lemma D.10. Third, by the $O\left(\log \frac{|\mathcal{H}|M}{\delta} \log |\mathcal{X}|\right)$-approximation guarantee of the online set cover algorithm implicit in (Dasgupta et al., 2019), each outer iteration makes at most $O\left(\text{Cost}(\mathcal{H}) \log \frac{|\mathcal{H}|M}{\delta} \log |\mathcal{X}|\right)$ label queries.

Remark 3.9. Via an argument similar to that in Proposition 3.5, we can show that, for computationally-efficient algorithms, the approximation factor in constructing an approximately-minimum (μ, ϵ)-specifying set for $\hat{h}$ cannot be significantly improved to, say, $0.3 \ln |\mathcal{H}|$.

Finally, in Appendix F, we empirically explore the performance of Algorithm 3 and active learning, and compare them with i.i.d sampling. As expected, our experiments confirm that under a fixed budget, Algorithm 3 is most effective at inducing a version space with a small μ-diameter, and can thus provide the strongest manipulation-proofness guarantee.

4. Statistical Limits of Estimation

In this section, we turn to direct estimation, the second of the two main guarantees one may wish to have for auditing. In particular, we focus on the statistical limits of direct estimation, which involves designing an efficient auditing algorithm that can output $\hat{\mu}$ such that $\left|\hat{\mu} - \mu(h^*)\right| \leq \epsilon$ with a small number of queries.

4.1. Separation between Estimation with and without Manipulation-proofness

To start, it is natural to contrast the guarantee of ϵ-manipulation-proofness against ϵ-direct estimation accuracy. Indeed, if the two guarantees are one and the same, we may simply use the MP estimation algorithms for direct estimation as well.

More specifically, we look to answer the question of whether achieving MP is strictly harder, and we answer this question in the affirmative. Indeed, following simple example suggests that MP estimation can sometimes require a much higher label complexity than direct estimation.

Example 4.1. Let $\epsilon = \frac{1}{4}$ and $n \gg 1$. $\mathcal{X} = \{0, 1, \ldots, n\}$, and $x \mid x_A = 0 \sim \text{Uniform}(\{0\})$, and $x \mid x_A = 1 \sim \text{Uniform}(\{1, \ldots, n\})$. Let $\mathcal{H} = \{h : \mathcal{X} \to \{-1, +1\}, h(0) = -1\}$.

First, as $\epsilon = \frac{1}{4}$, the iid sampling baseline makes $O(1)$ queries and ensures that it estimates $\mu(h^*)$ with error at most ϵ with probability ≥ 0.9.

However, for manipulation-proof estimation, at least $\Omega(n)$ labels are needed to ensure that the queried dataset S satisfies $\text{diam}_\mu(\mathcal{H}(h^*, S)) \leq \epsilon$. Indeed, let $h^* \equiv -1$. For any unlabeled dataset S of size $\leq n/2$, by the definition of $\mathcal{H}$, there always exist $h, h' \in \mathcal{H}(h^*, S)$, such that for all $x \in \{1, \ldots, n\} \setminus S$, $h(x) = -1$ and $h'(x) = +1$. As a result, $\mu(h) = \frac{0}{n} - \frac{0}{1} = 0$, and $\mu(h') = \frac{|\{1,\ldots,n\} \setminus S|}{n} - \frac{0}{1} \geq \frac{1}{2}$, which implies that $\text{diam}_\mu(\mathcal{H}(h^*, S)) \geq \frac{1}{2} > \epsilon$. □

4.2. Randomized Algorithms for Direct Estimation

The separation result above suggests that different algorithms may be needed if we are *only* interested in efficient direct estimation. Motivated by our previous exploration, a first question to answer is whether randomization should be a key ingredient in algorithm design. That is, can a randomized algorithm achieve query complexity smaller than that of the optimal deterministic algorithm? Through the example below, we answer this question in the affirmative.

Example 4.2. Same as the setting of Example 4.1; recall that iid sampling, a randomized algorithm, estimates $\mu(h^*)$ with error at most $\epsilon = \frac{1}{4}$ with probability ≥ 0.9; it has a query complexity of $O(1)$.

In contrast, consider any deterministic algorithm $\mathcal{A}$ with label budget $N \leq \frac{n}{2}$; we consider its interaction history with classifier $h_0 \equiv -1$, which can be summarized by a sequence of unlabeled examples $S = \langle x_1, \ldots, x_N \rangle$. Now, consider an alternative classifier h_1 such that $h_1(x) = -1$ on $S \cup \{0\}$, but $h_1(x) = +1$ on $\{1, \ldots, n\} \setminus S$. By an inductive argument, it can be shown that the interaction history between $\mathcal{A}$ and h_1 is also S, which implies that when the underlying hypotheses $h^* = h_0$ and $h^* = h_1$, $\mathcal{A}$ must output the same estimate $\hat{\mu}$ (see Lemma B.1 in Appendix B for a formal proof); however, $\mu(h_0) - \mu(h_1) \geq \frac{1}{2}$, implying that under at least one of the two hypotheses, we must have $\left|\hat{\mu} - \mu(h^*)\right| \geq \frac{1}{4} = \epsilon$.

In summary, in this setting, a randomized algorithm has a query complexity of $O(1)$, much smaller than $\Omega(n)$, the optimal query complexity of deterministic algorithms. □

4.3. Case Study: Non-homogeneous Linear Classifiers under Gaussian Populations

In this subsection, we identify a practically-motivated setting where we are able to comprehensively characterize the minimax (randomized) active fairness auditing query complexity up to logarithmic factors. Specifically, we present a positive result in the form of an algorithm that has a query complexity of $\tilde{O}\left(\min(d, \frac{1}{\epsilon^2})\right)$ as well as a matching lower bound that shows any (possibly randomized) algorithm must have a query complexity of $\Omega\left(\min(d, \frac{1}{\epsilon^2})\right)$.

Example 4.3. Let $d \geq 2$ and $\mathcal{X} = \mathbb{R}^d$. $x \mid x_A = 0 \sim \text{N}(m_0, \Sigma_0)$, whereas $x \mid x_A = 1 \sim \text{N}(m_1, \Sigma_1)$. Let hypothesis class $\mathcal{H}_{\text{lin}} = \left\{h_{a,b}(x) := \text{sign}(\langle a, x \rangle + b) : a \in \mathbb{R}^d, b \in \mathbb{R}\right\}$ be the class of non-homogenenous linear classifiers.

Recall that i.i.d sampling has a label complexity of $O\left(\frac{1}{\epsilon^2}\right)$. On the other hand, through a membership query-based active learning algorithm (Algorithm 6 in Appendix E.2), we can approximately estimate $\mu(h^*)$ (up to scaling) by doing d-binary searches, using active label queries. This approach incurs a total label complexity of $\tilde{O}(d)$. Choosing the better of these two algorithms gives an active fairness auditing strategy of label complexity $\tilde{O}\left(\min(d, \frac{1}{\epsilon^2})\right)$.

We only present the main idea of Algorithm 6 here, with its full analysis deferred to Appendix E.2. Its core component is Algorithm 4 below, which label-efficiently estimates $\gamma(h^*) = \mathbb{P}_{x \sim \text{N}(0, I_d)}(h^*(x) = +1)$, with black-box label queries to $h^*(x) = \text{sign}(\langle a^*, x \rangle + b^*)$. Algorithm 4 is based on the following insights. First, observe that $\gamma(h^*) = \Phi\left(\frac{b^*}{\|a^*\|_2}\right) =: \Phi(sr)$, where Φ is the standard normal CDF, $s := \text{sign}(b^*)$, and $r := \sqrt{\frac{1}{\sum_{i=1}^d m_i^{-2}}}$, for $m_i := -\frac{b^*}{a_i^*}$. On the one hand, s can be easily obtained by querying h^* on $\vec{0}$ (line 2). On the other hand, estimating r can be reduced to estimating each m_i. However, some m_i's can be unbounded, which makes their estimation challenging. To get around this challenge, we prove the following lemma, which shows that it suffices to accurately estimate those m_i's that are not unreasonably large (i.e. m_i's for $i \in S$, defined below):

Lemma 4.4. *Let* $\alpha := \sqrt{2d \ln \frac{1}{\epsilon}}$ *and* $\beta := 2d^{\frac{5}{2}} (\ln \frac{1}{\epsilon})^{\frac{3}{4}} (\frac{1}{\epsilon})^{\frac{1}{2}}$. *Suppose* $r \leq \alpha$. *If there is some* $S \subset [d]$, *such that:*

1. *for all* $i \notin S, |m_i| \geq \beta$,
2. *for all* $i \in S$, $|\hat{m}_i - m_i| \leq \epsilon$;

then, $\left| \sqrt{\frac{1}{\sum_{i \in S} \hat{m}_i^{-2}}} - r \right| \leq 2\epsilon$.

Algorithm 4 carefully utilizes this lemma to estimate r. First, it tests whether for all i, $h^*(\alpha e_i) = h^*(-\alpha e_i)$; if yes, for all i, $|m_i| \geq \alpha$, and $r \geq \sqrt{\ln \frac{1}{\epsilon}}$, and $\gamma(h^*)$ is ϵ-close to 0 or 1 depending on the value of s (line 5). Otherwise, it

must be the case that $r \leq \alpha$. In this case, we go over each coordinate i, first testing whether $|m_i| \leq \beta$ (line 8); if no, we skip this coordinate (do not add it to S); otherwise, we include i in S and estimate m_i to precision ϵ using binary search (line 11). By the guarantees of Lemma 4.4, we have $|s\hat{r} - sr| \leq 2\epsilon$, which, by the $\frac{1}{\sqrt{2\pi}}$-Lipschitzness of Φ, implies that $\left|\hat{\gamma} - \gamma(h^*)\right| \leq \epsilon$. The total query complexity of Algorithm 4 is $1 + 2d + 2d + d\log_2 \frac{\beta}{\epsilon} = \tilde{O}(d)$.

Algorithm 4 ESTIMATE-POSITIVE: A label efficient estimation algorithm for $\gamma(h^*)$ for non-homogenoeus linear classifiers

Require: query access to $h^* \in \mathcal{H}_{\text{lin}}$, target error ϵ.
Ensure: $\hat{\gamma}$ such that $\left|\hat{\gamma} - \gamma(h^*)\right| \leq \epsilon$.

1: Let $\alpha = \sqrt{2d\ln\frac{1}{\epsilon}}$, $\beta = 2d^{\frac{5}{2}}(\ln\frac{1}{\epsilon})^{\frac{3}{4}}(\frac{1}{\epsilon})^{\frac{1}{2}}$.
2: $s \leftarrow$ Query h^* on $\vec{0}$
3: Query h^* on $\{\rho\alpha e_i : \rho \in \{\pm 1\}, i \in [d]\}$
4: **if** for all $i \in [d]$, $h^*(\alpha e_i) = h^*(-\alpha e_i)$ **then**
5: **return** 1 if $s = +1$, 0 if $s = -1$
//Otherwise, $r \leq \alpha = \sqrt{2d\ln\frac{1}{\epsilon}}$
6: $S \leftarrow \emptyset$
7: **for** $i = 1, \ldots, d$ **do**
8: Query h^* on βe_i and $-\beta e_i$
9: **if** $h^*(\beta e_i) \neq h^*(-\beta e_i)$ **then**
10: $S \leftarrow S \cup \{i\}$
//Use binary search to obtain $\hat{m}_i$, an estimate of $m_i = -\frac{b^*}{a_i^*}$ with precision ϵ
11: $\hat{m}_i \leftarrow$ BINARY-SEARCH(i, β, ϵ) (Algorithm 5)
12: $\hat{r} \leftarrow \sqrt{\frac{1}{\sum_{i\in S}\hat{m}_i^{-2}}}$ //$\hat{r}$ is an estimate of r
13: **return** $\Phi(s\hat{r})$

Algorithm 5 BINARY-SEARCH

Require: i, β such that $h^*(\beta e_i) \neq h^*(-\beta e_i)$, precision ϵ
Ensure: m, an ϵ-accurate estimate of $m_i = -\frac{b}{a_i}$

1: $u \leftarrow \beta, l \leftarrow -\beta$
2: **while** $u - l \geq \epsilon$ **do**
3: $m \leftarrow \frac{u+l}{2}$
4: Query h^* on me_i
5: **if** $h^*(me_i) = h^*(le_i)$ **then**
6: $l \leftarrow m$
7: **else**
8: $u \leftarrow m$
9: **return** m

For the lower bound, we formulate a hypothesis testing problem, such that under hypotheses H_0 and H_1, the $\mu(h^*)$ values are approximately ϵ-separated. This is used to show that any active learning algorithm with label query budget $\leq \Omega\left(\min(d, \frac{1}{\epsilon^2})\right)$ cannot effectively distinguish H_0 and H_1. Our construction requires a delicate analysis on the KL divergence between the observation distributions under the two hypotheses, and we refer the readers to Theorem E.3 for details. □

4.4. General Distribution-Free Lower Bounds

Finally, in this subsection, we move beyond the Gaussian population setting and derive general query complexity lower bounds for randomized estimation algorithms that audit general hypothesis classes with finite VC dimension d. This result suggests that, when $d \gg \frac{1}{\epsilon^2}$, or equivalently $\epsilon \gg \frac{1}{\sqrt{d}}$, there exists some hard data distribution and target classifier in $\mathcal{H}$, such that active fairness auditing has a query complexity lower bound of $\Omega(\frac{1}{\epsilon^2})$. Put another way, iid sampling is near-optimal.

Theorem 4.5 (Lower bound for randomized auditing)**.** *Fix $\epsilon \in (0, \frac{1}{40}]$ and a hypothesis class $\mathcal{H}$ with VC dimension $d \geq 1600$. For any (possibly randomized) algorithm $\mathcal{A}$ with label budget $N \leq O(\min(d, \frac{1}{\epsilon^2}))$, there exists a distribution D_X over $\mathcal{X}$ and $h^* \in \mathcal{H}$, such that $\mathcal{A}$'s output $\hat{\mu}$ when interacting with h^*, satisfies:*

$$\mathbb{P}\left(\left|\hat{\mu} - \mu(h^*)\right| > \epsilon\right) > \frac{1}{8}$$

The proof of Theorem 4.5 can be found at Appendix E.1. The lower bound construction follows from a similar setting as in Example 4.1, except that we now choose h^* in a randomized fashion.

5. Conclusion

In this paper, we initiate the theoretical study of query-efficient algorithms that audit ML models. We focus on auditing demographic parity, one of the canonical fairness notions. We investigate the natural auditing guarantee of direct estimation accuracy, and introduce a new guarantee based on the possibility of post-audit manipulation: manipulation-proofness. We identify an optimal deterministic algorithm, a matching randomized algorithm and develop upper and lower bounds that mark the performance that any optimal auditing algorithm must meet. Our first exploration of active fairness estimation seeks to provide a more complete picture of the theory of auditing. A natural next direction is to explore guarantees for other fairness notions such as equalized odds. Indeed, how does one construct query-efficient algorithms when μ is a function of both $h^*(x)$ and y? Another natural question, motivated by the connection to disagreement-based active learning, is to design active fairness auditing algorithms based on some notion of disagreement with respect to μ.

Acknowledgments. We thank Stefanos Poulis for sharing the implementation of the black-box teaching algorithm of Dasgupta et al. (2019), and special thanks to Steve Hanneke and Sanjoy Dasgupta for helpful discussions. We also thank the anonymous ICML reviewers for their feedback.

References

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

Alon, N., Awerbuch, B., Azar, Y., Buchbinder, N., and Naor, J. The online set cover problem. *SIAM Journal on Computing*, 39(2):361–370, 2009.

Angluin, D. Queries and concept learning. *Machine learning*, 2(4):319–342, 1988.

Balcan, M.-F. and Long, P. Active and passive learning of linear separators under log-concave distributions. In *Conference on Learning Theory*, pp. 288–316. PMLR, 2013.

Balcan, M.-F., Blais, E., Blum, A., and Yang, L. Active property testing. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pp. 21–30. IEEE, 2012.

Ben-David, S., Pál, D., and Shalev-Shwartz, S. Agnostic online learning. In *COLT*, volume 3, pp. 1, 2009.

Bertsimas, D. and Vempala, S. Solving convex programs by random walks. *Journal of the ACM (JACM)*, 51(4): 540–556, 2004.

Blais, E., Ferreira Pinto Jr, R., and Harms, N. Vc dimension and distribution-free sample-based testing. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 504–517, 2021.

Blanc, G., Gupta, N., Lange, J., and Tan, L.-Y. Estimating decision tree learnability with polylogarithmic sample complexity. *Advances in Neural Information Processing Systems*, 33, 2020.

Blum, A. and Hu, L. Active tolerant testing. In *Conference On Learning Theory*, pp. 474–497. PMLR, 2018.

Cohn, D., Atlas, L., and Ladner, R. Improving generalization with active learning. *Machine learning*, 15(2): 201–221, 1994.

Dasgupta, S. Analysis of a greedy active learning strategy. *Advances in neural information processing systems*, 17: 337–344, 2005a.

Dasgupta, S. Coarse sample complexity bounds for active learning. In *NIPS*, volume 18, pp. 235–242, 2005b.

Dasgupta, S., Hsu, D. J., and Monteleoni, C. A general agnostic active learning algorithm. *Advances in Neural Information Processing Systems*, 20:353–360, 2007.

Dasgupta, S., Hsu, D., Poulis, S., and Zhu, X. Teaching a black-box learner. In *International Conference on Machine Learning*, pp. 1547–1555. PMLR, 2019.

Dicker, L. H. Variance estimation in high-dimensional linear models. *Biometrika*, 101(2):269–284, 2014.

Feige, U. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

Goldman, S. A. and Kearns, M. J. On the complexity of teaching. *Journal of Computer and System Sciences*, 50 (1):20–31, 1995.

Goldreich, O., Goldwasser, S., and Ron, D. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998.

Goldwasser, S., Rothblum, G. N., Shafer, J., and Yehudayoff, A. Interactive proofs for verifying machine learning. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

Hanneke, S. The cost complexity of interactive learning. *Unpublished manuscript*, 2006.

