On the Equivalence between Online and Private Learnability beyond Binary Classification

Young Hun Jung*
Department of Statistics
University of Michigan
Ann Arbor, MI 48109
yhjung@umich.edu

Baekjin Kim*
Department of Statistics
University of Michigan
Ann Arbor, MI 48109
baekjin@umich.edu

Ambuj Tewari
Department of Statistics
University of Michigan
Ann Arbor, MI 48109
tewaria@umich.edu

Abstract

Alon et al. [4] and Bun et al. [10] recently showed that online learnability and private PAC learnability are equivalent in binary classification. We investigate whether this equivalence extends to multi-class classification and regression. First, we show that private learnability implies online learnability in both settings. Our extension involves studying a novel variant of the Littlestone dimension that depends on a tolerance parameter and on an appropriate generalization of the concept of threshold functions beyond binary classification. Second, we show that while online learnability continues to imply private learnability in multi-class classification, current proof techniques encounter significant hurdles in the regression setting. While the equivalence for regression remains open, we provide non-trivial sufficient conditions for an online learnable class to also be privately learnable.

1 Introduction

Online learning and *differentially-private (DP) learning* have been well-studied in the machine learning literature. While these two subjects are seemingly unrelated, recent papers have revealed a strong connection between online and private learnability via the notion of *stability* [2, 3, 17]. The notion of differential privacy is, at its core, less about privacy and more about algorithmic stability since the output distribution of a DP algorithm should be robust to small changes in the input. Stability also plays a key role in developing online learning algorithms such as follow-the-perturbed-leader (FTPL) and follow-the-regularized-leader (FTRL) [1].

Recently Alon et al. [4] and Bun et al. [10] showed that online learnability and private PAC learnability are equivalent in binary classification. Alon et al. [4] showed that private PAC learnability implies finite Littlestone dimension (Ldim) in two steps; (i) every approximately DP learner for a class with Ldim d requires $\Omega(\log^* d)$ thresholds (see Section 2.4 for the definition of $\log^*$), and (ii) the class of thresholds over $\mathbb{N}$ cannot be learned in a private manner. Bun et al. [10] proved the converse statement via a notion of algorithmic stability, called *global stability*. They showed (i) every class with finite Ldim can be learned by a globally-stable learning algorithm and (ii) they use global stability to derive a DP algorithm. In this work, we investigate whether this equivalence extends to multi-class classification (MC) and regression, which is one of open questions raised by Bun et al. [10].

In general, online learning and private learning for MC and regression have been less studied. In binary classification without considering privacy, the Vapnik-Chervonenkis dimension (VCdim) of hypothesis classes yields tight sample complexity bounds in the batch learning setting, and Littlestone [20] defined Ldim as a combinatorial parameter that was later shown to fully characterize hypothesis classes that are learnable in the online setting [8]. Until recently, however, it was unknown what

*Equal Contribution

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

complexity measures for MC or regression classes characterize online or private learnability. Daniely et al. [11] extended the Ldim to the MC setting, and Rakhlin et al. [22] proposed the sequential fat-shattering dimension, an online counterpart of the fat-shattering dimension in the batch setting [6].

1.1 Related works

DP has been extensively studied in the machine learning literature [12, 14, 23]. Private PAC and agnostic learning were formally studied in the seminal work of Kasiviswanathan et al. [18], and the sample complexities of private learners were characterized in the later work of Beimel et al. [7].

Dwork et al. [14] identified stability as a common factor of learning and differential privacy. Abernethy et al. [2] proposed a DP-inspired stability-based methodology to design online learning algorithms with excellent theoretical guarantees, and Agarwal and Singh [3] showed that stabilization techniques such as regularization or perturbation in online learning preserve DP. Feldman and Xiao [16] relied on communication complexity to show that every purely DP learnable class has a finite Ldim. Purely DP learnability is a stronger condition than online learnability, which means that there exist online learnable classes that are not purely DP learnable. More recently, Alon et al. [4] and Bun et al. [10] established the equivalence between online and private learnability in a non-constructive manner. Gonen et al. [17] derived an efficient black-box reduction from purely DP learning to online learning. In the paper we will focus on approximate DP instead of pure DP (see Definition 2).

1.2 Main results and techniques

Our main technical contributions are as follows.

- In Section 3, we develop a novel variant of the Littlestone dimension that depends on a tolerance parameter τ, denoted by Ldim_τ. While online learnable regression problems do not naturally reduce to learnable MC problems by discretization, this relaxed complexity measure bridges online MC learnability and regression learnability in that it allows us to consider a regression problem as a relatively simpler MC problem (see Proposition 5).
- In Section 4, we show that private PAC learnability implies online learnability in both MC and regression settings. We appropriately generalize the concept of threshold functions beyond the binary classification setting and lower bound the number of these functions using the complexity measures (see Theorem 8). Then the argument of Alon et al. [4] that an infinite class of thresholds cannot be privately learned can be extended to both settings of interest.
- In Section 5, we show that while online learnability continues to imply private learnability in MC (see Theorem 11), current proof techniques based on *global stability* and *stable histogram* encounter significant obstacles in the regression problem. While this direction for regression setting still remains open, we provide non-trivial sufficient conditions for an online learnable class to also be privately learnable (see Theorem 15).

2 Preliminaries

We study multi-class classification and regression problems in this paper. In multi-class classification problems with $K \geq 2$ classes, we let $\mathcal{X}$ be the input space and $\mathcal{Y} = [K] \triangleq \{1, 2, \cdots, K\}$ be the output space, and the *standard zero-one loss* $\ell^{0-1}(\hat{y}; y) = \mathbb{I}(\hat{y} \neq y)$ is considered.