Hanneke, S. Teaching dimension and the complexity of active learning. In *International Conference on Computational Learning Theory*, pp. 66–81. Springer, 2007.

Hanneke, S. Rates of convergence in active learning. *The Annals of Statistics*, pp. 333–361, 2011.

Hanneke, S. Theory of active learning. *Foundations and Trends in Machine Learning*, 7(2-3), 2014.

Hegedűs, T. Generalized teaching dimensions and the query complexity of learning. In *Proceedings of the eighth annual conference on Computational learning theory*, pp. 108–117, 1995.

Hotten, R. Volkswagen: The scandal explained. *BBC News*, 2015. URL https://www.bbc.com/news/business-34324772.

Hsu, D. J. *Algorithms for active learning*. PhD thesis, UC San Diego, 2010.

Huang, T.-K., Agarwal, A., Hsu, D. J., Langford, J., and Schapire, R. E. Efficient and parsimonious agnostic active learning. *Advances in Neural Information Processing Systems*, 28, 2015.

Kong, W. and Valiant, G. Estimating learnability in the sublinear data regime. *Advances in Neural Information Processing Systems*, 31:5455–5464, 2018.

Laber, E. S. and Nogueira, L. T. On the hardness of the minimum height decision tree problem. *Discrete Applied Mathematics*, 144(1-2):209–212, 2004.

Larson, J., Mattu, S., Kirchner, L., and Angwin, J. How we analyzed the compas recidivism algorithm. *ProPublica (5 2016)*, 9(1):3–3, 2016.

Littlestone, N. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988.

McCarthy, D. To regulate ai, try playing in a sandbox. *Emerging Tech Brew*, 2021. URL https://www.morningbrew.com/emerging-tech/stories/2021/05/26/regulate-ai-just-play-sandbox.

Mitchell, T. M. Generalization as search. *Artificial intelligence*, 18(2):203–226, 1982.

Rastegarpanah, B., Gummadi, K., and Crovella, M. Auditing black-box prediction models for data minimization compliance. *Advances in Neural Information Processing Systems*, 34, 2021.

Ron, D. *Property testing: A learning theory perspective*. Now Publishers Inc, 2008.

Sabato, S., Sarwate, A. D., and Srebro, N. Auditing: active learning with outcome-dependent query costs. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 1*, pp. 512–520, 2013.

Valiant, L. G. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Xu, Z., Yu, T., and Sra, S. Towards efficient evaluation of risk via herding. *Negative Dependence: Theory and Applications in Machine Learning*, 2019.

Zhang, C. and Chaudhuri, K. Beyond disagreement-based agnostic active learning. *Advances in Neural Information Processing Systems*, 27, 2014.

A. Additional Related Works

Property Testing: Our notion of auditing that leverages knowledge of $\mathcal{H}$ is similar in theme to the topic of property testing (Goldreich et al., 1998; Ron, 2008; Balcan et al., 2012; Blum & Hu, 2018; Blanc et al., 2020; Blais et al., 2021) which tests whether h^* is in $\mathcal{H}$, or h^* is far away from any classifier in $\mathcal{H}$, given query access to h^*. These works provide algorithms with testing query complexity of lower order than sample complexity for learning with respect to $\mathcal{H}$, for specific hypothesis classes such as monomials, DNFs, decision trees, linear classifiers, etc. Our problem can be reduced to property testing by testing whether h^* is in $\left\{h \in \mathcal{H} : \mu(h) \in [i\epsilon, (i+1)\epsilon]\right\}$ for all $i \in \left\{0, 1, \ldots, \lceil \frac{1}{\epsilon} \rceil\right\}$; however, to the best of our knowledge, no such result is known in the context of property testing.

Feature Minimization Audits: Rastegarpanah et al. (2021) study another notion of auditing, focusing on assessing whether the model is trained inline with the GDPR's Data Minimization principle. Specifically, this work evaluates the necessity of each individual feature used in the ML model, and this is done by imputing each feature with constant values and checking the extent of variation in the predictions. One commonality with our work, and indeed across all auditing works, is the concern with minimizing the number queries needed to conduct the audit.

Herding for Sample-efficient Mean Estimation: Additionally, the estimation of DP may be viewed as estimating the difference of two means. Viewed in this light, herding (Xu et al., 2019) offers a way to use non-iid sampling to more efficiently estimate means. However, the key difference needed in herding is that h^*, whose output is $\{-1, 1\}$, may be well-approximated by $\langle w, \phi(x)\rangle$ for some mapping ϕ known apriori.

Comparison with Sabato et al. (2013): Lastly, Sabato et al. (2013) also uses the term "auditing" in the context of active learning with outcome-dependent query costs; although the term "auditing" is shared, our problem settings are completely different: (Sabato et al., 2013) focuses on active learning the model h^* as opposed to just estimating $\mu(h^*)$.

B. A General Lemma on Deterministic Query Learning

In this section, we present a general lemma inspired by Hanneke (2007), which are used in our proofs for establishing lower bounds on deterministic active fairness auditing algorithms.

Lemma B.1. *If an deterministic active auditing algorithm $\mathcal{A}$ with label budget N interacts with labeling oracle that uses classifier h_0, and generates the following interaction history: $\langle (x_1, h_0(x_1)), (x_2, h_0(x_2)), \ldots, (x_N, h_0(x_N))\rangle$, and there exists a classifier h_1 such that $h_1(x) = h_0(x)$ for all $x \in \{x_1, \ldots, x_N\}$. Then $\mathcal{A}$, when interacting with h_1, generates the same interaction history, and outputs the same auditing estimate; formally, $S_{\mathcal{A},h_1} = S_{\mathcal{A},h_0}$ and $\hat{\mu}_{\mathcal{A},h_1} = \hat{\mu}_{\mathcal{A},h_0}$.*

Proof. Recall from Section 1.1 that deterministic active auditing algorithm $\mathcal{A}$ can be viewed as a sequence of $N+1$ functions $f_1, f_2, \ldots, f_N, g$, where $\{f_i\}_{i=1}^N$ are the label query function used at each iteration, and g is the final estimator function. We show by induction that for steps $i = 0, 1, \ldots, N$, the interaction histories of $\mathcal{A}$ with h_0 and h_1 agree on their first i elements.

Base case. For step $i = 0$, both interaction histories are empty and agree trivially.

Inductive case. Suppose that the statement holds for step i, i.e. $\mathcal{A}$, when interacting with both h_0 and h_1, generates the same set of labeled examples

$$S_i = \langle (x_1, y_1), \ldots, (x_i, y_i)\rangle,$$

up to step i.

Now, at step $i+1$, $\mathcal{A}$ applies the query function f_{i+1} and queries the same example $x_{i+1} = f_{i+1}(S_i)$. By assumption of this lemma, $h_1(x_{i+1}) = h_0(x_{i+1})$, which implies that the $(i+1)$-st labeled example obtained when $\mathcal{A}$ interacts with h_1, $(x_{i+1}, h_1(x_{i+1}))$ is identical to $(x_{i+1}, h_1(x_{i+1}))$, the $(i+1)$-st example when $\mathcal{A}$ interacts with h_0. Combined with the inductive hypotheses that the two histories agree on the first i examples, we have shown that $\mathcal{A}$, when interacting with h_0 and h_1, generates the same set of labeled examples

$$S_{i+1} = \langle (x_1, y_1), \ldots, (x_i, y_i), (x_{i+1}, y_{i+1})\rangle$$

up to step $i+1$.

This completes the induction.

As the interaction histories $\mathcal{A}$ with h_0 and h_1 are identical, the unlabeled data part of the history are identical, formally, $S_{\mathcal{A},h_1} = S_{\mathcal{A},h_0}$. In addition, as in both interactive processes, $\mathcal{A}$ applies deterministic function g to the same interaction history of length N to obtain estimate $\hat{\mu}$, we have $\hat{\mu}_{\mathcal{A},h_1} = \hat{\mu}_{\mathcal{A},h_0}$. □

C. Deferred Materials from Section 1

The following lemma formalizes the idea that PAC learning with $O(\epsilon)$ error is sufficient for fairness auditing, given that $\underline{p} = \min\left(\Pr_{D_X}(x_A = 0), \Pr_{D_X}(x_A = 1)\right)$ is $\Omega(1)$.

Lemma C.1. *If h is such that $\mathbb{P}(h(x) \neq h^*(x)) \leq \alpha$, then $\left|\mu(h) - \mu(h^*)\right| \leq \frac{\alpha}{\underline{p}}$.*

Proof. First observe that

$$\begin{aligned}
&\left|\Pr(h(x) = +1 \mid x_A = 0) - \Pr(h^*(x) = +1 \mid x_A = 0)\right| \\
\leq &\Pr(h(x) \neq h^*(x) \mid x_A = 0) \\
= &\frac{\Pr(h(x) \neq h^*(x), x_A = 0)}{\Pr(x_A = 0)} \\
\leq &\frac{\Pr(h(x) \neq h^*(x), x_A = 0)}{\underline{p}},
\end{aligned}$$

where the first inequality is by triangle inequality; the second inequality is by the definition of $\underline{p}$. Symmetrically, we have $\left|\Pr(h(x) = +1 \mid x_A = 1) - \Pr(h^*(x) = +1 \mid x_A = 1)\right| \leq \frac{\Pr(h(x) \neq h^*(x), x_A = 1)}{\underline{p}}$. Adding up the two inequalities, we have:

$$\begin{aligned}
&\left|\mu(h) - \mu(h^*)\right| \\
\leq &\left|\Pr(h(x) = +1 \mid x_A = 0) - \Pr(h^*(x) = +1 \mid x_A = 0)\right| + \left|\Pr(h(x) = +1 \mid x_A = 1) - \Pr(h^*(x) = +1 \mid x_A = 1)\right| \\
\leq &\frac{\Pr(h(x) \neq h^*(x), x_A = 0)}{\underline{p}} + \frac{\Pr(h(x) \neq h^*(x), x_A = 1)}{\underline{p}} \\
= &\frac{\Pr(h(x) \neq h^*(x))}{\underline{p}} \leq \frac{\alpha}{\underline{p}}.
\end{aligned}$$

□

D. Deferred Materials from Section 3

D.1. Proof of Theorems 3.1 and 3.2

Proof of Theorem 3.1. Suppose Algorithm 1 (denoted as $\mathcal{A}$ throughout the proof) interacts with some target classifier $h^* \in \mathcal{H}$.

We will show the following claim: at any stage of $\mathcal{A}$, if the set of labeled examples L shown so far induces a version $V = \mathcal{H}[L]$, then $\mathcal{A}$ will subsequently query at most $\mathrm{Cost}(V)$ more labels before exiting the while loop.

Note that Theorem 3.1 follows from this claim by taking $L = \emptyset$ and $V = \mathcal{H}$: after $\mathrm{Cost}(\mathcal{H})$ label queries, it exits the while loop, which implies that, the queried unlabeled examples $S_{\mathcal{A},h^*}$ induces version space $V' = \mathcal{H}(h^*, S_{\mathcal{A},h^*})$ with

$$\max_{h \in V'} \mu(h) - \min_{h \in V'} \mu(h) = \mathrm{diam}_\mu(V') \leq 2\epsilon.$$

Also, note that $h^* \in V'$; this implies that $\mu(h^*) \in [\min_{h \in V'} \mu(h), \max_{h \in V'} \mu(h)]$. Combining these two observations, we have

$$\left|\hat{\mu} - \mu(h^*)\right| \leq \frac{1}{2}\left(\max_{h \in V'} \mu(h) - \min_{h \in V'} \mu(h)\right) \leq \epsilon.$$

We now come back to proving this claim by induction on $\mathrm{Cost}(V)$.

Base case. If $\mathrm{Cost}(V) = 0$, then $\mathcal{A}$ immediately exits the while loop without further label queries.

Inductive case. Suppose the claim holds for all V such that $\mathrm{Cost}(V) \leq n$. Now consider a version space V with $\mathrm{Cost}(V) = n+1$. In this case, first recall that

$$\mathrm{Cost}(V) = 1 + \min_{x \in \mathcal{X}} \max_{y \in \{-1,+1\}} \mathrm{Cost}\left(V_x^y\right),$$

i.e. $\min_{x \in \mathcal{X}} \max_{y \in \{-1,+1\}} \mathrm{Cost}\left(V_x^y\right) = \mathrm{Cost}(V) - 1 = n$. Also, recall that by the definition of Algorithm 1, when facing version space V, the next query example x_0 chosen by $\mathcal{A}$ is a solution of the following minimax optimization problem:

$$x_0 = \operatorname*{argmin}_{x \in \mathcal{X}} \max_{y \in \{-1,+1\}} \mathrm{Cost}\left(V_x^y\right),$$

which implies that $\max_{y \in \{-1,+1\}} \mathrm{Cost}\left(V_x^y\right) = n$. Specifically, this implies that the version space at the next iteration, $V\left(h^*, \{x_0\}\right) = V_{x_0}^{h^*(x_0)}$, satisfies that $\mathrm{Cost}(V\left(h^*, \{x_0\}\right)) \leq n$. Combining with the inductive hypothesis, we have seen that after a total of $1 + \mathrm{Cost}(V\left(h^*, \{x_0\}\right)) \leq n+1 = \mathrm{Cost}(V)$ number of label queries, $\mathcal{A}$ will exit the while loop.

This completes the inductive proof of the claim. □

Proof of Theorem 3.2. Fix a deterministic active fairness auditing algorithm $\mathcal{A}$. We will show the following claim: If $\mathcal{A}$ has already obtained an ordered sequence of labeled examples L, and has a remaining label budget $N \leq \mathrm{Cost}(\mathcal{H}[L]) - 1$, then there exists $h \in \mathcal{H}[L]$, such that, $\mathcal{A}$, when interacting with h as the target classifier:

1. obtains a sequence of labeled examples L in the first $|L|$ rounds;
2. has final version space $\mathcal{H}(h, S_{\mathcal{A},h})$ with μ-diameter $> 2\epsilon$.

The theorem follow from this claim by taking $L = \emptyset$. To see why, we let $h \in \mathcal{H}[\emptyset] = \mathcal{H}$ be the classifier described in the claim. First, note that there exists some other classifier $h' \neq h$ in the final version space $\mathcal{H}(h, S_{\mathcal{A},h})$, such that $\left|\mu(h') - \mu(h)\right| > 2\epsilon$. For such h', $h'(S_{\mathcal{A},h}) = h(S_{\mathcal{A},h})$. Therefore, by Lemma B.1, $S_{\mathcal{A},h} = S_{\mathcal{A},h'}$ (which we denote by S subsequently), and h and h' have the exact same labeling on S, and $\hat{\mu}_{\mathcal{A},h} = \hat{\mu}_{\mathcal{A},h'}$. This implies that, for $\mathcal{A}$, at least one of the following must be true:

$$\left|\hat{\mu}_{\mathcal{A},h} - \mu(h)\right| > \epsilon \text{ or } \left|\hat{\mu}_{\mathcal{A},h'} - \mu(h')\right| > \epsilon,$$

showing that it does not guarantee an estimation error $\leq \epsilon$ under all target $h \in \mathcal{H}$.

We now turn to proving the above claim by induction on $\mathcal{A}$'s remaining label budget N. In the following, denote by $V = \mathcal{H}[L]$.

Base case. If $N = 0$ and $\mathrm{Cost}(V) \geq 1$, then $\mathcal{A}$ at this point has zero label budget, which means that it is not allowed to make more queries. In this case, $S_{\mathcal{A},h} = L$, and $\mathcal{H}(S_{\mathcal{A},h}, h) = V$. As $\mathrm{Cost}(V) \geq 1$, we know that

$$\max_{h_1, h_2 \in \mathcal{H}(h, S_{\mathcal{A},h})} \left|\mu(h_1) - \mu(h_2)\right| = \max_{h_1, h_2 \in V} \left|\mu(h_1) - \mu(h_2)\right| > 2\epsilon.$$

This completes the proof of the base case.

Inductive case. Suppose the claim holds for all $N \leq n$. Now, suppose in the learning process, $\mathcal{A}$ has a remaining label budget $N = n+1$, and has obtained labeled examples L such that $V = \mathcal{H}[L]$ satisfies $\mathrm{Cost}(V) \geq n+2$. Let x be the next example $\mathcal{A}$ queries. By the definition of Cost, there exists some $y \in \{-1, +1\}$, such that

$$\mathrm{Cost}\left(\mathcal{H}\left[L \cup \left\{(x,y)\right\}\right]\right) = \mathrm{Cost}\left(V_x^y\right) \geq \mathrm{Cost}(V) - 1 \geq n+1,$$

and after making this query, the learner has a remaining label budget of $N - 1 = n$.

By inductive hypothesis, there exists some $h \in \mathcal{H}\left[L \cup \left\{(x,y)\right\}\right]$, such that when $\mathcal{A}$ interacts with h subsequently (with obtained labeled examples $L \cup \left\{(x,y)\right\}$ and label budget $< n$), the final unlabeled dataset $S_{\mathcal{A},h}$ satisfies

$$\mathrm{diam}_\mu\left(\mathcal{H}(h, S_{\mathcal{A},h})\right) = \max_{h_1, h_2 \in \mathcal{H}(h, S_{\mathcal{A},h})} \left|\mu(h_1) - \mu(h_2)\right| > 2\epsilon.$$

In addition, when interacting with h, $\mathcal{A}$ obtains the example sequence $\langle L, (x,y)\rangle$ in its first $|L|+1$ rounds of interaction, which implies that it obtains the example sequence L in its first $|L|$ rounds of interaction with h. This completes the induction. □

D.2. Proof Sketch of Proposition 3.3

Proof sketch. Let S_1 and S_2 be $O\left(\frac{1}{\epsilon^2}\ln|\mathcal{H}|\right)$ i.i.d samples from $D_X \mid x_A = 1$ and $D_X \mid x_A = 0$, respectively. Define

$$\hat{\mu}(h, S_1, S_2) = \Pr_{x\sim S_1}(h(x) = +1) - \Pr_{x\sim S_2}(h(x) = +1).$$

Hoeffding's inequality and union bound guarantees that with probability at least $\frac{1}{2}$, $\forall h \in \mathcal{H}$, $|\hat{\mu}(h, S_1, S_2) - \mu(h)| \le \epsilon$. Now consider the following deterministic algorithm $\mathcal{A}$:

- Let $n = O\left(\frac{1}{\epsilon^2}\ln|\mathcal{H}|\right)$;
- Find (the lexicographically smallest) S_1 and S_2 in $\mathcal{X}^n$, such that

$$\forall h \in \mathcal{H},\ \left|\hat{\mu}(h, S_1, S_2) - \mu(h)\right| \le \epsilon. \tag{3}$$

  This optimization problem is feasible, because as we have seen, a random choice of S_1, S_2 makes Equation (3) happen with nonzero probability.
- Return $\hat{\mu}(h^*, S_1, S_2)$ with $2n$ label queries to examples in $S_1 \cup S_2$.

By its construction, $\mathcal{A}$ queries $2n = O\left(\frac{1}{\epsilon^2}\ln|\mathcal{H}|\right)$ labels and returns $\hat{\mu}$ that is ϵ-close to $\mu(h^*)$. □

D.3. Proof of Proposition 3.4

Before we prove Proposition 3.4, we first recall the well-known Bernstein's inequality:

Lemma D.1 (Bernstein's inequality)**.** *Given a set of iid random variables $Z_1, \ldots, Z_n$ with mean μ and variance σ^2; in addition, $|Z_i| \le b$ almost surely. Then, with probability $1-\delta$,*

$$\left|\frac{1}{n}\sum_{i=1}^{n} Z_i - \mu\right| \le \sqrt{\frac{2\sigma^2 \ln\frac{2}{\delta}}{n}} + \frac{b\ln\frac{2}{\delta}}{3n}.$$

Proof of Proposition 3.4. We will analyze Algorithm 2, a derandomized version of the Phased CAL algorithm (Hsu, 2010, Chapter 2). To prove this proposition, using Theorem 3.2, it suffices to show that Algorithm 2 has a deterministic label complexity bound of $O\left(\theta(\epsilon)\cdot\ln|\mathcal{H}|\cdot\ln\frac{1}{\epsilon}\right)$.

We first show that for every n, the optimization problem in line 5 is always feasible. To see this, observe that if we draw $S_n = \{x_1, \ldots, x_{m_n}\}$ as sample of size m_n drawn iid from D_X, we have:

1. By Bernstein's inequality with $Z_i = I(x_i \in \mathrm{DIS}(V_n))$, with probability $1 - \frac{1}{4}$,

$$\begin{aligned}\Pr_{S_n}(x \in \mathrm{DIS}(V_n)) \le& \Pr_{D_X}(x \in \mathrm{DIS}(V_n)) + \sqrt{\frac{2\Pr_{D_X}(x \in \mathrm{DIS}(V_n))\ln 8}{m_n}} + \frac{\ln 8}{3m_n}\\ \le& 2\Pr_{D_X}(x \in \mathrm{DIS}(V_n)) + \frac{\ln 8}{m_n}.\end{aligned}$$

   where the second inequality uses Arithmetic Mean-Geometric Mean (AM-GM) inequality.

2. By Bernstein's inequality and union bound over $h, h' \in \mathcal{H}$, we have with probability $1 - \frac{1}{4}$,

$$\forall h, h' \in \mathcal{H}:\quad \Pr_{D_X}(h(x) \ne h'(x)) \le \Pr_{S_n}(h(x) \ne h'(x)) + \sqrt{\frac{4\Pr_{D_X}(h(x) \ne h'(x))\ln|\mathcal{H}|}{m_n}} + \frac{4\ln|\mathcal{H}|}{3m_n}$$

   in which,

$$\forall h, h' \in \mathcal{H}:\quad \Pr_{S_n}(h(x) \ne h'(x)) = 0 \implies \Pr_{D_X}(h(x) \ne h'(x)) \le \frac{16\ln|\mathcal{H}|}{m_n}.$$

By union bound, with nonzero probability, the above two condition hold simultaneously, showing the feasibility of the optimization problem.

We then argue that for all n, $V_{n+1} \subseteq B(h^*, \frac{16\ln|\mathcal{H}|}{m_n})$. This is because for all $h \in V_{n+1}$, it and h^* are both in V_n and therefore they agree on $S_n \setminus T_n$; on the other hand, h and h^* agree on T_n by the definition of of V_{n+1}. As a consequence, $\Pr_{S_n}(h(x) \neq h^*(x)) = 0$, which implies that $\Pr_{D_X}(h(x) \neq h^*(x)) \leq \frac{16\ln|\mathcal{H}|}{m_n}$. As a consequence, for all $h \in V_{N+1}$, $\Pr(h(x) \neq h^*(x)) \leq \frac{16\ln|\mathcal{H}|}{m_N} \leq \underline{p}\epsilon$, implying that $\left|\mu(h) - \mu(h^*)\right| \leq \epsilon$ (recall Lemma C.1).

We now turn to upper bounding Algorithm 2's label complexity:

$$\begin{aligned}\sum_{n=1}^{N}|T_n| =& \sum_{n=1}^{N} m_n \cdot (2\Pr_{D_X}(x \in \mathrm{DIS}(V_n)) + \frac{\ln 8}{m_n}) \\ \leq& \sum_{n=1}^{N} m_n \cdot (\theta(\epsilon)\cdot\frac{16\ln|\mathcal{H}|}{m_n}\cdot\frac{2}{\underline{p}} + \frac{\ln 8}{m_n}) \\ \leq& O\left(\theta(\epsilon)\cdot\ln|\mathcal{H}|\cdot\ln\frac{1}{\epsilon}\right),\end{aligned}$$

where the inequality uses the observation that for every $n \in [N]$,

$$\Pr_{D_X}(x \in \mathrm{DIS}(V_n)) \leq \Pr_{D_X}\left(x \in \mathrm{DIS}(B(h^*, \frac{16\ln|\mathcal{H}|}{m_n}))\right) \leq \theta(\frac{\underline{p}\epsilon}{2})\cdot\frac{16\ln|\mathcal{H}|}{m_n} \leq \theta(\epsilon)\cdot\frac{16\ln|\mathcal{H}|}{m_n}\cdot\frac{2}{\underline{p}},$$

where the second inequality is from the definition of disagreement coefficient (recall Section 1.1), and the last inequality is from a basic property of disagreement coefficient (Hanneke, 2014, Corollary 7.2). □

D.4. Proof of Proposition 3.5

We first prove the following theorem that gives a decision tree-based characterization of the $\mathrm{Cost}(\cdot)$ function. Connections between active learning and optimal decision trees have been observed in prior works (e.g. Laber & Nogueira, 2004; Balcan et al., 2012).

Definition D.2. An example-based decision tree $\mathcal{T}$ for (instance domain, hypothesis set) pair $(\mathcal{X}, V)$ is such that:

1. $\mathcal{T}$'s internal nodes are examples in $\mathcal{X}$; every internal node has two branches, with the left branch labeled as $+1$ and the right labeled as -1.
2. Every leaf l of $\mathcal{T}$ corresponds to a set of classifiers $V_l \subset V$, such that all $h \in V_l$ agree with the examples that appear in the root-to-leaf path to l. Formally, suppose the path from the root to leaf l is an alternating sequence of examples and labels $\langle x_1, y_1, \ldots, x_n, y_n\rangle$, then for every $i \in [n]$, $h(x_i) = y_i$.

Definition D.3. Fix D_X. An example-based decision tree $\mathcal{T}$ is said to (μ,ϵ)-separate a hypothesis set V, if for every leaf l of $\mathcal{T}$, V_l satisfies $\mathrm{diam}_\mu(V_l) \leq 2\epsilon$.

Theorem D.4. *Given a version space V, $\mathrm{Cost}(V)$ is the minimum depth of all decision trees that (μ,ϵ)-separates V.*

Proof. We prove the theorem by induction on $\mathrm{Cost}(V)$.

Base case. If $\mathrm{Cost}(V) = 0$, then $\mathrm{diam}_\mu(V) \leq 2\epsilon$. Then there exists a trivial decision tree (with leaf only) of depth 0 that (μ,ϵ)-separates V, which is also the smallest depth possible.

Inductive case. Suppose the statement holds for any V such that $\mathrm{Cost}(V) = n$. Now consider V such that $\mathrm{Cost}(V) = n+1$.

1. We first show that there exists a decision tree of depth $n+1$ that (μ,ϵ)-separates V. Indeed, pick $x = \mathrm{argmin}_{x\in\mathcal{X}}\max_y \mathrm{Cost}(V_x^y)$.

With this choice of x, we have both $\mathrm{Cost}(V_x^{-1})$ and $\mathrm{Cost}(V_x^{+1})$ are equal to n. Therefore, by inductive hypothesis for V_x^{-1} and V_x^{+1}, we can construct decision trees $\mathcal{T}^-$ and $\mathcal{T}^+$ of depths n that (μ, ϵ)-separate the two hypothesis classes respectively. Now define $\mathcal{T}$ to be such that it has root node x, and has left subtree $\mathcal{T}^+$ and right subtree $\mathcal{T}^-$, we see that $\mathcal{T}$ has depth $n+1$ and (μ, ϵ)-separates V.

2. We next show that any decision tree of depth n does not (μ, ϵ)-separate V. Indeed, assume for the sake of contradiction that such tree $\mathcal{T}$ exists. Then consider the example x at the root of the tree; by the definition of Cost, one of $\mathrm{Cost}(V_x^{-1})$ and $\mathrm{Cost}(V_x^{+1})$ must be $\geq n$. Without loss of generality, assume that $V' = V_x^{+1}$ is such that $\mathrm{Cost}(V') \geq n$. Therefore, there must exists some subset $V'' \subset V'$ such that $\mathrm{Cost}(V'') = n$. Applying the inductive hypothesis on V'', no decision tree of depth $n-1$ can (μ, ϵ)-separate V''. This contradicts with the observation that the left subtree of $\mathcal{T}$, which is of depth $n-1$, (μ, ϵ)-separates V'. □

We now restate a more precise version of Proposition 3.5. First we define the computational task of computing a $0.3\ln(|\mathcal{H}|)$-approximation of $\mathrm{Cost}(\mathcal{H})$ by the following problem:

Problem Minimax-Cost (MC):
Input: instance space $\mathcal{X}$, hypothesis class $\mathcal{H}$, data distribution D_X, precision parameter ϵ.
Output: a number L such that $\mathrm{Cost}(\mathcal{H}) \leq L \leq 0.3\ln(|\mathcal{H}|)\mathrm{Cost}(\mathcal{H})$.

Proposition D.5 (Proposition 3.5 restated)**.** *If there is an algorithm that solves Minimax-Cost in* $\mathrm{poly}(|\mathcal{H}|, |\mathcal{X}|, 1/\epsilon)$ *time, then* $\mathrm{NP} \subseteq \mathrm{TIME}(n^{O(\log\log n)})$.

Proof of Proposition D.5. Our proof takes after (Laber & Nogueira, 2004)'s reduction from set cover (SC) to Decision Tree Problem (DTP). Here, we reduce from SC to the Minimax-Cost problem (MC), i.e. computing $\mathrm{Cost}(\mathcal{H})$ for a given hypothesis class $\mathcal{H}$, taking into account the unique structure of active fairness auditing. Specifically, the following gap version of SC's decision problem has been shown to be computationally hard[4]:

Problem Gap-Set-Cover (Gap-SC):
Input: a universe $U = \{u_1, ..., u_n\}$ of size n with $n \geq 10$, and a family of subsets $\mathcal{C} = \{C_1, ..., C_m\}$, and an integer k, such that either of the following happens:

- Case 1: $\mathrm{OPT}_{\mathsf{SC}} \leq k$,
- Case 2: $\mathrm{OPT}_{\mathsf{SC}} \geq 0.99k\ln n$,

where $\mathrm{OPT}_{\mathsf{SC}}$ denotes the minimum set cover size of $(U, \mathcal{C})$.

Output: 1 or 2, which case the instance is in.

Specifically, it is well-known that obtaining a polynomial time algorithm for the above decision problem[5] on minimum set cover would imply that $\mathrm{NP} \subseteq \mathrm{TIME}(n^{O(\log\log n)})$ (Feige, 1998), which is believed to be false.

To start, recall that an instance of Gap-SC problem $I_{\mathsf{SC}} = (U, \mathcal{C}, k)$; an instance of the MC problem $I_{\mathsf{MC}} = (\mathcal{H}, \mathcal{X}, D_X, \epsilon)$.

With this, we define a coarse reduction β that constructs a MC-instance from a Gap-SC instance with universe $U = \{u_1, ..., u_n\}$ and sets $\mathcal{C} = \{C_1, ..., C_m\}$, which will be refined shortly:

1. Let $\mathcal{H} = \{h_0, h_1, \ldots, h_n\}$, where $h_0(x) \equiv -1$ always, and for all $j \in [n]$, h_j corresponds to u_j (the definitions of h_j's will be given shortly).
2. Create example x_0 such that for all $h \in \mathcal{H}$, $h(x_0) = -1$.
3. For every $i \in [m]$, create basis example x_i to correspond to C_i such that for every $j \in [n]$, $h_j(x_i) = 1$ iff $u_j \in C_i$.

[4]The definition of Gap-SC requires that $n \geq 10$, which is without loss of generality: all Gap-SC instances with $n < 10$ are solvable in constant time.

[5]The constant 0.99 can be changed to any constant < 1 (Feige, 1998).

4. For each set C_i, create $|C_i| - 1$ auxiliary x's as follows: Given set C_i with $|C_i| = s_i$ that corresponds to $\{h_{i1}, .., h_{is_i}\}$, create a balanced binary tree $\mathcal{T}_i$ with each leaf corresponding to a h_{ij}. Create an auxiliary example associated with each internal node in $\mathcal{T}_i$ as follows: for each internal node in the tree, define the corresponding auxiliary sample x such that its label is $+1$ under all the classifiers in the leaves of the subtree rooted at its left child, and its label is -1 under all remaining classifiers in $\mathcal{H}$. The total number of auxiliary x's is $\leq m \cdot (n-1)$.

5. Define $\mathcal{X}$ as the union of the example sets constructed in the above three items, which has at most $N \leq mn + 1$ examples. Define D_X to be such that: $x \mid x_A = 0 \sim \text{Uniform}(\mathcal{X} \setminus \{x_0\})$, and $x \mid x_A = 1 \sim \text{Uniform}(\{x_0\})$, and set $\epsilon = 1/(2N)$. With this setting of ϵ, for every $h \in \mathcal{H}$ such that $h \neq h_0$, $\left|\mu(h) - \mu(h_0)\right| = \left|\Pr(h(x) = +1 \mid x_A = 0) - \Pr(h_0(x) = +1 \mid x_A = 0)\right| \geq \frac{1}{N-1} > 2\epsilon$.

Recall that OPT_{SC} is defined as the size of an optimal solution for SC instance $(U, \mathcal{C})$; we let OPT_{MC} denote the height of the tree corresponding to the optimal query strategy for the MC instance I_{MC} obtained through reduction β. We have the following result:

Lemma D.6. $\text{OPT}_{\mathsf{SC}} \leq \text{OPT}_{\mathsf{MC}} \leq \text{OPT}_{\mathsf{SC}} + \max_{C \in \mathcal{C}} \log |C|$.

Proof. Let $k = \text{OPT}_{\mathsf{SC}}$. We show the two inequalities respectively.

1. By Theorem D.4, it suffices to show that any example-based decision tree $\mathcal{T}$ that (μ, ϵ)-separates $\mathcal{H}$ must have depth at least k. To see this, first note that by item 5 in the reduction β and the definition of (μ, ϵ)-separation, the leaf in $\mathcal{T}$ that contains h_0 must not contain other hypotheses in $\mathcal{H}$. In addition, as $h_0 \equiv -1$, h_0 must lie in the rightmost leaf of $\mathcal{T}$.

   Now to prove the statement, we know that the examples along the rightmost path of $\mathcal{T}$ corresponds to a collection of sets that form a set cover of $\mathcal{C}$. It suffices to show that this set cover has size no greater than the set cover of I_{SC}. This is because the examples along the rightmost path are either x_i's, which correspond to some set in $\mathcal{C}$, or auxiliary examples which correspond to some subset of a set in $\mathcal{C}$. A set cover instance with U and $\mathcal{C}'$ where $\mathcal{C}'$ comprises of sets from $\mathcal{C}$ and subsets of sets from $\mathcal{C}$ will not have a smaller set cover.

   Therefore, the length of the path from the root to the rightmost leaf is at least k, the size of the smallest set cover of the original SC instance I_{SC}.

2. Let an optimal solution for I_{SC} be $G = \{i_1, ..., i_k\}$. Below, we construct an example-based decision tree $\mathcal{T}$ of depth $k + \max_{C \in \mathcal{C}} \log |C|$ that (μ, ϵ)-separates $\mathcal{H}$:

   Let the rightmost path of $\mathcal{T}$ contain nodes corresponding to $x_{i_1}, ..., x_{i_k}$ (the order of these are not important). At level $l = 1, ..., k$, the left subtree of x_{i_l} is defined to be $\mathcal{T}_{i_l}$ as defined in step 4 of reduction β. Note that this may result in $\mathcal{T}$ with potentially empty leaves, in that for some h covered by multiple x_{i_l}'s, it only appears in x_{i_o} where $o = \min\left\{l : h(x_{i_l}) = +1\right\}$.

   We will prove that by the above construction, $\mathcal{T}$ (μ, ϵ)-separates $\mathcal{H}$, as every leaf corresponds to a version space V that is a singleton set (and thus has $\text{diam}_\mu(V) = 0 \leq 2\epsilon$):

   (a) For all but the rightmost leaf, this holds by the construction of $\mathcal{T}_{i_l}$'s.

   (b) For the rightmost leaf, we will show that only h_0 is in the version space. Since G is a set cover, we have that $\cup_{l=1}^{k} C_{i_l} = U$. Therefore, $\forall j \in [n], \exists l \in [k]$ such that $u_j \in C_{i_l} \Leftrightarrow h_j(x_{i_l}) = 1$ by construction. This implies that the all zero labeling of $x_{i_1}, ..., x_{i_k}$ can only correspond to h_0. Therefore, the version space at the rightmost leaf V satisfies $|V| = \{h_0\}$.