The regression problem is similar to the classification problem, except that the label becomes continuous, $\mathcal{Y} = [-1, 1]$, and the goal is to learn a real-valued function $f : \mathcal{X} \to \mathcal{Y}$ that approximates well labels of future instances. We consider the *absolute loss* $\ell^{abs}(\hat{y}; y) = |\hat{y} - y|$ in this setting. Results under the absolute loss can be generalized to any other Lipschitz losses with modified rates.

2.1 PAC learning

Let $\mathcal{X}$ be an input space, $\mathcal{Y}$ be an output space, and $\mathcal{D}$ be an unknown distribution over $\mathcal{X} \times \mathcal{Y}$. A *hypothesis* is a function mapping from $\mathcal{X}$ to $\mathcal{Y}$. The *population loss* of a hypothesis $h : \mathcal{X} \to \mathcal{Y}$ with respect to a loss function ℓ is defined by $\mathrm{loss}_{\mathcal{D}}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\big[\ell\big(h(x); y\big)\big]$. We also define the *empirical loss* of a hypothesis h with respect to a loss function ℓ and a sample $S = \big((x_i, y_i)\big)_{1:n}$

as $\text{loss}_S(h) = \frac{1}{n}\sum_{i=1}^{n} \ell\big(h(x_i); y_i\big)$. The distribution $\mathcal{D}$ is said to be *realizable* with respect to $\mathcal{H}$ if there exists $h^\star \in \mathcal{H}$ such that $\text{loss}_{\mathcal{D}}(h^\star) = 0$.

Definition 1 (PAC learning). *A hypothesis class $\mathcal{H}$ is PAC learnable with sample complexity $m(\alpha, \beta)$ if there exists an algorithm $\mathcal{A}$ such that for any $\mathcal{H}$-realizable distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, an accuracy and confidence parameters $\alpha, \beta \in (0, 1)$, if $\mathcal{A}$ is given input samples $S = \big((x_i, y_i)\big)_{1:m} \sim \mathcal{D}^m$ such that $m \geq m(\alpha, \beta)$, then it outputs a hypothesis $h : \mathcal{X} \to \mathcal{Y}$ satisfying $\text{loss}_{\mathcal{D}}(h) \leq \alpha$ with probability at least $1 - \beta$. A learner which always returns hypotheses inside the class $\mathcal{H}$ is called a proper learner, otherwise is called an improper learner.*

2.2 Differential privacy

Differential privacy (DP) [14], a standard notion of statistical data privacy, was introduced to study data analysis mechanism that do not reveal too much information on any single sample in a dataset.

Definition 2 (Differential privacy [14]). *Data samples $S, S' \in (\mathcal{X} \times \mathcal{Y})^n$ are called neighboring if they differ by exactly one example. A randomized algorithm $\mathcal{A} : (\mathcal{X} \times \mathcal{Y})^n \to \mathcal{Y}^{\mathcal{X}}$ is (ϵ, δ)-differentially private if for all neighboring data samples $S, S' \in (\mathcal{X} \times \mathcal{Y})^n$, and for all measurable sets T of outputs,*

$$\mathbb{P}\big(\mathcal{A}(S) \in T\big) \leq e^{\epsilon} \cdot \mathbb{P}\big(\mathcal{A}(S') \in T\big) + \delta.$$

The probability is taken over the randomness of $\mathcal{A}$. When $\delta = 0$ we say that $\mathcal{A}$ preserves pure differential privacy, otherwise (when $\delta > 0$) we say that $\mathcal{A}$ preserves approximate differential privacy.

Combining the requirements of PAC and DP learnability yields the definition of private PAC learner.

Definition 3 (Private PAC learning [18]). *A hypothesis class $\mathcal{H}$ is (ϵ, δ)-differentially private PAC learnable with sample complexity $m(\alpha, \beta)$ if it is PAC learnable with sample complexity $m(\alpha, \beta)$ by an algorithm $\mathcal{A}$ which is (ϵ, δ)-differentially private.*

2.3 Online learning

The online learning problem can be viewed as a repeated game between a learner and an adversary. Let T be a time horizon and $\mathcal{H} \subset \mathcal{Y}^{\mathcal{X}}$ be a class of predictors over a domain $\mathcal{X}$. At time t, the adversary chooses a pair $(x_t, y_t) \in \mathcal{X} \times \mathcal{Y}$, and the learner observes the instance x_t, predicts a label $\hat{y}_t \in \mathcal{Y}$, and finally observes the loss $\ell\big(\hat{y}_t; y_t\big)$. This work considers the *full-information setting* where the learner receives the true label information y_t. The goal is to minimize the *regret*, namely the cumulative loss that the learner actually observed compared to the best prediction in hindsight:

$$\sum_{t=1}^{T} \ell\big(\hat{y}_t; y_t\big) - \min_{h^\star \in \mathcal{H}} \sum_{t=1}^{T} \ell\big(h^\star(x_t); y_t\big).$$

A class $\mathcal{H}$ is *online learnable* if for every T, there is an algorithm that achieves sub-linear regret $o(T)$ against any sequence of T instances.

The *Littlestone dimension* is a combinatorial parameter that exactly characterizes online learnability for binary hypothesis classes [8, 20]. Daniely et al. [11] further extended this to the multi-class setting. We need the notion of mistake trees to define this complexity measure. A *mistake tree* is a binary tree whose internal nodes are labeled by elements of $\mathcal{X}$. Given a node x, its descending edges are labeled by distinct $k, k' \in \mathcal{Y}$. Then any root-to-leaf path can be expressed as a sequence of instances $\big((x_i, y_i)\big)_{1;d}$, where x_i represents the i-th internal node in the path, and y_i is the label of its descending edge in the path. We say that a tree T is *shattered* by $\mathcal{H}$ if for any root-to-leaf path $\big((x_i, y_i)\big)_{1:d}$ of T, there is $h \in \mathcal{H}$ such that $h(x_i) = y_i$ for all $i \leq d$. The Littlestone dimension of multi-class hypothesis class $\mathcal{H}$, $\text{Ldim}(\mathcal{H})$, is the maximal depth of any $\mathcal{H}$-shattered mistake tree. Just like binary classification, a set of MC hypotheses $\mathcal{H}$ is online learnable if and only if $\text{Ldim}(\mathcal{H})$ is finite.