   Recall from Theorem D.4 that the depth of $\mathcal{T}$ upper bounds OPT_{MC}. $\mathcal{T}$'s maximum root to leaf path is of length at most $k + \max_{C \in \mathcal{C}} \log |C|$. □

Built from β, we now construct an improved gap preserving reduction β', defined as follows. Given any Gap-SC instance $I_{\mathsf{SC}} = (U, \mathcal{C}, k)$ with universe $U = \{u_1, ..., u_n\}$ and sets $\mathcal{C} = \{C_1, ..., C_m\}$:

1. Take constant $z = \log n$. Construct a Gap-SC instance $I_{\mathsf{SC},z} = (U^z, \mathcal{C}^z, kz)$, containing z copies of the original set covering instance: $U^z = \{u_1^1, \ldots, u_n^1, \ldots, u_1^z, \ldots, u_n^z\}$, $\mathcal{C}^z = \{C_1, \ldots, C_{zm}\}$, where $C_{(p-1)m+i} = \{u_{i1}^p, \ldots, u_{is_i}^p\}$ for $p \in [z]$, $i \in [m]$. Note that $\text{OPT}_{\mathsf{SC},z} = k\text{OPT}_{\mathsf{SC}}$.

2. Apply reduction β to obtain $I_{\mathsf{MC},z}$ from $I_{\mathsf{SC},z}$.

Now, we will argue that β' is a gap-preserving reduction:

1. Suppose the original Gap-SC instance $I_{\mathsf{SC}} = (U, \mathcal{C}, k)$ is in case 1, i.e., $\mathrm{OPT}_{\mathsf{SC}} \leq k$. Then, $\mathrm{OPT}_{\mathsf{SC},z} \leq kz$. By Lemma D.6, $\mathrm{OPT}_{\mathsf{MC},z} \leq kz + \max_{C \in \mathcal{C}^z} \log |C| \leq kz + \log n \leq z(k+1) \leq 2zk$.
2. Suppose the original Gap-SC instance $I_{\mathsf{SC}} = (U, \mathcal{C}, k)$ is in case 2, i.e., $\mathrm{OPT} \geq 0.99k \ln n$. Then, $\mathrm{OPT}_{\mathsf{SC},z} \geq 0.99zk \ln n$, which by Lemma D.6, yields that $\mathrm{OPT}_{\mathsf{MC},z} \geq 0.99zk \ln n$.

Now suppose that there exists an algorithm $\mathcal{A}$ that solves the MC problem in $\mathrm{poly}(|\mathcal{H}|, |\mathcal{X}|, \frac{1}{\epsilon})$ time. We propose the following algorithm $\mathcal{A}'$ that solves the Gap-SC problem in polynomial time, which, as mentioned above, implies that $\mathrm{NP} \subseteq \mathrm{TIME}(n^{O(\log \log n)})$:

Input: $I_{\mathsf{SC}} = (U, \mathcal{C}, k)$.

- Apply β' on I_{SC} to obtain an instance of MC, $I_{\mathsf{MC},z}$
- Let $L \leftarrow \mathcal{A}(I_{\mathsf{MC},z})$. Output 1 if $L \leq 0.7zk \ln n$, and 2 otherwise.

Correctness. As seen above, if I_{SC} is in case 1, then $\mathrm{OPT}_{\mathsf{MC},z} \leq 2zk$. For $n \geq 10$, by the guarantee of $\mathcal{A}$, $L \leq 0.3 \ln |\mathcal{H}| \cdot \mathrm{OPT}_{\mathsf{MC},z} \leq 0.6 \ln(n \log n) \cdot zk \leq 0.7zk \ln n$, and $\mathcal{A}'$ outputs 1. Otherwise, I_{SC} is in case 2, then $\mathrm{OPT}_{\mathsf{MC},z} \geq 0.99zk \ln n$, and by the guarantee of $\mathcal{A}$, $L \geq 0.99zk \ln n > 0.7zk \ln n$, and $\mathcal{A}'$ outputs 2.

Time complexity. In $I_{\mathsf{MC},z}$, $|\mathcal{X}| \leq (mz \cdot nz + 1) = O(mn \log^2 n)$, $|\mathcal{H}| = nz = n \log n$, and $\epsilon = \frac{1}{2N} = \frac{1}{2(mz \cdot nz + 1)} = \Omega(\frac{1}{mn \log^2 n})$. As $\mathcal{A}$ runs in time $O(\mathrm{poly}(|\mathcal{X}|, |\mathcal{H}|, \frac{1}{\epsilon}))$, $\mathcal{A}'$ runs in time $O(\mathrm{poly}(m, n))$. □

D.5. Deferred Materials for Section 3.2

D.5.1. (μ, ϵ)-SPECIFYING SET, (μ, ϵ)-TEACHING DIMENSION AND THEIR PROPERTIES

The following definitions are inspired by the teaching and exact active learning literature (Hegedűs, 1995; Hanneke, 2007).

Definition D.7 ((μ, ϵ)-specifying set)**.** Fix hypothesis class $\mathcal{H}$ and any function $h : \mathcal{X} \to \mathcal{Y}$,[6] a set of unlabeled examples S is said to be a (μ, ϵ)-*specifying set* for h and $\mathcal{H}$, if $\forall h_1, h_2 \in \mathcal{H}(h, S) \,.\, |\mu(h_1) - \mu(h_2)| \leq 2\epsilon$.

Definition D.8 ((μ, ϵ)-extended teaching dimension)**.** Fix hypothesis class $\mathcal{H}$ and any function $h : \mathcal{X} \to \mathcal{Y}$, define $t(h, \mathcal{H}, \mu, \epsilon)$ as the size of the minimum (μ, ϵ)-specifying set for h and $\mathcal{H}$, i.e. it is the optimal solution of the following optimization problem (OP-h):

$$\min |S|, s.t. \forall h_1, h_2 \in \mathcal{H}(h, S) \,.\, |\mu(h_1) - \mu(h_2)| \leq 2\epsilon$$

Definition D.9. We define the μ-extended teaching dimension $\mathrm{XTD}(\mathcal{H}, \mu, \epsilon) := \max_{h: \mathcal{X} \to \mathcal{Y}} t(h, \mathcal{H}, \mu, \epsilon)$.

The improper teaching dimension is related to $\mathrm{Cost}(\mathcal{H})$ in that:

Lemma D.10.

$$\mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \leq \mathrm{Cost}(\mathcal{H}).$$

Proof. Let $h_0 = \mathrm{argmax}_{h: \mathcal{X} \to \mathcal{Y}} t(h, \mathcal{H}, \mu, \epsilon)$. Let k denote $t(h_0, \mathcal{H}, \mu, \epsilon) - 1$. It suffices to show that $\mathrm{Cost}(\mathcal{H}) \geq k$. To see this, first note that

$$\begin{aligned}
\mathrm{Cost}(\mathcal{H}) &= 1 + \min_x \max_y \mathrm{Cost}(\mathcal{H}[(x, y)]) \\
&\geq 1 + \min_{x_1 \in \mathcal{X}} \mathrm{Cost}(\mathcal{H}[(x, h_0(x))]) \\
&\geq 2 + \min_{x_1 \in \mathcal{X}} \min_{x_2 \in \mathcal{X}} \mathrm{Cost}(\mathcal{H}[\{(x_1, h_0(x_1)), (x_2, h_0(x_2))\}])
\end{aligned}$$

[6] Note that h is allowed to be outside $\mathcal{H}$.

We can repeatedly unroll the above expression as long as $\operatorname{diam}_\mu(\mathcal{H}[\{(x_1, h_0(x_1)), \ldots, (x_i, h_0(x_i))\}])$ is at least $> 2\epsilon$. After unrolling $k-1$ times where $U_{k-1} = \langle x_1, \ldots, x_{k-1}\rangle$, we have

$$\operatorname{Cost}(\mathcal{H}) \geq k - 1 + \min_{U_{k-1}} \operatorname{Cost}(\mathcal{H}(h_0, U_{k-1})).$$

By the definition of $t(h, \mathcal{H}, \mu, \epsilon)$, for any U with $U \leq k-1$, there exists $h', h'' \in \mathcal{H}(h_0, U)$ such that $|\mu(h') - \mu(h'')| > \epsilon \Rightarrow \operatorname{diam}_\mu(\mathcal{H}(h_0, U)) > \epsilon$. Thus, for any unlabeled dataset U_{k-1} of size $k-1$, $\operatorname{Cost}(\mathcal{H}(h_0, U_{k-1})) \geq 1$. Therefore, $\operatorname{Cost}(\mathcal{H}) \geq k$. □

D.5.2. Proof of Theorem 3.8

Proof. We prove the theorem as follows:

Correctness. Observe that right before Algorithm 3 returns, it must execute lines 9 and 17. Since the condition on line 17 is also satisfied, the dataset T must be such that $\hat{h}(T) = h^*(T)$. Combined with the definitions of optimization problems (1) and (2), this implies that, the h_1 and h_2 used in line 9 right before return satisfy that

$$\mu(h_1) = \min_{h \in \mathcal{H}(h^*, T)} \mu(h), \quad \mu(h_2) = \max_{h \in \mathcal{H}(h^*, T)} \mu(h).$$

Therefore, $\mu(h^*) \in [\min_{h \in \mathcal{H}(h^*, T)} \mu(h), \max_{h \in \mathcal{H}(h^*, T)} \mu(h)] = [\mu(h_1), \mu(h_2)]$. Furthermore, by line 9, $\mu(h_1) - \mu(h_2) \leq 2\epsilon$. Hence, $\hat{\mu}$, the output of Algorithm 3, satisfies that,

$$\left|\hat{\mu} - \mu(h^*)\right| = \left|\frac{1}{2}\left(\mu(h_1) + \mu(h_2)\right) - \mu(h^*)\right| \leq \epsilon.$$

Label complexity. We now bound the label complexity of the algorithm, specifically, in terms of $\operatorname{XTD}(\mathcal{H}, \mu, \epsilon)$.

First, at the end of the t-th iteration of the outer loop, the newly collected dataset T_t must be such that $\exists x \in T_t$ and $\hat{h}(x) \neq h^*(x)$. As $\mathcal{O}$ has a mistake bound of M, the total number of outer loop iterations, denoted by N, must be most M. In addition, by Lemma D.11 given below, with probability $1 - \delta/M$, $|T_t| \leq O\left(\operatorname{XTD}(\mathcal{H}, \mu, \epsilon) \cdot \log \frac{|\mathcal{H}|M}{\delta} \log |\mathcal{X}|\right)$. Therefore, by a union bound, with probability $1 - \delta$, the total number of label queries made by Algorithm 3 is at most

$$\sum_{t=1}^{N} |T_t| \leq O\left(M \cdot \operatorname{XTD}(\mathcal{H}, \mu, \epsilon) \cdot \log \frac{|\mathcal{H}|M}{\delta} \log |\mathcal{X}|\right). \qquad \square$$

Lemma D.11. *For every outer iteration of Algorithm 3, with probability $\geq 1 - \frac{\delta}{M}$, T, the dataset at the end of this iteration, satisfies $|T| \leq O\left(\operatorname{XTD}(\mathcal{H}, \mu, \epsilon) \cdot \log \frac{|\mathcal{H}|M}{\delta} \log |\mathcal{X}|\right)$.*

Proof. The inner loop is similar to the "black-box teaching" algorithm of (Dasgupta et al., 2019) except that we are teaching $\mu(\hat{h})$ as opposed to $\hat{h}$ itself. Although (Dasgupta et al., 2019)'s algorithm was originally designed for exact (interactive) teaching, it implicitly gives an oracle-efficient algorithm for approximately computing the minimum set cover; we will use this insight throughout the proof. As the analysis of (Dasgupta et al., 2019) is only on the *expected* number of teaching examples, we use a different filtration to obtain a high probability bound over the number of teaching examples.

First we setup some useful notations for the proof. let $\mathcal{X} = \{x_1, \ldots, x_m\}$. Recall that $\lambda = \ln \frac{|\mathcal{H}|^2 M}{\delta}$. Let $W_i(x)$ denote the weight of point $x \in \mathcal{X}$ (denoted by $w(x)$ in the algorithm) at the end of round i of the inner loop and let τ_{x_j} be the exponentially-distributed threshold associated with x_j. Define random variable $U_{i,j} = \mathbb{1}\{\tau_{x_j} > W_i(x_j)\}$. Let M_i denotes the number of teaching examples selected in the ith round of doubling; it can be seen that $M_i = \sum_{j \in [m]} U_{i,j}$. Also define $(i, j) \preceq (i', j')$ iff (i, j) precedes (i', j') lexicographically.

Define two filtrations:

1. Let $\mathcal{F}_{i,j}$ be the sigma-field of all indicator events $\{U_{i',j'} : (i', j') \preceq (i, j)\}$. As a convention, $\mathcal{F}_{i,0} := \mathcal{F}_{i-1,m}$.

2. Let $\mathcal{F}_i$ be the sigma-field of all indicator events $\{U_{i',j'} : j' \in [m], 1 \le i' \le i\}$; this is the filtration used by (Dasgupta et al., 2019). It can be easily seen that $\mathcal{F}_i = \mathcal{F}_{i,m}$.

Define $Y_{i,j} = \sum_{(i',j') \preceq (i,j)} Z_{i',j'}$, where $Z_{i,j} = U_{i,j} - \mathbb{E}\left[U_{i,j} \mid \mathcal{F}_{i,j-1}\right] \in [-1, +1]$. Then $Y_{i,j}$ is a martingale as $\mathbb{E}[Y_{i,j}|\mathcal{F}_{i,j-1}] = \mathbb{E}[Z_{i,j}|\mathcal{F}_{i,j-1}] + \mathbb{E}[Y_{i,j-1}|\mathcal{F}_{i,j-1}] = Y_{i,j-1}$.

Let N be the total number of rounds, which by item 1 of Lemma D.13, is $O(\mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \ln |\mathcal{X}|)$ (Lemma 4 of (Dasgupta et al., 2019)) with probability 1. We may then apply Freedman's inequality (Lemma D.12): since $Y_{i,j} - Y_{i,j-1} = Z_{i,j} \le 1$ almost surely, for any s and any $\sigma^2 > 0$,

$$\Pr\left(\exists n, m, Y_{nm} \ge s, \sum_{(i,j) \preceq (n,m)} \mathbb{E}[Z_{ij}^2 | \mathcal{F}_{i(j-1)}] \le \sigma^2\right) \le \exp\left(-\frac{s^2}{2(\sigma^2 + s/3)}\right) \tag{4}$$

Next, we let $\sigma^2 = \lambda(1 + \mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \ln(2|\mathcal{X}|))$; we have for any n, m:

$$\begin{aligned}
&\sum_{(i,j) \preceq (n,m)} \mathbb{E}[Z_{ij}^2 | \mathcal{F}_{i(j-1)}] \\
&= \sum_{(i,j) \preceq (n,m)} \mathbb{E}[U_{ij}^2 | \mathcal{F}_{i(j-1)}] - \mathbb{E}[U_{ij} | \mathcal{F}_{i(j-1)}]^2 \\
&\le \sum_{(i,j) \preceq (n,m)} \mathbb{E}[U_{ij}^2 | \mathcal{F}_{i(j-1)}] \\
&= \sum_{(i,j) \preceq (n,m)} \mathbb{E}[U_{ij} | \mathcal{F}_{i(j-1)}] \\
&= \sum_{i=1}^{n} \mathbb{E}_{\mathcal{F}_{i-1}}[M_i] \\
&\le \lambda \sum_{x \in \mathcal{X}} W_n(x) && \text{(Lemma D.14)} \\
&\le \lambda(1 + \mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \ln(2|\mathcal{X}|)) = \sigma^2. && \text{(Lemma D.13)}
\end{aligned}$$

Meanwhile, we choose $s = \frac{1}{6}\log(\frac{1}{\delta}) + \sqrt{2\sigma^2 \log\frac{1}{\delta} + \frac{1}{6}\log(\frac{1}{\delta})} = O\left(\sqrt{\ln\frac{1}{\delta}}\sigma + \ln\frac{1}{\delta}\right)$, which ensures that the right hand side of Eq. (4) is at most δ.

Thus, by Equation (4), we have with probability $1 - \delta$, for all n, m,

$$Y_{nm} = \sum_{(i',j') \preceq (n,m)} U_{i'j'} - \sum_{i=1}^{n} \mathbb{E}_{\mathcal{F}_{i-1}}[M_i] \le O\left(\sqrt{\ln\frac{1}{\delta}}\sigma + \ln\frac{1}{\delta}\right).$$

Also, using Lemma D.14 and D.13, with probability 1, $\sum_{i=1}^{N} \mathbb{E}_{\mathcal{F}_{i-1}}[M_i] \le \lambda(1 + \mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \ln(2|\mathcal{X}|))$.

Therefore, for Y_{Nm} in particular,

$$\begin{aligned}
Y_{Nm} &\le O\left(\lambda(1 + \mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \ln(2|\mathcal{X}|)) + \sqrt{\lambda(1 + \mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \ln(2|\mathcal{X}|)) \ln(1/\delta)} + \ln(1/\delta)\right) \\
&= O\left(\lambda(1 + \mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \ln(2|\mathcal{X}|)) + \ln\frac{1}{\delta}\right) \\
&= O\left(\mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \ln(|\mathcal{X}|) \ln((|\mathcal{H}|M)/\delta)\right).
\end{aligned}$$

□

Lemma D.12 (Freedman's Inequality)**.** *Let martingale $\{Y_k\}_{k=0}^{\infty}$ with difference sequence $\{X_k\}_{k=0}^{\infty}$ be such that $X_k \le R$ a.s for all k and $Y_0 = 0$. Let $W_k = \sum_{j=1}^{k} \mathbb{E}_{j-1}[X_j^2]$. Then, for all $t \ge 0$ and $\sigma^2 > 0$:*

$$\Pr(\exists k \ge 0 : Y_k \ge t \wedge W_k \le \sigma^2) \le \exp\left(-\frac{t^2/2}{\sigma^2 + Rt/3}\right).$$

Lemma D.13. *For any outer iteration of Algorithm 3:*

1. *The number of inner loop iterations is at most* $\mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \cdot \log(2|\mathcal{X}|)$.
2. *At any point in the inner loop, we have that,* $\sum_{x \in \mathcal{X}} w(x) \leq 1 + \mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \cdot \log(2|\mathcal{X}|)$.