The (sequential) *fat-shattering dimension* is the scale-sensitive complexity measure for real-valued function classes [22]. A mistake tree for real-valued function class $\mathcal{F}$ is a binary tree whose internal nodes are labeled by $(x, s) \in \mathcal{X} \times \mathcal{Y}$, where s is called a *witness to shattering*. Any root-to-leaf path in a mistake tree can be expressed as a sequence of tuples $\big((x_i, \epsilon_i)\big)_{1:d}$, where x_i is the label of the

i-th internal node in the path, and $\epsilon_i = +1$ if the $(i+1)$-th node is the right child of the i-th node, and otherwise $\epsilon_i = -1$ (for the leaf node, ϵ_d can take either value). A tree T is γ-shattered by $\mathcal{F}$ if for any root-to-leaf path $\big((x_i, \epsilon_i)\big)_{1:d}$ of T, there exists $f \in \mathcal{F}$ such that $\epsilon_i \left(f(x_i) - s_i\right) \geq \gamma/2$ for all $i \leq d$. The fat-shattering dimension at scale γ, denoted by $\mathrm{fat}_\gamma(\mathcal{F})$, is the largest d such that $\mathcal{F}$ γ-shatters a mistake tree of depth d. For any function class $\mathcal{F} \subset [-1,1]^{\mathcal{X}}$, $\mathcal{F}$ is online learnable in the supervised setting under the absolute loss if and only if $\mathrm{fat}_\gamma(\mathcal{F})$ is finite for any $\gamma > 0$ [22].

The (sequential) *Pollard pseudo-dimension* is a scale-free fat-shattering dimension for real-valued function classes. For every $f \in \mathcal{F}$, we define a binary function $B_f : \mathcal{X} \times \mathcal{Y} \to \{-1, +1\}$ by $B_f(x, s) = \mathrm{sign}\left(f(x) - s\right)$ and let $\mathcal{F}^+ = \{B_f \mid f \in \mathcal{F}\}$. Then we define the Pollard pseudo-dimension by $\mathrm{Pdim}(\mathcal{F}) = \mathrm{Ldim}(\mathcal{F}^+)$. It is easy to check that $\mathrm{fat}_\gamma(\mathcal{F}) \leq \mathrm{Pdim}(\mathcal{F})$ for all γ. That being said, finite Pollard pseudo-dimension is a sufficient condition for online learnability but not a necessary condition (e.g., bounded Lipschitz functions on [0,1] separate the two notions).

2.4 Additional notation

We define a few functions in a recursive manner. The *tower function* twr_t and the *iterated logarithm* $\log^{(m)}$ are defined respectively as

$$\mathrm{twr}_t(x) = \begin{cases} x & \text{if } t = 0, \\ 2^{\mathrm{twr}_{t-1}(x)} & \text{if } t > 0, \end{cases} \qquad \log^{(m)} x = \begin{cases} \log x & \text{if } m = 1, \\ \log^{(m-1)} \log x & \text{if } m > 1. \end{cases}$$

Lastly, we use $\log^* x$ to denote the minimal number of recursions for the iterated logarithm to return the value less than or equal to one:

$$\log^* x = \begin{cases} 0 & \text{if } x \leq 1, \\ 1 + \log^* \log x & \text{if } x > 1. \end{cases}$$

3 A link between multi-class and regression problems

As a tool to analyze regression problems, we discretize the continuous space $\mathcal{Y}$ into intervals and consider the problem as a multi-class problem. Specifically, given a function $f \in [-1,1]^{\mathcal{X}}$ and a scalar γ, we split the interval $[-1,1]$ into $\lceil \frac{2}{\gamma} \rceil$ intervals of length γ and define $[f]_\gamma(x)$ to be the index of interval that $f(x)$ belongs to. We can also define $[\mathcal{F}]_\gamma = \{[f]_\gamma \mid f \in \mathcal{F}\}$. In this way, if the multi-class problem associated with $[\mathcal{F}]_\gamma$ is learnable, we can infer that the original regression problem is learnable up to accuracy $O(\gamma)$. Quite interestingly, however, the fact that $\mathcal{F}$ is (regression) learnable does not imply that $[\mathcal{F}]_\gamma$ is (multi-class) learnable. For example, it is well known that a class $\mathcal{F}$ of bounded Lipschitz functions on [0,1] is learnable, but $[\mathcal{F}]_1$ includes all binary functions on $[0,1]$, which is not online learnable.

In order to tackle this issue, we propose a generalized zero-one loss in multi-class problems. In particular, we define a *zero-one loss with tolerance* τ,

$$\ell_\tau^{0\text{-}1}(\hat{y}; y) = \mathbb{I}(|y - \hat{y}| > \tau).$$

Note that the classical zero-one loss is simply $\ell_0^{0\text{-}1}$. This generalized loss allows the learner to predict labels that are not equal to the true label but close to it. This property is well-suited in our setting since as far as $|y - \hat{y}|$ is small, the absolute loss in the regression problem remains small.

We also extend the Littlestone dimension with tolerance τ. Fix a tolerance level τ. When we construct a mistake tree T, we add another constraint that each node's descending edges are labeled by two labels $k, k' \in [K]$ such that $\ell_\tau^{0\text{-}1}(k; k') = 1$. Let $\mathrm{Ldim}_\tau(\mathcal{H})$ be the maximal height of such binary shattered trees. (Again, $\mathrm{Ldim}_0(\mathcal{H})$ becomes the standard $\mathrm{Ldim}(\mathcal{H})$.)

We record several useful observations. The proofs can be found in Appendix A.