Proof. The proof is very similar to Dasgupta et al. (2019, Lemma 4) with some differences; for completeness, we include a proof here.

We first prove the second item. First, note that at any point of the algorithm, for all x, $w(x) \leq 2$. Let $S^*(\hat{h})$ be the optimal solution of optimization problem (OP-$\hat{h}$) - we have $|S^*(\hat{h})| = t(\hat{h}, \mathcal{H}, \mu, \epsilon) \leq \mathrm{XTD}(\mathcal{H}, \mu, \epsilon)$. Note that every time when line 13 is called, by the feasibility of $S^*(\hat{h})$ with respect to (OP-$\hat{h}$), $\Delta(h_1, h_2) \cap S^*(\hat{h}) \neq \emptyset$, therefore, the weight of some element $x \in S^*(\hat{h})$ gets doubled. This implies that the total number of times line 13 is executed is at most $|S^*(\hat{h})| \cdot \log(2|\mathcal{X}|)$. Otherwise, if the number of time line 13 is executed is $\geq |S^*(\hat{h})| \cdot \log(2|\mathcal{X}|) + 1$, by the pigeonhole principle, there must exist some element $x \in S^*(\hat{h})$ whose weight exceeds 1, which is a contradiction.

Finally, note that each weight doubling only increases the total weight by ≤ 1, we have the final total weight is at most

$$1 + 1 \cdot |S^*(\hat{h})| \cdot \log(2|\mathcal{X}|) \leq 1 + \mathrm{XTD}(\mathcal{H}, \mu, \epsilon) \cdot \log(2|\mathcal{X}|).$$

The first item follows since the number of inner iterations is at most the number of weight doublings. □

Lemma D.14. *For every inner iteration,* $\mathbb{E}[M_i|\mathcal{F}_{i-1}] \leq \sum_{x \in \mathcal{X}} \lambda(W_i(x) - W_{i-1}(x))$.

Proof. The proof is almost a verbatim copy of Dasgupta et al. (2019, Lemma 6), which we include here:

$$\begin{aligned}
\mathbb{E}[M_i|\mathcal{F}_{i-1}] &= \sum_{x \in \mathcal{X}} \Pr(x \text{ chosen in round } i | x \text{ not chosen before round } i, \mathcal{F}_{i-1}) \\
&= \sum_{x \in \mathcal{X}} 1 - \Pr(\tau_x > W_i(x) | \tau_x > W_{i-1}(x)) \\
&= \sum_{x \in \mathcal{X}} (1 - \exp(-\lambda(W_i(x) - W_{i-1}(x)))) \\
&\leq \sum_{x \in \mathcal{X}} \lambda(W_i(x) - W_{i-1}(x)).
\end{aligned}$$

□

E. Deferred Materials from Section 4

E.1. Distribution-free Query Complexity Lower Bounds for Auditing with VC classes

Theorem E.1 (Lower bound for randomized auditing)**.** *If hypothesis class $\mathcal{H}$ has VC dimension $d \geq 1600$, and $\epsilon \in (0, \frac{1}{40}]$, then for any (possibly randomized) algorithm $\mathcal{A}$, there exists a distribution D realizable by $h^* \in \mathcal{H}$, such that when $\mathcal{A}$ is given a querying budget $N \leq \Omega(\min(d, \frac{1}{\epsilon^2}))$, its output $\hat{\mu}$ is such that*

$$\mathbb{P}\left(|\hat{\mu} - \mu(h^*)| > \epsilon\right) > \frac{1}{8}.$$

Proof. We will be using Le Cam's method with several subtle modifications. First, we will reduce the estimation problem to a hypothesis testing problem, where under different hypotheses, the $\mu(h^*)$ will be centered around two $\Omega(\epsilon)$-separated values with high probability. Second, we will upper bound the distribution divergence of the interaction history under the two hypotheses; this requires some delicate handling, as the label on a queried example depends not only on the identity of the example, but also historical labeled examples.

Step 1: the construction. As $\mathrm{VC}(\mathcal{H}) = d$, there exists a set of examples $Z = \{z_0, z_1, \ldots, z_{d-1}\} \subset \mathcal{X}$ shattered by $\mathcal{H}$. Let $Z_+ = \{z_1, \ldots, z_{d-1}\}$. Let D_X be as follows: $x \mid x_A = 0$ is uniform over Z_+, whereas $x \mid x_A = 1$ is the delta mass on z_0.

Let $\tilde{\epsilon} = 10\max(\epsilon, \frac{1}{\sqrt{d}})$; by the conditions that $d \geq 1600$ and $\epsilon \leq \frac{1}{40}$, we have $\tilde{\epsilon} \leq \frac{1}{4}$. Let label budget $N = \frac{1}{24\tilde{\epsilon}^2} = \Omega\left(\min(d, \frac{1}{\epsilon^2})\right)$.

Consider two hypotheses that choose h^* randomly from $\{-1, +1\}^{Z_+}$, subject to $h^*(z_0) = 0$:

- H_0: choose h^* such that for every $i \in [d-1]$, independently, $h^*(z_i) = \begin{cases} +1, & \text{with probability } \frac{1}{2} - \tilde{\epsilon} \\ -1, & \text{with probability } \frac{1}{2} + \tilde{\epsilon} \end{cases}$
- H_1: choose h^* such that for every $i \in [d-1]$, independently, $h^*(z_i) = \begin{cases} +1, & \text{with probability } \frac{1}{2} + \tilde{\epsilon} \\ -1, & \text{with probability } \frac{1}{2} - \tilde{\epsilon} \end{cases}$

We have the following simple claim that shows the separation of $\mu(h^*)$ under the two hypotheses. Its proof is deferred to the end of the main proof.

Claim E.2. $\mathbb{P}_{h^* \sim H_0}\left(\mu(h^*) \leq \frac{1}{2} - \frac{1}{2}\tilde{\epsilon}\right) \geq \frac{15}{16}$, and $\mathbb{P}_{h^* \sim H_1}\left(\mu(h^*) \geq \frac{1}{2} + \frac{1}{2}\tilde{\epsilon}\right) \geq \frac{15}{16}$.

Step 2: upper bounding the statistical distance. Next, we show that H_0 and H_1 are hard to distinguish with $\mathcal{A}$ having a label budget of N. To this end, we upper bound the KL divergence of the joint distributions of $\langle (x_1, y_1), \ldots, (x_n, y_n)\rangle =: (x, y)_{\leq n}$ under H_0 and H_1, denoted as $\mathbb{P}_0$ and $\mathbb{P}_1$ respectively. Applying Lemma E.14, we have:

$$\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_1) = \sum_{i=1}^{n} \mathbb{E}\left[\mathrm{KL}\left(\mathbb{P}_0(y_i = \cdot \mid (x, y)_{\leq i-1}, x_i)), \mathbb{P}_1(y_i = \cdot \mid (x, y)_{\leq i-1}, x_i)\right)\right]. \tag{5}$$

We claim that for every i and $((x, y)_{\leq i-1}, x_i) \in (\mathcal{X} \times \mathcal{Y})^{i-1} \times \mathcal{X}$ on the support of $\mathbb{P}_0$,

$$\mathrm{KL}\left(\mathbb{P}_0(y_i = \cdot \mid (x, y)_{\leq i-1}, x_i)), \mathbb{P}_1(y_i = \cdot \mid (x, y)_{\leq i-1}, x_i)\right) \leq 3\tilde{\epsilon}^2. \tag{6}$$

First, observe that if $\langle (x, y)_{\leq i-1}, x_i\rangle$ is in the support of $\mathbb{P}_0$, there must exists some $h^* : Z \to \{-1, +1\}$ such that $h^*(x_j) = y_j$ for all $j \in [i-1]$; in particular, this means there must not exist $j_1 \neq j_2$ in $[i-1]$, such that $x_{j_1} = x_{j_2}$ but $y_{j_1} \neq y_{j_2}$.

Next, we note that, under H_0, conditioned on $(x, y)_{\leq i-1}$, the posterior distribution of h^* is supported over the set $\left\{h \mid h : Z \to \{-1, +1\}, \forall j \in [i-1], h(x_j) = y_j\right\}$, and specifically, for all $x \in Z \setminus \left\{x_j : j \in [i-1]\right\}$, the $h^*(x)$'s are independent conditioned on $(x, y)_{\leq i-1}$, and

$$\mathbb{P}_0\left(h^*(x) = +1 \mid (x, y)_{\leq i-1}\right) = \frac{1}{2} - \tilde{\epsilon}.$$

The same statement holds for H_1 except that for all $x \in Z \setminus \left\{x_j : j \in [i-1]\right\}$, we now have $\mathbb{P}_1(h^*(x) = +1 \mid (x, y)_{\leq i-1}) = \frac{1}{2} + \tilde{\epsilon}$. In addition, the conditional distribution of $y_i \mid (x, y)_{\leq i-1}, x_i$, equals the conditional distribution of $h^*(x_i) \mid (x, y)_{\leq i-1}$, under both H_0 and H_1. We now perform a case analysis:

1. If $x_i \in \left\{x_j : j \in [i-1]\right\}$, then under both H_0 and H_1, the distributions of $h^*(x_i) \mid (x, y)_{\leq i-1}$ are equal: they both equal to the delta mass supported on the only element of the singleton set $\left\{y_j : j \in [i-1], x_j = x_i\right\}$. In this case, $\mathrm{KL}\left(\mathbb{P}_0(y_i = \cdot \mid (x, y)_{\leq i-1}, x_i)), \mathbb{P}_1(y_i = \cdot \mid (x, y)_{\leq i-1}, x_i)\right) = 0 \leq 3\tilde{\epsilon}^2$.
2. Otherwise, $x_i \notin \left\{x_j : j \in [i-1]\right\}$. Under H_0, $h^*(x_i) \mid (x, y)_{\leq i-1}$ takes value $+1$ with probability $\frac{1}{2} - \tilde{\epsilon}$, and takes value -1 with probability $\frac{1}{2} + \tilde{\epsilon}$; similarly, under H_1, $h^*(x_i) \mid (x, y)_{\leq i-1}$ takes value $+1$ with probability $\frac{1}{2} + \tilde{\epsilon}$, and takes value -1 with probability $\frac{1}{2} - \tilde{\epsilon}$. In this case, by Fact E.13 and that $\tilde{\epsilon} \leq \frac{1}{4}$, $\mathrm{KL}\left(\mathbb{P}_0(y_i = \cdot \mid (x, y)_{\leq i-1}, x_i)), \mathbb{P}_1(y_i = \cdot \mid (x, y)_{\leq i-1}, x_i)\right) = \mathrm{kl}\left(\frac{1}{2} - \tilde{\epsilon}, \frac{1}{2} + \tilde{\epsilon}\right) \leq 3\tilde{\epsilon}^2$.

In summary, in both cases, Equation (6) holds, and plugging this back to Equation (5) with $n = \frac{1}{24\tilde{\epsilon}^2}$, we have $\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_1) \leq 3n\tilde{\epsilon}^2 \leq \frac{1}{8}$. By Pinsker's inequality (Lemma E.11), $d_{\mathrm{TV}}(\mathbb{P}_0, \mathbb{P}_1) \leq \sqrt{\frac{1}{2}\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_1)} \leq \frac{1}{2}$. By Le Cam's

Lemma (Lemma E.10), for any hypothesis tester $\hat{b}$, we have

$$\frac{1}{2}\mathbb{P}_0\left(\hat{b}=1\right)+\frac{1}{2}\mathbb{P}_1\left(\hat{b}=0\right)\geq\frac{1}{2}\left(1-d_{\mathrm{TV}}(\mathbb{P}_0,\mathbb{P}_1)\right)\geq\frac{1}{4}. \tag{7}$$

Step 3: concluding the proof. Given $\mathcal{A}$'s output auditing estimate $\hat{\mu}$, consider the following hypothesis test:

$$\hat{b}=\begin{cases}0, & \hat{\mu}<\frac{1}{2},\\ 1, & \hat{\mu}\geq\frac{1}{2}.\end{cases}$$

Plugging into Equation (7), we have

$$\frac{1}{2}\mathbb{P}_0\left(\hat{\mu}\geq\frac{1}{2}\right)+\frac{1}{2}\mathbb{P}_1\left(\hat{\mu}<\frac{1}{2}\right)\geq\frac{1}{4}. \tag{8}$$

Now, recall Claim E.2, and using the fact that $\mathbb{P}(A\cap B)\geq\mathbb{P}(A)-\mathbb{P}(B^C)=\mathbb{P}(A)+\mathbb{P}(B)-1$, we have

$$\mathbb{P}_0\left(\left|\hat{\mu}-\mu(h^*)\right|\geq\frac{1}{2}\tilde{\epsilon}\right)\geq\mathbb{P}_0\left(\hat{\mu}\geq\frac{1}{2},\mu(h^*)\leq\frac{1}{2}-\frac{1}{2}\tilde{\epsilon}\right)\geq\mathbb{P}_0\left(\hat{\mu}\geq\frac{1}{2}\right)+\frac{15}{16}-1\geq\mathbb{P}_0\left(\hat{\mu}\geq\frac{1}{2}\right)-\frac{1}{16}. \tag{9}$$

Symmetrically, we also have

$$\mathbb{P}_1\left(\left|\hat{\mu}-\mu(h^*)\right|\geq\frac{1}{2}\tilde{\epsilon}\right)\geq\mathbb{P}_1\left(\hat{\mu}<\frac{1}{2},\mu(h^*)\geq\frac{1}{2}+\frac{1}{2}\tilde{\epsilon}\right)\geq\mathbb{P}_1\left(\hat{\mu}<\frac{1}{2}\right)-\frac{1}{16}. \tag{10}$$

Combining Equations (8), (9), and (10), we have

$$\frac{1}{2}\mathbb{P}_0\left(\left|\hat{\mu}-\mu(h^*)\right|\geq\frac{1}{2}\tilde{\epsilon}\right)+\frac{1}{2}\mathbb{P}_1\left(\left|\hat{\mu}-\mu(h^*)\right|\geq\frac{1}{2}\tilde{\epsilon}\right)\geq\frac{1}{4}-\frac{1}{16}>\frac{1}{8}.$$

As $\frac{1}{2}\tilde{\epsilon}>\epsilon$, and the left hand side can be viewed as the total probability of $\left|\hat{\mu}-\mu(h^*)\right|>\epsilon$ when h^* is drawn from the uniform mixture distribution of the h^* distributions under H_0 and H_1. By the probabilistic method, there exists some h^* such that $\mathbb{P}_{h^*,\mathcal{A}}\left(\left|\hat{\mu}-\mu(h^*)\right|>\epsilon\right)>\frac{1}{8}$. □

Proof of Claim E.2. Without loss of generality, we show the first inequality; the second inequality can be shown symmetrically. Note that under H_0, the random h^*'s DP value satisfies

$$\mu(h^*)=\Pr(h^*(x)=+1\mid x_A=0)-\Pr(h^*(x)=+1\mid x_A=1)=\frac{1}{d-1}\sum_{i=1}^{d-1}\mathbb{1}\{h^*(z_i)=+1\},$$

where the second equality follows from that $\Pr(h^*(x)=+1\mid x_A=1)=0$ as $h^*(z_0)=-1$ is always true.

Under H_0, $(d-1)\mu(h^*)$ is the sum of $(d-1)$ iid Bernoulli random variables with mean parameter $\frac{1}{2}-\tilde{\epsilon}$. Therefore, by Hoeffding's inequality, we have

$$\mathbb{P}_0\left(\mu(h^*)>\frac{1}{2}-\frac{1}{2}\tilde{\epsilon}\right)\leq\exp\left(-2(d-1)\cdot\left(\frac{1}{2}\tilde{\epsilon}\right)^2\right)\leq\frac{1}{16},$$

where the second inequality uses the fact that $\tilde{\epsilon}=10\max\left(\epsilon,\frac{1}{\sqrt{d}}\right)\geq\frac{10}{\sqrt{d}}$. □

E.2. Query Complexity for Auditing Non-homogeneous Halfspaces under Gaussian Subpopulations

Theorem E.3 (Lower bound)**.** *Let $d\geq 6400$ and $\epsilon\in(0,\frac{1}{80}]$. If D_X is such that $x\mid x_A=0\sim\mathrm{N}(0_d,I_d)$, whereas $x\mid x_A=1\sim\mathrm{N}(0_d,(0)_{d\times d})$ (i.e. the delta-mass supported at 0_d). For any (possibly randomized) algorithm $\mathcal{A}$, there exists h^* in $\mathcal{H}_{lin}$ the class of nonhomogeneous linear classifiers, such that when $\mathcal{A}$ is given a query budget $N\leq\Omega\left(\min(d,\frac{1}{\epsilon^2})\right)$, its output $\hat{\mu}$ is such that*

$$\mathbb{P}_{\mathcal{A},h^*}\left(\left|\hat{\mu}-\mu(h^*)\right|>\epsilon\right)>\frac{1}{8}.$$

Proof. Similar to the proof of Theorem E.1, we will use Le Cam's method. In addition to the same challenges in the proof of Theorem E.1, in the active fairness auditing for halfspaces setting, we are faced with the extra challenge that the posterior distributions of $h^*(x_i) \mid (x,y)_{\le i-1}$ deviates significantly from the prior distribution of $h^*(x_i)$, and cannot be easily calculated in closed form. To get around this difficulty, using the chain rule of KL divergence, along with the posterior formula for noiseless Bayesian linear regression with Gaussian prior, we calculate a tight upper bound on the KL divergence between two carefully constructed, well-separated hypotheses.

Step 1: the construction. Let $\tilde{\epsilon} = 40\max(\epsilon, \frac{1}{\sqrt{d}})$; by the assumption that $\epsilon \le \frac{1}{80}$ and $d \ge 6400$, we have $\tilde{\epsilon} \le \frac{1}{2}$. Let label budget $N = \frac{1}{64\tilde{\epsilon}^2} = \Omega\left(\min(d, \frac{1}{\epsilon^2})\right)$.