Lemma 4. *Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ be a class of multi-class hypotheses.*

1. $\mathrm{Ldim}_\tau(\mathcal{H})$ *is decreasing in* τ.
2. SOA_τ *(Algorithm 1) makes at most* $\mathrm{Ldim}_\tau(\mathcal{H})$ *mistakes with respect to* $\ell_\tau^{0\text{-}1}$.

Algorithm 1 Standard optimal algorithm with tolerance τ (SOA_τ)

Initialize: $V_0 = \mathcal{H}$
for $t = 1, \cdots, T$ **do**
  Receive x_t
  For $k \in [K]$, let $V_t^{(k)} = \{h \in V_{t-1} \mid h(x_t) = k\}$
  Predict $\hat{y}_t = \arg\max_k \mathrm{Ldim}_\tau(V_t^{(k)})$
  Receive true label y_t and update $V_t = V_t^{(y_t)}$
end for

3. *For any deterministic learning algorithm, an adversary can force* $\mathrm{Ldim}_{2\tau}(\mathcal{H})$ *mistakes with respect to* ℓ_τ^{0-1}.

Equipped with the relaxed loss, the following proposition connects regression learnability to multi-class learnability with discretization. We emphasize that even though the regression learnability does not imply multi-class learnability with the standard zero-one loss, learnability under ℓ_τ^{0-1} can be derived. In addition to that, it can be shown that finite $\mathrm{Ldim}_\tau([\mathcal{F}]_\gamma)$ implies finite $\mathrm{fat}_\gamma(\mathcal{F})$.

Proposition 5. *Let* $\mathcal{F} \subset [-1,1]^{\mathcal{X}}$ *be a regression hypothesis class and suppose* $\mathrm{fat}_\gamma(\mathcal{F}) = d$*. Then we have for any positive integer* n*,*

$$\mathrm{Ldim}_n([\mathcal{F}]_{\gamma/2(n+1)}) \geq d \geq \mathrm{Ldim}_n([\mathcal{F}]_{\gamma/n}).$$

Proof. Since $\mathrm{fat}_\gamma(\mathcal{F}) = d$, in the online learning setting an adversary can force any deterministic learner to suffer at least $\gamma/2$ absolute loss for d rounds. If we think of this problem as a multi-class classification problem using the hypothesis class $[\mathcal{F}]_{\gamma/2(n+1)}$, using the same strategy, the adversary can force any deterministic learner to make mistakes with respect to ℓ_n^{0-1} for d rounds. Note that the adversary reveals less information to the learner in the discretized multi-class problem. Then Lemma 4 implies $\mathrm{Ldim}_n([\mathcal{F}]_{\gamma/2(n+1)}) \geq d$.

On the other hand, suppose $\mathrm{Ldim}_n([\mathcal{F}]_{\gamma/n}) > d$ and let T be the binary shattered tree with tolerance n. For each node, we can set the witness point to be the middle point between the two labels of descending edges, and the resulting tree is γ-shattered by $\mathcal{F}$. This contradicts the fact that $\mathrm{fat}_\gamma(\mathcal{F}) = d$, and hence we obtain $d \geq \mathrm{Ldim}_n([\mathcal{F}]_{\gamma/n})$. □

There exist a few works that used regression models in multi-class classification [21, 24]. To the best of our knowledge, however, our work is the first one that studies regression learnability by transforming the problem into a discretized classification problem along with a novel bridge, *Littlestone dimension with tolerance*.

4 Private learnability implies online learnability

In this section, we show that if a class of functions is privately learnable, then it is online learnable. To do so, we prove a lower bound of the sample complexity of privately learning algorithms using either $\mathrm{Ldim}(\mathcal{H})$ for the multi-class hypotheses or $\mathrm{fat}_\gamma(\mathcal{F})$ for the regression hypotheses. Alon et al. [4] proved this in the binary classification setting first by showing that any large Ldim class contains sufficiently many threshold functions and then providing a lower bound of the sample complexity to privately learn threshold functions. We adopt their arguments, but one of the first non-trivial tasks is to define analogues of threshold functions in multi-class or regression problems. Note that, a priori, it is not clear what the right analogy is. Let us first introduce threshold functions in the binary case. We say a binary hypothesis class $\mathcal{H}$ has n thresholds if there exist $\{x_i\}_{1:n} \subset \mathcal{X}$ and $\{h_i\}_{1:n} \subset \mathcal{H}$ such that $h_i(x_j) = 1$ if $i \leq j$ and $h_i(x_j) = 0$ if $i > j$. We extend this as below.

Definition 6 (Threshold functions in multi-class problems)**.** *Let* $\mathcal{H} \subset [K]^{\mathcal{X}}$ *be a hypothesis class. We say* $\mathcal{H}$ *contains* n *thresholds with a gap* τ *if there exist* $k, k' \in [K]$*,* $\{x_i\}_{1:n} \subset \mathcal{X}$*, and* $\{h_i\}_{1:n} \subset \mathcal{H}$ *such that* $|k - k'| > \tau$ *and* $h_i(x_j) = k$ *if* $i \leq j$ *and* $h_i(x_j) = k'$ *if* $i > j$*.*

Definition 7 (Threshold functions in regression problems)**.** *Let* $\mathcal{F} \subset [-1,1]^{\mathcal{X}}$ *be a hypothesis class. We say* $\mathcal{F}$ *contains* n *thresholds with a margin* γ *if there exist* $\{x_i\}_{1:n} \subset \mathcal{X}$*,* $\{f_i\}_{1:n} \subset \mathcal{F}$*, and* $u, u' \in [-1,1]$ *such that* $|u - u'| \geq \gamma$ *and* $|f_i(x_j) - u| \leq \frac{\gamma}{20}$ *if* $i \leq j$ *and* $|f_i(x_j) - u'| \leq \frac{\gamma}{20}$ *if* $i > j$*.*