Consider two hypotheses that choose $h^* = h_{a^*,b^*}$, such that $b^* = -1$, and a^* is chosen randomly from different distributions:

- $H_0 : a^* \sim \mathrm{N}(0, \frac{1}{d}(1+\tilde{\epsilon})I_d)$
- $H_1 : a^* \sim \mathrm{N}(0, \frac{1}{d}(1-\tilde{\epsilon})I_d)$

We have the following claim that shows the separation of $\mu(h^*)$ under the two hypotheses. Its proof is deferred to the end of the main proof.

Claim E.4. $\mathbb{P}_{h^*\sim H_0}\left(\mu(h^*) > \Phi(-1) + \frac{\tilde{\epsilon}}{36}\right) \ge \frac{15}{16}$, and $\mathbb{P}_{h^*\sim H_1}\left(\mu(h^*) < \Phi(-1) - \frac{\tilde{\epsilon}}{36}\right) \ge \frac{15}{16}$, where $\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}\,\mathrm{d}z$ is the standard normal CDF.

Step 2: upper bounding the statistical distance. Next, we show that H_0 and H_1 are hard to distinguish with $\mathcal{A}$ making $n \le N$ label queries. To this end, we upper bound the KL divergence of the joint distributions of $(x,y)_{\le n}$ under H_0 and H_1, denoted as $\mathbb{P}_0$ and $\mathbb{P}_1$ respectively. To this end, define $\tilde{y}_i = \langle a^*, x_i\rangle - 1$ for $i \in [n]$, and $y_i = \mathrm{sign}(\tilde{y}_i)$. Define $\tilde{\mathbb{P}}_0$ and $\tilde{\mathbb{P}}_1$ (resp. $\mathbb{Q}_0$ and $\mathbb{Q}_1$) as the joint distributions of $(x,\tilde{y})_{\le n}$ (resp. $(x,y,\tilde{y})_{\le n}$) under H_0 and H_1 respectively. By the chain rule of KL divergence (Lemma E.12 with $Z = (x,y)_{\le n}, W = \tilde{y}_{\le n}$ and $Z = (x,\tilde{y})_{\le n}, W = y_{\le n}$ respectively), we get:

$$\begin{aligned}
&\mathrm{KL}(\mathbb{Q}_0((x,y,\tilde{y})_{\le n}), \mathbb{Q}_1((x,y,\tilde{y})_{\le n}) \\
=&\underbrace{\mathrm{KL}(\mathbb{Q}_0((x,y)_{\le n}), \mathbb{Q}_1((x,y)_{\le n})}_{\mathrm{KL}(\mathbb{P}_0,\mathbb{P}_1)} + \underbrace{\mathrm{KL}(\mathbb{Q}_0((\tilde{y})_{\le n} \mid (x,y)_{\le n}), \mathbb{Q}_1((\tilde{y})_{\le n} \mid (x,y)_{\le n}))}_{\ge 0} \\
=&\underbrace{\mathrm{KL}(\mathbb{Q}_0((x,\tilde{y})_{\le n}), \mathbb{Q}_1((x,\tilde{y})_{\le n})}_{\mathrm{KL}(\tilde{\mathbb{P}}_0,\tilde{\mathbb{P}}_1)} + \underbrace{\mathrm{KL}(\mathbb{Q}_0((y)_{\le n} \mid (x,\tilde{y})_{\le n}), \mathbb{Q}_1((y)_{\le n} \mid (x,\tilde{y})_{\le n}))}_{0},
\end{aligned}$$

where the last term is 0 because under both $\mathbb{Q}_0$ and $\mathbb{Q}_1$, $(y)_{\le n} \mid (x,\tilde{y})_{\le n}$ is the delta mass supported on $(\mathrm{sign}(\tilde{y}))_{\le n}$. As a consequence,

$$\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_1) \le \mathrm{KL}(\tilde{\mathbb{P}}_0, \tilde{\mathbb{P}}_1)$$

Also, note that $\mathcal{A}$ can be viewed as a query learning algorithm that at round i, receives $(x,\tilde{y})_{\le i-1}$ as input, and choose the next example for query (i.e., it elects to only use the thresholded value y_j's as opposed to the $\tilde{y}_j$'s). Applying Lemma E.14, we have:

$$\mathrm{KL}(\tilde{\mathbb{P}}_0, \tilde{\mathbb{P}}_1) = \sum_{i=1}^{n} \mathbb{E}\left[\mathrm{KL}(\mathbb{P}_0(\tilde{y}_i = \cdot \mid (x,\tilde{y})_{\le i-1}, x_i)), \mathbb{P}_1(\tilde{y}_i = \cdot \mid (x,\tilde{y})_{\le i-1}, x_i))\right]. \tag{11}$$

We claim that for every i and $((x,\tilde{y})_{\le i-1}, x_i) \in (\mathcal{X}\times\mathcal{Y})^{i-1}\times\mathcal{X}$ on the support of $\tilde{\mathbb{P}}_0$,

$$\mathrm{KL}(\mathbb{P}_0(\tilde{y}_i = \cdot \mid (x,\tilde{y})_{\le i-1}, x_i)), \mathbb{P}_1(\tilde{y}_i = \cdot \mid (x,\tilde{y})_{\le i-1}, x_i)) \le 3\tilde{\epsilon}^2. \tag{12}$$

First, by Lemma E.5 (deferred to the end of the proof), under H_0, conditioned on $(x,\tilde{y})_{\le i-1}$ on the support of $\tilde{\mathbb{P}}_0$, the posterior distribution of a^* is the same as $a^* \sim \mathrm{N}(0, \frac{1}{d}(1+\tilde{\epsilon})I_d)$ conditioned on the affine set $S = \left\{a \in \mathbb{R}^d : \langle a, x_l\rangle + 1 = \tilde{y}_l, \forall l \in [i-1]\right\}$. Denote $X_{i-1} = [x_1^\top; x_2^\top; \ldots; x_{i-1}^\top] \in \mathbb{R}^{(i-1)\times d}$, and $\tilde{Y}_{i-1} = (\tilde{y}_1, \ldots, \tilde{y}_{i-1})$; for $(x,\tilde{y})_{\le i-1}$ on the support of $\tilde{\mathbb{P}}_0$, it must be the case that $S \ne \emptyset$, and as a result, $\hat{a} = X_{i-1}^{\dagger}(\tilde{Y}_{i-1} - \mathbb{1}_{i-1}) \in S$. Also, denote by $X_{i-1}^{\perp}$ a matrix whose columns are an orthonormal basis of $\mathrm{span}(x_1, \ldots, x_{i-1})$; such a $X_{i-1}^{\perp}$ is always well-defined as $i - 1 \le n - 1 \le d - 1$. Applying Lemma E.17, we have

$$a^* \mid (x,\tilde{y})_{\le i-1} \sim \mathrm{N}\left(\hat{a}, \frac{1}{d}(1+\tilde{\epsilon})X_{i-1}^{\perp}(X_{i-1}^{\perp})^\top\right),$$

with its covariance matrix $\frac{1}{d}(1+\tilde{\epsilon})X_{i-1}^{\perp}(X_{i-1}^{\perp})^\top$ being rank-deficient.

Now, observe that $\tilde{y}_i \mid (x,\tilde{y})_{\le i-1}, x_i$ has the same distribution as $\langle a^*, x_i\rangle + 1 \mid (x,\tilde{y})_{\le i-1}$, which is $\mathrm{N}\left(\langle \hat{a}, x_i\rangle + 1, \frac{1}{d}(1+\tilde{\epsilon})x_i^\top X_{i-1}^{\perp}(X_{i-1}^{\perp})^\top x_i\right)$.

Similarly, under H_1, we have $\tilde{y}_i \mid (x,\tilde{y})_{\le i-1}, x_i$ has distribution $\mathrm{N}\left(\langle \hat{a}, x_i\rangle + 1, \frac{1}{d}(1-\tilde{\epsilon})x_i^\top X_{i-1}^{\perp}(X_{i-1}^{\perp})^\top x_i\right)$. We now prove (12) by a case analysis:

1. If $x_i \in \mathrm{span}(x_1,\dots,x_{i-1})$, then $(X_{i-1}^{\perp})^\top x_i = 0$, and under both H_0 and H_1, the posterior distributions of $\tilde{y}_i \mid (x,\tilde{y})_{\le i-1}, x_i$ are both delta mass on $\langle \hat{a}, x_i\rangle + 1$, and therefore, $\mathrm{KL}(\mathbb{P}_0(\tilde{y}_i = \cdot \mid (x,\tilde{y})_{\le i-1}, x_i)), \mathbb{P}_1(\tilde{y}_i = \cdot \mid (x,\tilde{y})_{\le i-1}, x_i)) = 0 \le 3\tilde{\epsilon}^2$.
2. If $x_i \notin \mathrm{span}(x_1,\dots,x_{i-1})$, then $(X_{i-1}^{\perp})^\top x_i \neq 0$, and under H_0 and H_1, the posterior distributions of $\tilde{y}_i \mid (x,\tilde{y})_{\le i-1}, x_i$ are $\mathrm{N}(\hat{\mu}_i, (1+\tilde{\epsilon})\sigma_i^2)$ and $\mathrm{N}(\hat{\mu}_i, (1-\tilde{\epsilon})\sigma_i^2)$ respectively, where $\hat{\mu}_i = \langle \hat{a}, x_i\rangle + 1$, and $\sigma_i^2 = \frac{1}{d}x_i^\top X_{i-1}^{\perp}(X_{i-1}^{\perp})^\top x_i$. In this case, by Fact E.15,

$$\begin{aligned}
&\mathrm{KL}\left(\mathbb{P}_0(\tilde{y}_i = \cdot \mid (x,\tilde{y})_{\le i-1}, x_i)), \mathbb{P}_1(\tilde{y}_i = \cdot \mid (x,\tilde{y})_{\le i-1}, x_i)\right)\\
=&\mathrm{KL}\left(\mathrm{N}(\hat{\mu}_i, (1+\tilde{\epsilon})\sigma_i^2), \mathrm{N}(\hat{\mu}_i, (1-\tilde{\epsilon})\sigma_i^2)\right)\\
=&\frac{1}{2}\left(\frac{1+\tilde{\epsilon}}{1-\tilde{\epsilon}} - 1 + \ln(\frac{1-\tilde{\epsilon}}{1+\tilde{\epsilon}})\right)\\
\le&\frac{1}{2}\left(\frac{2\tilde{\epsilon}}{1-\tilde{\epsilon}}\right)^2\\
\le&8\tilde{\epsilon}^2,
\end{aligned}$$

where the first inequality is by the fact that $\ln(1+x) \ge x - x^2$ when $x \ge 0$, and taking $x = \frac{2\tilde{\epsilon}}{1-\tilde{\epsilon}}$, and the second inequality is from $\tilde{\epsilon} \le \frac{1}{2}$ and algebra.

In summary, in both cases, Equation (12) holds, and plugging this back to Equation (11) with $n \le \frac{1}{64\tilde{\epsilon}^2}$, we have $\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_1) \le 8n\tilde{\epsilon}^2 \le \frac{1}{8}$. By Pinsker's inequality (Lemma E.11), $d_{\mathrm{TV}}(\mathbb{P}_0, \mathbb{P}_1) \le \sqrt{\frac{1}{2}\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_1)} \le \frac{1}{2}$. Le Cam's lemma (Lemma E.10) implies that, for any hypothesis tester $\hat{b}$, we have

$$\frac{1}{2}\mathbb{P}_0(\hat{b}=1) + \frac{1}{2}\mathbb{P}_1(\hat{b}=0) = \frac{1}{2}(1 - d_{\mathrm{TV}}(\mathbb{P}_0, \mathbb{P}_1)) \ge \frac{1}{4}. \tag{13}$$

Step 3: concluding the proof. Given $\mathcal{A}$'s output auditing estimate $\hat{\mu}$, consider the following hypothesis tester:

$$\hat{b} = \begin{cases} 0, & \hat{\mu} > \Phi(-1), \\ 1, & \hat{\mu} \le \Phi(-1). \end{cases}$$

Plugging into Equation (7), we have

$$\frac{1}{2}\mathbb{P}_0\left(\hat{\mu} \le \Phi(-1)\right) + \frac{1}{2}\mathbb{P}_1\left(\hat{\mu} > \Phi(-1)\right) \ge \frac{1}{4}. \tag{14}$$

Now, recall Claim E.4, and using the fact that $\mathbb{P}(A \cap B) \ge \mathbb{P}(A) - \mathbb{P}(B^C) = \mathbb{P}(A) + \mathbb{P}(B) - 1$, we have

$$\mathbb{P}_0\left(\left|\hat{\mu} - \mu(h^*)\right| \ge \frac{1}{36}\tilde{\epsilon}\right) \ge \mathbb{P}_0\left(\hat{\mu} \le \Phi(-1), \mu(h^*) > \Phi(1) - \frac{1}{36}\tilde{\epsilon}\right) \ge \mathbb{P}_0\left(\hat{\mu} \le \Phi(-1)\right) + \frac{15}{16} - 1 \ge \mathbb{P}_0\left(\hat{\mu} \le \Phi(-1)\right) - \frac{1}{16}. \tag{15}$$

Symmetrically, we also have

$$\mathbb{P}_1\left(\left|\hat{\mu} - \mu(h^*)\right| \ge \frac{1}{36}\tilde{\epsilon}\right) \ge \mathbb{P}_1\left(\hat{\mu} > \Phi(-1)\right) - \frac{1}{16}. \tag{16}$$

Combining Equations (14), (15), and (16), we have

$$\frac{1}{2}\mathbb{P}_0\left(\left|\hat{\mu}-\mu(h^*)\right|\geq\frac{1}{36}\tilde{\epsilon}\right)+\frac{1}{2}\mathbb{P}_1\left(\left|\hat{\mu}-\mu(h^*)\right|\geq\frac{1}{36}\tilde{\epsilon}\right)\geq\frac{1}{4}-\frac{1}{16}>\frac{1}{8}.$$

As $\frac{1}{36}\tilde{\epsilon}\geq\epsilon$, and the left hand side can be viewed as the total probability of $\left|\hat{\mu}-\mu(h^*)\right|\geq\epsilon$ when h^* is drawn from the uniform mixture distribution of the h^* distributions under H_0 and H_1. By the probabilistic method, there exists some $h^*\in\mathcal{H}$ such that $\mathbb{P}_{h^*}\left(\left|\hat{\mu}-\mu(h^*)\right|>\epsilon\right)>\frac{1}{8}$. □

Lemma E.5. *Given the same setting above. For any fixed $i\in\mathbb{N}$ and $(x,\tilde{y})_{\leq i}$, the posterior distribution $a^*\mid(x,\tilde{y})_{\leq i}$ is the same as $a^*\mid\{a^*\in U\}$, where $U=\left\{a:\forall j\in[i]:\langle x_j,a\rangle+1=\tilde{y}_j\right\}$.*

Proof. We use the Bayes formula to expand the posterior; below $\propto$ denotes equality up to a multiplicative factor independent of a^*.

$$\begin{aligned}
\mathbb{P}(a^*\mid(x,\tilde{y})_{\leq i})\propto&\mathbb{P}(a^*,(x,\tilde{y})_{\leq i})\\
\propto&\mathbb{P}(a^*)\prod_{j=1}^{i}\mathbb{P}(x_j\mid a^*,(x,\tilde{y})_{\leq j-1})\mathbb{P}(\tilde{y}_j\mid x_j,a^*,(x,\tilde{y})_{\leq j-1})\\
\propto&\mathbb{P}(a^*)\prod_{j=1}^{i}\mathbb{P}(x_j\mid(x,\tilde{y})_{\leq j-1})\mathbb{1}\left\{\tilde{y}_j=\langle x_j,a^*\rangle+1\right\}\\
\propto&\mathbb{P}(a^*)\prod_{j=1}^{i}\mathbb{1}\left\{\tilde{y}_j=\langle x_j,a^*\rangle+1\right\}
\end{aligned}$$

where the second equality uses the definition of conditional probability; the third equality uses the fact that for any fixed query learning algorithm $\mathcal{A}$, x_j is independent of a^* conditioned on $(x,\tilde{y})_{\leq j-1}$, and the observation that given x_j and a^*, $\tilde{y}_j=\langle x_j,a^*\rangle+1$ deterministically. This concludes the proof. □

Proof of Claim E.4. For $h^*(x)=\mathrm{sign}(\langle a^*,x\rangle+b^*)$ where $b^*=-1$, it can be seen that,

$$\mathbb{P}_0(h^*(x)=+1\mid x_A=1)=0,$$

On the other hand,

$$\mathbb{P}_0(h^*(x)=+1\mid x_A=0)=\mathbb{P}_{z\sim\mathrm{N}(0,I_d)}(\langle a^*,z\rangle\geq 1)=\mathbb{P}_{z\sim\mathrm{N}(0,I_d)}\left(\left\langle\frac{a^*}{\|a^*\|},z\right\rangle\geq\frac{1}{\|a^*\|}\right)=1-\Phi\left(\frac{1}{\|a^*\|}\right).$$

Also, note that under H_0, $\frac{d\|a^*\|_2^2}{(1+\tilde{\epsilon})}\sim\chi^2(d)$; Therefore, by Fact E.16, we have that with probability $\geq\frac{15}{16}$, $\frac{d\|a^*\|_2^2}{(1+\tilde{\epsilon})}\geq d\cdot(1-10\sqrt{\frac{1}{d}})$, which implies that

$$\frac{1}{\|a^*\|}\leq\sqrt{\frac{1}{(1+\tilde{\epsilon})(1-10\sqrt{\frac{1}{d}})}}\leq\sqrt{\frac{1}{(1+\tilde{\epsilon})(1-\frac{\tilde{\epsilon}}{4})}}\leq 1-\frac{\tilde{\epsilon}}{4}.$$

Therefore, as for every $a,b\in[\frac{3}{4},1]$, $\left|\Phi(a)-\Phi(b)\right|\geq\min_{\xi\in[\frac{3}{4},1]}\Phi'(\xi)|a-b|\geq\frac{1}{9}|a-b|$, we have:

$$1-\Phi\left(\frac{1}{\|a^*\|}\right)\geq 1-\Phi\left(1-\frac{\tilde{\epsilon}}{4}\right)\geq 1-(\Phi(1)-\frac{\tilde{\epsilon}}{36})\geq\Phi(-1)+\frac{\tilde{\epsilon}}{36}.$$

This concludes the proof of the first inequality. The second inequality is proved symmetrically. □

We now present our (deterministic) active fairness auditing algorithm, Algorithm 6 and its guarantees. Algorithm 6 works under the setting when the two subpopulations are Gaussian, whose mean and covariance parameters (m_0, Σ_0), (m_1, Σ_1) are known. It also assumes access to black-box queries to $h^* \in \mathcal{H}_{\text{lin}} = \left\{h_{a,b}(x) := \text{sign}(\langle a, x\rangle + b) : a \in \mathbb{R}^d, b \in \mathbb{R}\right\}$, and aims to estimate $\mu(h^*)$ within precision ϵ. Recall that

$$\mu(h^*) = \Pr_{x\sim D_X}\left(h^*(x) = 1 \mid x_A = 0\right) - \Pr_{x\sim D_X}\left(h^*(x) = 1 \mid x_A = 1\right),$$

it suffices to estimate $\gamma_b := \Pr_{x\sim D_X}\left(h^*(x) = 1 \mid x_A = 0\right)$ within precision $\epsilon/2$, for each $b \in \{0, 1\}$. To this end, we note that

$$\gamma_b = \Pr_{x\sim \text{N}(m_b,\Sigma_b)}\left(h^*(x) = 1\right) = \Pr_{\tilde{x}\sim \text{N}(0,I_d)}\left(h^*(m_b + \Sigma_b^{1/2}\tilde{x}) = 1\right);$$

if we define $\tilde{h}_b : \mathbb{R}^d \to \{-1, +1\}$ such that

$$\tilde{h}_b(\tilde{x}) = h^*(m_b + \Sigma_b^{1/2}\tilde{x}), \tag{17}$$

γ_b equals to $\gamma(\tilde{h}_b)$, where $\gamma(h) = \mathbb{P}_{\tilde{x}\sim \text{N}(0,I_d)}\left(h(\tilde{x}) = 1\right)$ is the probability of positive prediction of h under the standard Gaussian distribution. Importantly, as h^* is a linear classifier, $\tilde{h}_b$ is also a linear classifier and lies in $\mathcal{H}_{\text{lin}}$.