Algorithm 2 COLORANDCHOOSE

1: **Input:** multi-class hypothesis class $\mathcal{H} \subset [K]^{\mathcal{X}}$, shattered binary tree T, tolerance τ
2: Choose an arbitrary hypothesis $h_0 \in \mathcal{H}$
3: Color each vertex x of T by $h_0(x) \in [K]$
4: Find a color k such that the sub-tree $T' \subset T$ of color k has the largest height
5: Let x_0 be the root node of T'
6: Let x_1 be a child of x_0 such that the edge (x_0, x_1) is labeled as k' with $|k - k'| > \frac{\tau}{2}$
7: Let T'' be a sub-tree of T' rooted at x_1
8: Let $\mathcal{H}' = \{h \in \mathcal{H} \mid h(x_0) = k'\}$
9: **Output:** $k, k', h_0, x_0, \mathcal{H}', T''$

In Definition 7, we allow the functions to oscillate with a margin $\frac{\gamma}{20}$ which is arbitrary. Any small margin compared to $|u - u'|$ would work, but this number is chosen to facilitate later arguments.

Next we show that complex hypothesis classes contain a sufficiently large set of threshold functions. The following theorem extends the results by Alon et al. [4, Theorem 3]. A complete proof can be found in Appendix B.

Theorem 8 (Existence of a large set of thresholds)**.** *Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ and $\mathcal{F} \subset [-1, 1]^{\mathcal{X}}$ be multi-class and regression hypothesis classes, respectively.*

1. *If $\mathrm{Ldim}_{2\tau}(\mathcal{H}) \geq d$, then $\mathcal{H}$ contains $\lfloor \frac{\log_K d}{K^2} \rfloor$ thresholds with a gap τ.*

2. *If $\mathrm{fat}_\gamma(\mathcal{F}) \geq d$, then $\mathcal{F}$ contains $\lfloor \frac{\gamma^2}{10^4} \log_{100/\gamma} d \rfloor$ thresholds with a margin $\frac{\gamma}{5}$.*

Proof sketch. We begin with the multi-class setting. Suppose $d = K^{K^2 t}$. It suffices to show $\mathcal{H}$ contains t thresholds. Let T be a shattered binary tree of height d and tolerance 2τ. Letting $\mathcal{H}_0 = \mathcal{H}$ and $T_0 = T$, we iteratively apply COLORANDCHOOSE (Algorithm 2). Namely, we write

$$k_n, k'_n, h_n, x_n, \mathcal{H}_n, T_n = \textsc{ColorAndChoose}(\mathcal{H}_{n-1}, T_{n-1}, 2\tau). \tag{1}$$

Observe that for all n, we can infer $h_n(x_n) = h_n(x) = k_n$ for all internal vertices x of T_n ($\because$ line 4 of Algorithm 2) and $h(x_n) = k'_n$ for all $h \in \mathcal{H}_n$ ($\because$ line 8 of Algorithm 2).

Additionally, it can be shown that the height of T_n is no less than $\frac{1}{K}$ times the height of T_{n-1} (see Lemma 16 in Appendix B). This means that the iterative step (1) can be repeated $K^2 t$ times since $d = K^{K^2 t}$. Then there exist k, k' and indices $\{n_i\}_{i=1}^t$ such that $k_{n_i} = k$ and $k'_{n_i} = k'$ for all i.

It is not hard to check that the functions $\{h_{n_i}\}_{1:t}$ and the arguments $\{x_{n_i}\}_{1:t}$ form thresholds with labels k, k'. Since $|k - k'| > \tau$ ($\because$ line 6 of Algorithm 2), this completes the proof.

The result in the regression setting can also be shown in a similar manner using Proposition 5. □

Alon et al. [4, Theorem 1] proved a lower bound of the sample complexity in order to privately learn threshold functions. Then the multi-class result (with $\tau = 0$) of Theorem 8 immediately implies that if $\mathcal{H}$ is privately learnable, then it is online learnable. For the regression case, we need to slightly modify the argument to deal with the margin condition in Definition 7. The next theorem summarizes the result, and the proof appears in Appendix B.

Theorem 9 (Lower bound of the sample complexity to privately learn thresholds)**.** *Let $\mathcal{F} = \{f_i\}_{1:n} \subset [-1, 1]^{\mathcal{X}}$ be a set of threshold functions with a margin γ on a domain $\{x_i\}_{1:n} \subset \mathcal{X}$ along with bounds $u, u' \in [-1, 1]$. Suppose $\mathcal{A}$ is a $(\frac{\gamma}{200}, \frac{\gamma}{200})$-accurate learning algorithm for $\mathcal{F}$ with sample complexity m. If $\mathcal{A}$ is (ϵ, δ)-DP with $\epsilon = 0.1$ and $\delta = O(\frac{1}{m^2 \log m})$, then it can be shown that $m \geq \Omega(\log^* n)$.*

Combining Theorem 8 and 9, we present our main result.

Corollary 10 (Private learnability implies online learnability)**.** *Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ and $\mathcal{F} \subset [-1, 1]^{\mathcal{X}}$ be multi-class and regression hypothesis classes, respectively. Let $\mathrm{Ldim}(\mathcal{H}) = \mathrm{fat}_\gamma(\mathcal{F}) = d$. Suppose there is a learning algorithm $\mathcal{A}$ that is $(\frac{1}{16}, \frac{1}{16})$-accurate for $\mathcal{H}$ ($(\frac{\gamma}{200}, \frac{\gamma}{200})$-accurate for $\mathcal{F}$) with sample complexity m. If $\mathcal{A}$ is (ϵ, δ)-DP with $\epsilon = 0.1$ and $\delta = O(\frac{1}{m^2 \log m})$, then $m \geq \Omega(\log^* d)$.*

5 Online learnability implies private learnability

In this section, we show that online-learnable multi-class hypothesis classes can be learned in a DP manner. For regression hypothesis classes, we provide sufficient conditions for private learnability.

5.1 Multi-class classification

Bun et al. [10] proved that every binary hypothesis class with a finite Ldim is privately learnable by introducing a new notion of algorithmic stability called *global stability* as an intermediate property between online learnability and differentially-private learnability. Their arguments can be naturally extended to MC hypothesis classes, which is summarized in the next theorem.

Theorem 11 (Online MC learning implies private MC learning)**.** *Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ be a MC hypothesis class with $\mathrm{Ldim}(\mathcal{H}) = d$. Let $\epsilon, \delta \in (0,1)$ be privacy parameters and let $\alpha, \beta \in (0, 1/2)$ be accuracy parameters. For $n = O_d\big(\frac{\log(1/\beta\delta)}{\alpha\epsilon}\big)$, there exists an (ϵ, δ)-DP learning algorithm such that for every realizable distribution $\mathcal{D}$, given an input sample $S \sim \mathcal{D}^n$, the output hypothesis $f = \mathcal{A}(S)$ satisfies $loss_{\mathcal{D}}(f) \leq \alpha$ with probability at least $1 - \beta$.*

While we consider the realizable setting in Theorem 11, a similar result also holds in the agnostic setting. The extension to the agnostic setting is discussed in Appendix C.3 due to limited space.

As a key to the proof of Theorem 11, we introduce global stability (GS) as follows.

Definition 12 (Global stability [10])**.** *Let $n \in \mathbb{N}$ be a sample size and $\eta > 0$ be a global stability parameter. An algorithm $\mathcal{A}$ is (n, η)-GS with respect to $\mathcal{D}$ if there exists a hypothesis h such that $\mathbb{P}_{S\sim\mathcal{D}^n}\big(\mathcal{A}(S) = h\big) \geq \eta$.*

Theorem 11 can be proved in two steps. We first show that every MC hypothesis class with a finite Ldim is learnable by a GS algorithm $\mathcal{A}$ (Theorem 13). Then we prove that any GS algorithm can be extended to a DP learning algorithm with a finite sample complexity.

Theorem 13 (Online MC learning implies GS learning)**.** *Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ be a MC hypothesis class with $\mathrm{Ldim}(\mathcal{H}) = d$. Let $\alpha > 0$, and $m = \big((4K)^{d+1} + 1\big) \times [\frac{d \log K}{\alpha}]$. Then there exists a randomized algorithm $G : (\mathcal{X} \times [K])^m \to [K]^{\mathcal{X}}$ such that for a realizable distribution $\mathcal{D}$ and an input sample $S \sim \mathcal{D}^m$, there exists a h such that*

$$\mathbb{P}\big(G(S) = h\big) \geq \frac{K-1}{(d+1)K^{d+1}} \quad \textit{and} \quad loss_{\mathcal{D}}(h) \leq \alpha.$$

Next, we give a brief overview on how to construct a GS learner G and a DP learner M in order to prove Theorem 11. The complete proofs are deferred to Appendix C.

5.1.1 Online multi-class learning implies globally-stable learning

Let $\mathcal{H}$ be a MC hypothesis class with $\mathrm{Ldim}(\mathcal{H}) = d$ and $\mathcal{D}$ be a realizable distribution over examples $\big(x, c(x)\big)$ where $c \in \mathcal{H}$ is an unknown target hypothesis. Recall that $\mathcal{H}$ is learnable by SOA_0 (Algorithm 1) with at most d mistakes on any realizable sequence. Prior to building a GS learner G, we construct a distribution $\mathcal{D}_k$ by appending k *tournament examples* between random samples from $\mathcal{D}$, which force SOA_0 to make at least k mistakes when run on S drawn from $\mathcal{D}_k$. Using the fact that SOA_0 identifies the true labeling function after making d mistakes, we can show that there exists $k \leq d$ and a hypothesis $f : \mathcal{X} \to [K]$ such that

$$\mathbb{P}_{S\sim\mathcal{D}_k, T\sim\mathcal{D}^n}\big(\mathrm{SOA}_0(S \circ T) = f\big) \geq K^{-d}.$$

A GS learner G is built by firstly drawing $k \in \{0, 1, \cdots, d\}$ uniformly at random and then running the SOA_0 on $S \circ T$ where $S \sim \mathcal{D}_k, T \sim \mathcal{D}^n$. The learner G outputs a good hypothesis that enjoys small population loss with probability at least $\frac{K^{-d}}{d+1}$. We defer the detailed construction of $\mathcal{D}_k$ and proofs to Appendix C.

5.1.2 Globally-stable learning implies private multi-class learning

Let G be a (η, m)-GS algorithm with respect to a target distribution $\mathcal{D}$. We run G on k independent samples of size m to non-privately produce a long list $H := (h_i)_{1:k}$. The *Stable Histogram* algorithm

is a primary tool that allows us to publish a short list of frequent hypotheses in a DP manner. The fact that G is GS ensures that some good hypotheses appear frequently in H. Then Lemma 14 implies that these good hypotheses remain in the short list with high probability. Once we obtain a short list, a generic DP learning algorithm [18] is applied to privately select an accurate hypothesis.

Lemma 14 (Stable Histogram [13, 19])**.** *Let X be any data domain. For $n \geq O(\frac{\log(1/\eta\beta\delta)}{\eta\epsilon})$, there exists an (ϵ, δ)-DP algorithm* HIST *which with probability at least $1-\beta$, on input $S = (x_i)_{1:n}$ outputs a list $L \subset X$ and a sequence of estimates $a \in [0,1]^{|L|}$ such that (i) every x with $\mathrm{Freq}_S(x) \geq \eta$ appears in L, and (ii) for every $x \in L$, the estimate a_x satisfies $|a_x - \mathrm{Freq}_S(x)| \leq \eta$ where $\mathrm{Freq}_S(x) := \big|\{i \in [n] \mid x_i = x\}\big|/n$.*

5.2 Regression

In classification, *Global Stability* was an essential intermediate property between online and private learnability. A natural approach to obtaining a DP algorithm from an online-learnable real-valued function class $\mathcal{F}$ is to transform the problem into a multi-class problem with $[\mathcal{F}]_\gamma$ for some γ and then construct a GS learner using the previous techniques. If $[\mathcal{F}]_\gamma$ is privately-learnable, then we can infer that the original regression problem is also private-learnable up to an accuracy $O(\gamma)$.