Recall that procedure ESTIMATE-POSITIVE (Algorithm 4) label-efficiently estimates $\gamma(h)$ for any $h \in \mathcal{H}_{\text{lin}}$, using query access to h. Algorithm 6 uses it as a subprocedure to estimate $\gamma_b = \gamma(\tilde{h}_b)$ (line 3). To simulate label queries to $\tilde{h}_b$ using query access to h^*, according to Equation (17), it suffices to apply an affine transformation on the input $\tilde{x}$, obtaining transformed input $m_b + \Sigma_b^{1/2}\tilde{x}$, and query h^* on the transformed input.

Finally, after $\hat{\gamma}_0, \hat{\gamma}_1$, $\epsilon/2$-accurate estimators of γ_0, γ_1 are obtained, Algorithm 6 takes their difference as our estimator $\hat{\mu}$ for $\mu(h^*)$ (line 4).

Algorithm 6 Active fairness auditing for nonhomogeneous linear classifiers under Gaussian subpopulations

Require: Subpopulation parameters (m_0, Σ_0), (m_1, Σ_1), query access to $h^* \in \mathcal{H}_{\text{lin}}$, target error ϵ.
Ensure: $\hat{\mu}$ such that $\left|\hat{\mu} - \mu(h^*)\right| \le \epsilon$.
1: **for** $b \in \{0, 1\}$ **do**
2: Define $\tilde{h}_b : \mathbb{R}^d \to \{-1, +1\}$ such that $\tilde{h}_b(\tilde{x}) = h^*(m_b + \Sigma_b^{1/2}\tilde{x})$; //$\tilde{h}_b \in \mathcal{H}_{\text{lin}}$, and each query to $\tilde{h}_b$ can be simulated by one query to h^*
3: $\hat{\gamma}_b \leftarrow$ ESTIMATE-POSITIVE$(\tilde{h}_b, \frac{\epsilon}{2})$
4: **return** $\hat{\gamma}_0 - \hat{\gamma}_1$

Theorem E.6 (Upper bound)**.** *If $h^* \in \mathcal{H}_{lin}$, D_X is such that $x \mid x_A = 0 \sim \text{N}(m_0, \Sigma_0)$, $x \mid x_A = 1 \sim \text{N}(m_1, \Sigma_1)$. Algorithm 6 outputs $\hat{\mu}$, such that with probability 1, $\left|\hat{\mu} - \mu(h^*)\right| \le \epsilon$; moreover, Algorithm 6 makes at most $O(d \ln \frac{d}{\epsilon})$ label queries to h^*.*

Proof. As we will see from Lemma E.7, for $b \in \{0, 1\}$, the respective calls of ESTIMATE-POSITIVE ensures that

$$|\hat{\gamma}_b - \gamma_b| \le \frac{\epsilon}{2}.$$

Therefore,

$$\left|\hat{\mu} - \mu(h^*)\right| \le |\hat{\gamma}_0 - \gamma_0| + |\hat{\gamma}_1 - \gamma_1| \le \epsilon.$$

Moreover, for every b, Lemma E.7 ensures that each call to ESTIMATE-POSITIVE only makes at most $O(d \ln \frac{d}{\epsilon})$ label queries to $\tilde{h}_b$; as simulating each query to $\tilde{h}_b$ takes one query to h^*, for every b, it also makes at most $O(d \ln \frac{d}{\epsilon})$ label queries to h^*. Summing the number of label queries over $b \in \{0, 1\}$, the total number of label queries by Algorithm 6 is $O(d \ln \frac{d}{\epsilon})$. □

We now turn to presenting the guarantee of the key subprocedure ESTIMATE-POSITIVE and its proof. This expands the analysis sketch in Section 4.3.

Lemma E.7 (Guarantees of ESTIMATE-POSITIVE)**.** *Recall that* $\gamma(h) = \Pr_{x\sim \mathrm{N}(0,I_d)}(h(x) = +1)$. ESTIMATE-POSITIVE *(Algorithm 4) receives inputs query access to* $h^* \in \mathcal{H}_{lin}$*, and target error* ϵ*, and outputs* $\hat{\gamma}$ *such that*

$$\left|\hat{\gamma} - \gamma(h^*)\right| \leq \epsilon. \tag{18}$$

Furthermore, it makes at most $O(d \ln \frac{d}{\epsilon})$ *queries to* h^*.

Proof. Let $h^*(x) = \mathrm{sign}(\langle a^*, x\rangle + b^*)$ be the target classifier. First, observe that $\gamma(h^*) = \Phi\left(\frac{b^*}{\|a^*\|_2}\right) =: \Phi(sr)$, where Φ is the standard normal CDF, $s := \mathrm{sign}(b^*)$, and $r := \sqrt{\frac{1}{\sum_{i=1}^d m_i^{-2}}}$, for $m_i := -\frac{b^*}{a_i^*}$. Note that line 2 of ESTIMATE-POSITIVE correctly obtains s, as $s = h^*(\vec{0}) = \mathrm{sign}(\langle a^*, \vec{0}\rangle + b) = \mathrm{sign}(b)$.

Recall that $\alpha = \sqrt{2d \ln \frac{1}{\epsilon}}$ and $\beta = 2d^{\frac{5}{2}}(\ln \frac{1}{\epsilon})^{\frac{3}{4}}(\frac{1}{\epsilon})^{\frac{1}{2}}$. We consider two cases depending on the line in which ESTIMATE-POSITIVE returns:

1. If ESTIMATE-POSITIVE returns in line 5, then it must be the case that for all $i \in [d]$, $h^*(\alpha e_i) = h^*(-\alpha e_i)$. In this case, by Lemma E.9, we have that for every $i, |m_i| \geq \alpha$. This implies that $r = \sqrt{\frac{1}{\sum_{i=1}^d m_i^{-2}}} \geq \sqrt{\frac{1}{d\alpha^{-2}}} \geq \sqrt{2 \ln \frac{1}{\epsilon}}$.

   For the case that $s = -1$, we have that $\gamma(h^*) = \Phi(sr) \leq \epsilon$, where we use the standard fact that $\Phi(x) \leq \exp(-\frac{x^2}{2})$ for $x \leq 0$; in this case $\hat{\gamma} = 0$ ensures Equation (18) holds; for the symmetric case that $s = +1$, $\gamma(h^*) = \Phi(sr) \geq 1 - \epsilon$ and $\hat{\gamma} = 1$, which also ensures Equation (18).

2. On the other hand, ESTIMATE-POSITIVE returns in line 13, it must be the case that there exists some $i_0 \in [d]$, such that $|m_{i_0}| \leq \alpha$. This implies that $r = \sqrt{\frac{1}{\sum_{i=1}^d m_i^{-2}}} \leq \sqrt{\frac{1}{m_{i_0}^{-2}}} = |m_{i_0}| \leq \alpha$.

   Now, ESTIMATE-POSITIVE must execute lines 6 to 11. The final S it computes has the following properties: for every $i \in S$ added, by the guarantee of procedure BINARY-SEARCH (Algorithm 5), $|\hat{m}_i - m_i| \leq \epsilon$; otherwise, for $i \notin S$, it must be the case that $h^*(\beta e_i) \neq h^*(-\beta e_i)$, which, by Lemma E.9, implies that $|m_i| \geq \beta$. Therefore, all the conditions of Lemma 4.4 are satisfied, and thus, $|\hat{r} - r| \leq 2\epsilon$. This also yields that $|s\hat{r} - sr| \leq 2\epsilon$. Finally, note that Φ is $\frac{1}{\sqrt{2\pi}}$-Lipschitz, we have

$$\left|\hat{\gamma} - \gamma(h^*)\right| = \left|\Phi(s\hat{r}) - \Phi(sr)\right| \leq \frac{1}{\sqrt{2\pi}} \cdot |s\hat{r} - sr| \leq \epsilon.$$

In summary, in both cases, ESTIMATE-POSITIVE outputs $\hat{\gamma}$ such that Equation (18) is satisfied.

We now calculate the total query complexity of ESTIMATE-POSITIVE. Line 2 makes 1 label query; line 3 makes $2d$ label queries; for each $i \in [d]$, line 8 makes 2 label queries, and BINARY-SEARCH makes $\log \frac{2\beta}{\epsilon}$ label queries. In summary, the total label query complexity of ESTIMATE-POSITIVE is:

$$1 + 2d + d(2 + \log \frac{2\beta}{\epsilon}) = O\left(d \ln \frac{d}{\epsilon}\right).$$

□

We now present the proof of Lemma 4.4, which is key to the proof of Lemma E.7.

Proof of Lemma 4.4. First, by Lemma E.8, and the assumption that for all $i \in S, |\hat{m}_i - m_i| \leq \epsilon$, we have

$$\left|\sqrt{\frac{1}{\sum_{i\in S} \hat{m}_i^{-2}}} - \sqrt{\frac{1}{\sum_{i\in S} m_i^{-2}}}\right| \leq \epsilon.$$

It remains to prove that

$$\left|\sqrt{\frac{1}{\sum_{i\in S} m_i^{-2}}} - \sqrt{\frac{1}{\sum_{i=1}^d m_i^{-2}}}\right| \leq \epsilon,$$

which combined with the above inequality, will conclude the proof.

To see this, let $z = \sum_{i=1}^{d} m_i^{-2}$ and $z_S = \sum_{i \in S} m_i^{-2}$; since for all $i \notin S, |m_i| \geq \beta$, this implies that

$$|z - z_S| \leq \frac{d}{\beta^2} \leq \frac{2\epsilon}{(4d \ln \frac{1}{\epsilon})^{\frac{3}{2}}},$$

Also, note that $\sqrt{\frac{1}{\sum_{i=1}^{d} m_i^{-2}}} = r \leq \alpha$ implies that $z \geq \frac{1}{\alpha^2} = \frac{1}{2d \ln \frac{1}{\epsilon}}$; therefore, $z_S \geq z - \frac{2\epsilon}{(4d \ln \frac{1}{\epsilon})^{\frac{3}{2}}} \geq \frac{1}{4d \ln \frac{1}{\epsilon}}$. Now, by Lagrange mean value theorem,

$$\left| \frac{1}{\sqrt{z_S}} - \frac{1}{\sqrt{z}} \right| \leq \max_{z' \in (z_S, z)} \frac{1}{2}(z')^{-\frac{3}{2}} \cdot |z_s - z| \leq \frac{1}{2}(z_S)^{-\frac{3}{2}} \cdot |z_s - z| \leq \frac{1}{2}(4d \ln \frac{1}{\epsilon})^{\frac{3}{2}} \cdot \frac{2\epsilon}{(4d \ln \frac{1}{\epsilon})^{\frac{3}{2}}} \leq \epsilon.$$

This concludes the proof. □

Lemma E.8. *Let $l \in \mathbb{N}_+$ and $f(m_1, \ldots, m_l) := \sqrt{\frac{1}{\sum_{i=1}^{l} m_i^{-2}}}$; then f is 1-Lipschitz with respect to $\|\cdot\|_\infty$.*

Proof. First, we show that f is 1-Lipschitz with respect to $\|\cdot\|_\infty$ in each of the orthants of $\mathbb{R}^l$. Without loss of generality, we focus on the positive orthant $R =: \{m \in \mathbb{R}^l : m_i \geq 0, \forall i\}$. We now check that for any two points $\vec{m}$ and $\vec{n}$ in R, $|f(\vec{m}) - f(\vec{n})| \leq \|\vec{m} - \vec{n}\|_\infty$. By Lagrange mean value theorem, there exists some $\theta \in \{t\vec{m} + (1-t)\vec{n} : t \in (0,1)\}$, such that

$$|f(\vec{m}) - f(\vec{n})| = |\langle \nabla f(\theta), \vec{m} - \vec{n} \rangle| \leq \|\nabla f(\theta)\|_1 \|\vec{m} - \vec{n}\|_\infty,$$

where the second inequality is from Hölder's inequalty. Therefore, it suffices to check that for all $\vec{m}$ in the $R_0 =: \{\vec{m} \in \mathbb{R}^l : m_i > 0, \forall i\}$ (interior of R), $\|\nabla f(m_1, \ldots, m_l)\|_1 \leq 1$. To see this, note that

$$\nabla f(m_1, \ldots, m_d) = \left(\frac{m_1^{-3}}{(\sum_{i=1}^{l} m_i^{-2})^{\frac{3}{2}}}, \ldots, \frac{m_l^{-3}}{(\sum_{i=1}^{l} m_i^{-2})^{\frac{3}{2}}} \right) =: g,$$

Observe that $\sum_{i=1}^{l} |g_i|^{\frac{2}{3}} = 1$; this implies that for every $i \in [l], |g_i| \leq 1$, and therefore,

$$\|g\|_1 = \sum_{i=1}^{l} |g_i| \leq 1.$$

Now consider $\vec{m}, \vec{n} \in \mathbb{R}^l$ that do not necessarily lie in the same orthant. Suppose the line segment $\{t\vec{m} + (1-t)\vec{n} : t \in [0,1]\}$ consists of k pieces, where piece i is $\{t\vec{m} + (1-t)\vec{n} : t \in [t_{i-1}, t_i]\}$, where $1 = t_0 > t_1 > \ldots > t_k = 0$, where each piece is contained in an orthant. Then we have:

$$\begin{aligned}
|f(\vec{m}) - f(\vec{n})| &\leq \sum_{i=1}^{k} |f(t_{i-1}\vec{m} + (1-t_{i-1})\vec{n}) - f(t_i\vec{m} + (1-t_i)\vec{n})| \\
&\leq \sum_{i=1}^{k} \|(t_{i-1}\vec{m} + (1-t_{i-1})\vec{n}) - (t_i\vec{m} + (1-t_i)\vec{n})\|_\infty \\
&= \sum_{i=1}^{k} (t_{i-1} - t_i) \|\vec{m} - \vec{n}\|_\infty \\
&= \|\vec{m} - \vec{n}\|_\infty,
\end{aligned}$$

where the second inequality uses the Lipchitzness of f within the orthant that contains piece i, for each i in $[k]$. □

Lemma E.9. *Given $i \in [d]$ and $\xi > 0$, if $h^*(\xi e_i) = h^*(-\xi e_i)$, then $|m_i| \geq \xi$.*

Proof. Suppose $h^*(\xi e_i) = h^*(-\xi e_i) = +1$; in this case, $-b_i \leq \xi a_i^* \leq b_i$, and therefore, $|\xi a_i^*| \leq b_i$, which implies that $|m_i| \geq \xi$. The case of $h^*(\xi e_i) = h^*(-\xi e_i) = +1$ can be proved symmetrically. □

E.3. Auxiliary Lemmas for Query Learning Lower Bounds

In this subsection we collect a few standard and useful lemmas for establishing lower bounds for general adaptive sampling and query learning algorithms, including active fairness auditing algorithms. Throughout, denote by $\mathbb{P}$ the distribution of interaction transcript (the sequence of N labeled examples $\langle (x_1, y_1), \ldots, (x_N, y_N)\rangle$) obtained by the query learning algorithm by interacting with the environment, and use the shorthand $(x, y)_{\le i}$ to denote $\langle (x_1, y_1), \ldots, (x_i, y_i)\rangle$.

Lemma E.10 (Le Cam's Lemma)**.** *Given two distributions $\mathbb{P}_0$, $\mathbb{P}_1$ over observation space $z \in \mathcal{Z}$, and let $\hat{b} : \mathcal{Z} \to \{0, 1\}$ be any hypothesis tester. Then,*

$$\frac{1}{2}\mathbb{P}_0\left(\hat{b}(Z) = 1\right) + \frac{1}{2}\mathbb{P}_1\left(\hat{b}(Z) = 0\right) \ge \frac{1}{2}\left(1 - d_{\mathrm{TV}}(\mathbb{P}_0, \mathbb{P}_1)\right),$$

where $d_{\mathrm{TV}}(\mathbb{P}_0, \mathbb{P}_1)$ denotes the total variation distance between $\mathbb{P}_0$ and $\mathbb{P}_1$.