Unfortunately, however, finite $\mathrm{fat}_\gamma(\mathcal{F})$ only implies finite $\mathrm{Ldim}_1([\mathcal{F}]_\gamma)$, and $\mathrm{Ldim}([\mathcal{F}]_\gamma)$ can still be infinite (see Proposition 5). This forces us to run SOA_1 instead of SOA_0, and as a consequence, after making $\mathrm{Ldim}_1([\mathcal{F}]_\gamma)$ mistakes, the algorithm can identify the true function up to some tolerance. Therefore we only get the relaxed version of GS property as follows; there exist $k \leq d$ and a hypothesis $f : \mathcal{X} \to [K]$ such that

$$\mathbb{P}_{S\sim\mathcal{D}_k, T\sim\mathcal{D}^n}\big(\mathrm{SOA}_1(S \circ T) \approx_1 f\big) \geq (\gamma/2)^d$$

where $f \approx_1 g$ means $\sup_{x\in\mathcal{X}} \big|f(x) - g(x)\big| \leq 1$. If we proceed with this relaxed condition, it is no longer guaranteed the long list H contains a good hypothesis with sufficiently high frequency. This hinders us from using Lemma 14, and a private learner cannot be produced in this manner. The limitation of proving the equivalence in regression stems from existing proof techniques. With another method, it is still possible to show that online-learnable real-valued function classes can be learned by a DP algorithm. Instead, we provide sufficient conditions for private learnability in regression problems.

Theorem 15 (Sufficient conditions for private regression learnability)**.** *Let $\mathcal{F} \subset \mathcal{Y}^\mathcal{X}$ be a real-valued function class such that $\mathrm{fat}_\gamma(\mathcal{F}) < \infty$ for every $\gamma > 0$. If one of the following conditions holds, then $\mathcal{F}$ is privately learnable.*

1. *Either $\mathcal{F}$ or $\mathcal{X}$ is finite.*
2. *The range of $\mathcal{F}$ over $\mathcal{X}$ is finite (i.e., $\big|\{f(x) \mid f \in \mathcal{F}, x \in \mathcal{X}\}\big| < \infty$).*
3. *$\mathcal{F}$ has a finite cover with respect to the sup-norm at every scale.*
4. *$\mathcal{F}$ has a finite sequential Pollard Pseudo-dimension.*

We present the proof of Condition 4, and proofs of other conditions are deferred to Appendix C.4.

Proof of Condition 4. Assume for contradiction that there exists γ such that $\mathrm{Ldim}([\mathcal{F}]_\gamma) = \infty$. Then we can obtain a shattered tree T of an arbitrary depth. Choose an arbitrary node x. Note that its descending edges are labeled by $k, k' \in [\lceil 2/\gamma \rceil]$. We can always find a witness to shattering s between the intervals corresponding to k and k'. With these witness values, the tree T must be zero-shattered by $\mathcal{F}$. Since the depth of T can be arbitrarily large, this contradicts to $\mathrm{Pdim}(\mathcal{F})$ being finite. From this, we can claim that $\mathrm{Ldim}([\mathcal{F}]_\gamma) \leq \mathrm{Pdim}(\mathcal{F})$ for any γ. Then using the ideas in Section 5.1, we can conclude that $[\mathcal{F}]_\gamma$ is private-learnable for any γ. Therefore the original class $\mathcal{F}$ is also private-learnable. □

We emphasize that Conditions 3 and 4 do not imply each other. For example, a class of point functions $\mathcal{F}^{\text{point}} := \{\mathbb{I}(\cdot = x) \mid x \in \mathcal{X}\}$ does not have a finite sup-norm cover because any two distinct functions have the sup-norm difference one, but $\mathrm{Pdim}(\mathcal{F}^{\text{point}}) = 1$. A class $\mathcal{F}^{\text{Lip}}$ of bounded Lipschitz functions on $[0, 1]$ has an infinite sequential Pollard pseudo-dimension, but $\mathcal{F}^{\text{Lip}}$ has a finite cover with respect to the sup-norm due to compactness of $[0, 1]$ along with the Lipschitz property.

6 Discussion

We have pushed the study of the equivalence between online and private learnability beyond binary classification. We proved that private learnability implies online learnability in the MC and regression settings. We also showed the converse in the MC setting and provided sufficient conditions for an online learnable class to also be privately learnable in regression problems.

We conclude with a few suggestions for future work. First, we need to understand whether online learnability implies private learnability in the regression setting. Second, like [10], we create an improper DP learner for an online learnable class. It would be interesting to see if we can construct *proper* DP learners. Third, Gonen et al. [17] provide an efficient black-box reduction from *pure* DP learning to online learning. It is natural to explore whether such efficient reductions are possible for *approximate* DP algorithms for MC and regression problems. Finally, there are huge gaps between the lower and upper bounds for sample complexities in both classification and regression settings. It would be desirable to show tighter bounds and reduce these gaps.

Broader Impact

As this paper is purely theoretical, discussing broader impact is not applicable.

Acknowledgments and Disclosure of Funding

We acknowledge the support of NSF via grants CAREER IIS-1452099 and IIS-2007055.

References

[1] Jacob Abernethy, Chansoo Lee, Abhinav Sinha, and Ambuj Tewari. Online linear optimization via smoothing. In *Conference on Learning Theory*, pages 807–823, 2014.