Lemma E.11 (Pinsker's Inequality)**.** *For two distributions $\mathbb{P}$ and $\mathbb{Q}$, $d_{\mathrm{TV}}(\mathbb{P}_0, \mathbb{P}_1) \le \sqrt{\frac{1}{2}\mathrm{KL}(\mathbb{P}, \mathbb{Q})}$.*

Lemma E.12 (Chain rule of KL divergence)**.** *For two distributions $\mathbb{Q}^0(Z, W)$ and $\mathbb{Q}^1(Z, W)$ over $\mathcal{Z} \times \mathcal{W}$, we have*

$$\mathrm{KL}(\mathbb{Q}^0, \mathbb{Q}^1) = \mathrm{KL}(\mathbb{Q}^0_Z, \mathbb{Q}^1_Z) + \mathbb{E}_{z\sim\mathbb{Q}^0_Z}\left[\mathrm{KL}(\mathbb{Q}^0_{W|Z}(\cdot \mid z), \mathbb{Q}^1_{W|Z}(\cdot \mid z))\right].$$

Fact E.13. Let $\mathrm{kl}(\cdot, \cdot)$ denote the binary relative entropy function. For $a, b \in [\frac{1}{4}, \frac{3}{4}]$, $\mathrm{kl}(a, b) \le 3(b - a)^2$.

The following lemma is well-known.

Lemma E.14 (Divergence decomposition)**.** *For a (possibly randomized) query learning algorithm $\mathcal{A}$ with label budget N, under two hypotheses H_0, H_1 (represented by distributions over the target concept h^*), we have:*

$$\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_1) = \sum_{i=1}^{N} \mathbb{E}\left[\mathrm{KL}(\mathbb{P}_0(y_i = \cdot \mid (x, y)_{\le i-1}, x_i)), \mathbb{P}_1(y_i = \cdot \mid (x, y)_{\le i-1}, x_i))\right]$$

Proof. We simplify $\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_1)$ as follows:

$$\begin{aligned}
\mathrm{KL}(\mathbb{P}_0, \mathbb{P}_1) &= \sum_{(x,y)_{\le N}} \mathbb{P}_0((x, y)_{\le N}) \ln \frac{\mathbb{P}_0((x, y)_{\le N})}{\mathbb{P}_0((x, y)_{\le N})} \\
&= \sum_{(x,y)_{\le N}} \mathbb{P}_0((x, y)_{\le N}) \sum_{i=1}^{N} \ln \frac{\mathbb{P}_{\mathcal{A}}(x_i \mid (x, y)_{\le i-1})}{\mathbb{P}_{\mathcal{A}}(x_i \mid (x, y)_{\le i-1})} + \ln \frac{\mathbb{P}_0(y_i \mid (x, y)_{\le i-1}, x_i)}{\mathbb{P}_1(y_i \mid (x, y)_{\le i-1}, x_i)} \\
&= \sum_{i=1}^{N} \sum_{(x,y)_{\le i}} \mathbb{P}_0((x, y)_{\le i}) \ln \frac{\mathbb{P}_0(y_i \mid (x, y)_{\le i-1}, x_i)}{\mathbb{P}_1(y_i \mid (x, y)_{\le i-1}, x_i)} \\
&= \sum_{i=1}^{N} \sum_{(x,y)_{\le i-1}, x_i} \mathbb{P}_0((x, y)_{\le i-1}, x_i) \cdot \sum_{y_i} \mathbb{P}_0(y_i \mid (x, y)_{\le i-1}, x_i) \ln \frac{\mathbb{P}_0(y_i \mid (x, y)_{\le i-1}, x_i)}{\mathbb{P}_1(y_i \mid (x, y)_{\le i-1}, x_i)} \\
&= \sum_{i=1}^{N} \mathbb{E}\left[\mathrm{KL}(\mathbb{P}_0(y_i = \cdot \mid (x, y)_{\le i-1}, x_i)), \mathbb{P}_1(y_i = \cdot \mid (x, y)_{\le i-1}, x_i))\right],
\end{aligned}$$

where the first equality is by the definition of KL divergence; the second equality is from the chain rule of conditional probability; the third equality is by canceling out the conditional probabilities of unlabeled examples given history, as we run the same algorithm $\mathcal{A}$ under two environments; the fourth equality is by the law of total probability; the fifth equality is again by the definition of the KL divergence. □

Fact E.15 (KL divergence between Gaussians of the same mean)**.** If $\mu \in \mathbb{R}$ and $\sigma_1, \sigma_2 > 0$, then,

$$\mathrm{KL}\left(\mathrm{N}(\mu, \sigma_1^2), \mathrm{N}(\mu, \sigma_2^2))\right) = \frac{\sigma_1^2}{\sigma_2^2} - 1 + \ln \frac{\sigma_2^2}{\sigma_1^2}.$$

Fact E.16 (Concentration of χ^2 random variables)**.** For $d \geq 1$, $Z \sim \chi^2(d)$, and $\delta > 0$,

$$\mathbb{P}\left(|Z - d| \leq 2\sqrt{d \ln \frac{1}{\delta}} + 2 \ln \frac{1}{\delta}\right) \geq 1 - \delta.$$

Specifically,

$$\mathbb{P}\left(|Z - d| \leq 10\sqrt{d}\right) \geq \frac{15}{16}.$$

The lemma below is a standard fact on normal distribution conditioned on affine subspaces; we include a proof here as we cannot find a reference.

Lemma E.17. *Suppose $U = \left\{\theta \in \mathbb{R}^d : X\theta = y\right\}$ is an nonempty affine subspace of $\mathbb{R}^d$, where $X \in \mathbb{R}^{m \times d}$ has rows $x_1, \ldots, x_m \in \mathbb{R}^d$. Let $\dim(\mathrm{span}(x_1, \ldots, x_m)) = l$, and let $W \in \mathbb{R}^{d \times (d-l)}$ be a matrix whose columns form an orthonormal basis of $\mathrm{span}(x_1, \ldots, x_m)^\perp$. Consider $Z \sim \mathrm{N}(0, I_d)$; then,*

$$Z \mid \{Z \in U\} \sim \mathrm{N}(X^\dagger y, WW^\top).$$

Proof. Denote by $\hat{\theta} = X^\dagger y$ the least norm solution of equation $X\theta = y$. It is well-known that $\hat{\theta} \in \mathrm{span}(x_1, \ldots, x_m)$. As $U \neq \emptyset$, $X\hat{\theta} = y$. We now claim that U can be equivalently written as $\left\{\hat{\theta} + W\alpha : \alpha \in \mathbb{R}^{d-l}\right\}$:

1. On one hand, for all $\theta = \hat{\theta} + W\alpha$, $X\theta = X\hat{\theta} + XW\alpha = y + 0 = y$.
2. On the other hand, for every $\theta \in U$, as $X\theta = y$, we have $X(\theta - \hat{\theta}) = \vec{0}$, which implies that $\theta - \hat{\theta} \in \mathrm{span}(x_1, \ldots, x_m)^\perp$. Therefore, there exists some $\alpha \in \mathbb{R}^{d-l}$ such that $\theta = \hat{\theta} + W\alpha$.

Define $V \in \mathbb{R}^{d \times l}$ to be a matrix whose columns form an orthonormal basis of $\mathrm{span}(x_1, \ldots, x_m)$. We also claim that given a vector $z \in \mathbb{R}^d$, $z \in U \Leftrightarrow V^\top z = V^\top \hat{\theta}$:

1. If $z \in U$, by the previous claim, $z = \hat{\theta} + W\alpha$, and therefore $V^\top z = V^\top \hat{\theta} + V^\top W\alpha = V^\top \hat{\theta}$.
2. If $V^\top z = V^\top \hat{\theta}$, then note that $z = VV^\top z + WW^\top z = VV^\top \hat{\theta} + W(W^\top z) = \hat{\theta} + W(W^\top z)$, where the last equality follows from that $\hat{\theta} \in \mathrm{span}(x_1, \ldots, x_m)$. Taking $\alpha_z = W^\top z \in \mathbb{R}^{d-l}$, we have $z = \hat{\theta} + W\alpha_z$, implying that $z \in U$.

For the rest of the proof, let $\overset{d}{=}$ denote equality in distribution. Consider random variable $Z \overset{d}{=} \mathrm{N}(0, I_d)$. Let $\epsilon_V = V^\top Z, \epsilon_W = W^\top Z$. Now, note that the matrix $T = \begin{pmatrix} W^\top \\ V^\top \end{pmatrix} \in \mathbb{R}^{d \times d}$ is a orthonormal matrix,

$$\begin{pmatrix} \epsilon_V \\ \epsilon_W \end{pmatrix} = \begin{pmatrix} V^\top \\ W^\top \end{pmatrix} Z = TZ \overset{d}{=} \mathrm{N}(0, I_d),$$

Therefore, ϵ_V, ϵ_W are two independent, standard normal random variables with distributions $\mathrm{N}(0, I_l)$ and $\mathrm{N}(0, I_{d-l})$, respectively.

Note from the second claim that the event $\{Z \in U\}$ is equivalent to $\{\epsilon_V = V^\top \hat{\theta}\}$; therefore, $\epsilon_W \mid \{Z \in U\} \overset{d}{=} \mathrm{N}(0, I_{d-l})$. As a result,

$$Z \mid \{Z \in U\} \overset{d}{=} V\epsilon_V + W\epsilon_W \mid \{Z \in U\} \overset{d}{=} \hat{\theta} + W\epsilon_W \mid \{Z \in U\} \overset{d}{=} \mathrm{N}(X^\dagger y, WW^\top). \qquad \square$$

F. Experiments

In this section, we empirically explore the shrinkage of the version space under various baseline methods and Algorithm 3. The two baseline methods of sampling we will consider are: 1) i.i.d sampling (without replacement) 2) active learning (CAL).

Procedure: We train a logistic regression model to find h^* on two datasets commonly used in Fairness literature. The first is COMPAS (Larson et al., 2016), where the two groups are defined to be Caucasian and non-Caucasian. And the second is the Student Performance Dataset, where the two groups are defined to be Female and Male. Then, we run the three methods with an alloted label budget of: $20, 50, 80, 100, 120$. These are a small fraction of the total dataset size (much smaller for COMPAS than Student Performance).

Evaluation: Our evaluation will be on the version space induced by the labels requested by the three methods. We will evaluate the version space in two ways:

1. Given $\mathcal{H}[S]$, we will compute its μ-diameter $\max_{h,h' \in \mathcal{H}[S]} \mu(h) - \mu(h')$. The μ-diameter of the version space captures the largest extent that the algorithm's μ estimate may be changed by post-hoc manipulation. The smaller it is the higher the degree of manipulation-proofness.

   To compute $\max_{h,h' \in \mathcal{H}[S]} \mu(h) - \mu(h')$, we will evaluate $\max_{h \in \mathcal{H}[S]} \mu(h)$ and $\min_{h \in \mathcal{H}[S]} \mu(h)$. Let $G_1 = \{x \in \mathcal{X} : x_A = 1\}$ and $G_0 = \{x \in \mathcal{X} : x_A = 0\}$. To implement the maximization program, we may move the constraint into the objective as a Lagrangian:

$$\max_h \frac{1}{|G_1|} \sum_{x \in G_1} \mathbb{1}\{h(x) = 1\} - \frac{1}{|G_0|} \sum_{x \in G_0} \mathbb{1}\{h(x) = 1\} + \lambda(\sum_{x \in S} \mathbb{1}\{h(x) = h^*(x)\})$$

   or equivalently:

$$\max_h \frac{1}{|G_1|} \sum_{x \in G_1} \mathbb{1}\{h(x) = 1\} + \frac{1}{|G_0|} \sum_{x \in G_0} \mathbb{1}\{h(x) = -1\} + \lambda(\sum_{x \in S} \mathbb{1}\{h(x) = h^*(x)\})$$

   As mentioned earlier, we observe that this objective may be framed as a cost-sensitive classification problem, which is commonly used in fairness literature (Agarwal et al., 2018). In particular, the cost for predicting 1 for $x \in G_1$ is $-\frac{1}{|G_1|}$ and 0 o.w, the cost for predicting 1 is 0 for $x \in G_0$ and $-\frac{1}{|G_0|}$ o.w and the cost for predicting $h^*(x)$ for $x \in S$ is $-\lambda$ and 0 o.w. By using iterative doubling and grid search, we look for the smallest λ such that we may enforce $h(x) = h^*(x)\ \forall x \in S$ (since these hard constraints) and find the maximizing h in the version space given this λ. The same procedure is applied for the minimizing h in the version space.

2. Since we may choose any $\mu(h)$ for $h \in \mathcal{H}[S]$ to return as an estimate for $\mu(h^*)$, we will evaluate $\mathbb{E}_{h \sim \text{unif}(\mathcal{H}[S])}[|\mu(h) - \mu(h^*)|]$ – this corresponds to the average error and is proportional to estimation accuracy.

   For sampling from the version space, we will use the classic hit-and-run algorithm and sample 500 models from the version space at each budget and then average the error.

Results: In terms of the μ-diameter of the version space, which may be interpreted as the maximum possible degree of post-audit manipulation of μ, we see in Figure 1 that Algorithm 3 is the best of the three methods at all budgets. This is expected since Algorithm 3 is designed to make use of $\max_{h \in \mathcal{H}[S]} \mu(h)$ and $\min_{h \in \mathcal{H}[S]} \mu(h)$ estimates in its query selection to "shrink" the version space in μ-space. Behind Algorithm 3, CAL looks to be generally better or on-par with i.i.d sampling.

In terms of estimation error, going by the average μ estimation error in the version space, we see in Figure 2 that in general, one of the active approaches outperforms that of i.i.d sampling. Between the two active approaches, there are budgets setting where one is better than the other and vice versa.

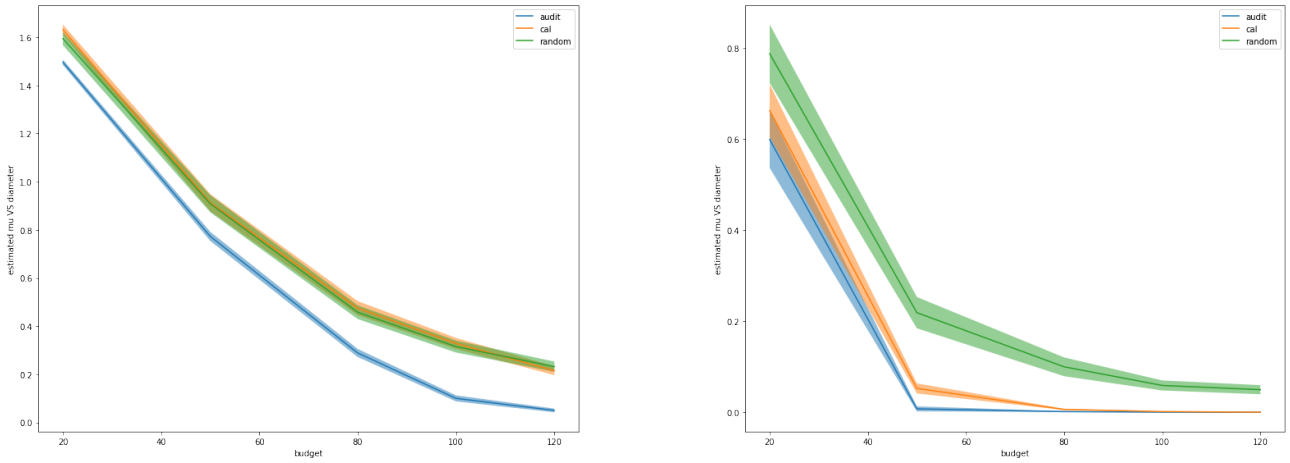

Figure 1. Left: Comparison of the three methods on the Student Performance dataset on μ-diameters of the final version spaces, as a function of label query budget. Right: Comparison of the three methods on the COMPAS dataset. For the error bars, a 95 percent confidence interval is constructed using the 50 repeats at each budget.

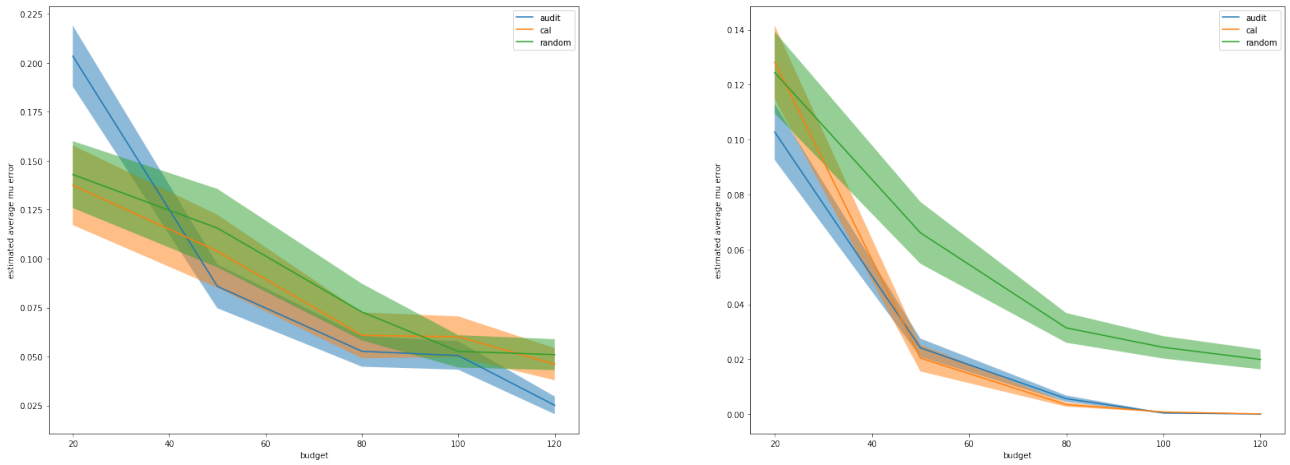

Figure 2. Left: Comparison of the three methods on the Student Performance dataset on average μ-estimation errors of the final version spaces, as a function of label query budget. Right: Comparison of the three methods on the COMPAS dataset. For the error bars, a 95 percent confidence interval is constructed using the 50 repeats at each budget.