[2] Jacob D Abernethy, Young Hun Jung, Chansoo Lee, Audra McMillan, and Ambuj Tewari. Online learning via the differential privacy lens. In *Advances in Neural Information Processing Systems*, pages 8892–8902, 2019.

[3] Naman Agarwal and Karan Singh. The price of differential privacy for online learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 32–40. JMLR. org, 2017.

[4] Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private pac learning implies finite littlestone dimension. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 852–860, 2019.

[5] Noga Alon, Amos Beimel, Shay Moran, and Uri Stemmer. Closure properties for private classification and online prediction. volume 125 of *Proceedings of Machine Learning Research*, pages 119–152. PMLR, 2020.

[6] Peter L Bartlett, Philip M Long, and Robert C Williamson. Fat-shattering and the learnability of real-valued functions. *Journal of Computer and System Sciences*, 52(3):434–452, 1996.

[7] Amos Beimel, Kobbi Nissim, and Uri Stemmer. Characterizing the sample complexity of private learners. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 97–110, 2013.

[8] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *Conference on Learning Theory*, volume 3, page 1, 2009.

[9] Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Differentially private release and learning of threshold functions. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 634–649. IEEE, 2015.

[10] Mark Bun, Roi Livni, and Shay Moran. An equivalence between private classification and online prediction. *arXiv preprint arXiv:2003.00563*, 2020.

[11] Amit Daniely, Sivan Sabato, Shai Ben-David, and Shai Shalev-Shwartz. Multiclass learnability and the erm principle. *The Journal of Machine Learning Research*, 16(1):2377–2404, 2015.

[12] Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 371–380, 2009.

[13] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[14] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[15] Paul Erdos and Richard Rado. Combinatorial theorems on classifications of subsets of a given set. *Proceedings of the London mathematical Society*, 3(1):417–439, 1952.

[16] Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. In *Conference on Learning Theory*, pages 1000–1019, 2014.

[17] Alon Gonen, Elad Hazan, and Shay Moran. Private learning implies online learning: An efficient reduction. In *Advances in Neural Information Processing Systems*, pages 8699–8709, 2019.

[18] Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

[19] Aleksandra Korolova, Krishnaram Kenthapadi, Nina Mishra, and Alexandros Ntoulas. Releasing search queries and clicks privately. In *Proceedings of the 18th international conference on World wide web*, pages 171–180, 2009.

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

[21] Katuwal Rakesh and Ponnuthurai Nagaratnam Suganthan. An ensemble of kernel ridge regression for multi-class classification. 2017.

[22] Alexander Rakhlin, Karthik Sridharan, and Ambuj Tewari. Online learning via sequential complexities. *The Journal of Machine Learning Research*, 16(1):155–186, 2015.

[23] Anand D Sarwate and Kamalika Chaudhuri. Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data. *IEEE signal processing magazine*, 30(5):86–94, 2013.

[24] Zhixia Yang, Naiyang Deng, and Yingjie Tian. A multi-class classification algorithm based on ordinal regression machine. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, volume 2, pages 810–815. IEEE, 2005.

A Section 3 details

We prove Lemma 4.

Lemma 4 (restated)**.** *Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ be a class of multi-class hypotheses.*

1. $\mathrm{Ldim}_\tau(\mathcal{H})$ *is decreasing in* τ.
2. SOA_τ *(Algorithm 1) makes at most* $\mathrm{Ldim}_\tau(\mathcal{H})$ *mistakes with respect to* ℓ_τ^{0-1}.
3. *For any deterministic learning algorithm, an adversary can force* $\mathrm{Ldim}_{2\tau}(\mathcal{H})$ *mistakes with respect to* ℓ_τ^{0-1}.

Proof. Part 1 follows by observing that if T is a binary shattered tree with tolerance τ, then so is it with tolerance $\tau' < \tau$.

For part 2, assume SOA_τ makes a mistake at round t. We claim that $\mathrm{Ldim}_\tau(V_{t+1}) < \mathrm{Ldim}_\tau(V_t)$. If Ldim_τ does not decrease, we can infer that

$$\mathrm{Ldim}_\tau(V_t^{(\hat{y}_t)}) = \mathrm{Ldim}_\tau(V_t^{(y_t)}) = \mathrm{Ldim}_\tau(V_t) =: d.$$

Then we can find binary trees T_1 and T_2 of height d that are shattered by $V_t^{(\hat{y}_t)}$ and $V_t^{(y_t)}$, respectively. By concatenating T_1 and T_2 with a root node x_t and its edges labeled by $\hat{y}_t$ and y_t, we can obtain a binary tree T of height $d+1$ that is shattered by V_t. This contradicts to $\mathrm{Ldim}_\tau(V_t) = d$ and proves our assertion.

To prove part 3, let T be a binary shattered tree of height $\mathrm{Ldim}_{2\tau}(\mathcal{H})$. For a given node x, suppose the adversary shows x to the learner. Since the descending edges have labels apart from each other by more than 2τ, the adversary can choose a label that incurs a mistake with respect to ℓ_τ^{0-1}. Thus by following down the tree T from the root node, the adversary can force $\mathrm{Ldim}_{2\tau}(\mathcal{H})$ mistakes. □

B Section 4 details

In this section, the proofs omitted in Section 4 are presented.

B.1 Proof of Theorem 8

We first define *sub-trees*. Let T be a binary tree. Any node of T becomes its sub-tree of height 1. For $h > 1$, choose a node x and let T_1 and T_2 be the trees that are rooted at its two children. A sub-tree of height h is obtained by aggregating a sub-tree of height $h-1$ of T_1 and a sub-tree of height $h-1$ of T_2 at the root node x. Note that if the original tree T is shattered by some hypothesis class, then so is any sub-tree of it.

Next we prove a helper lemma.

Lemma 16. *Suppose there are n colors $C = \{c_i\}_{1:n}$ and n positive integers $\{d_i\}_{1:n}$. Let T be a binary tree of height $-(n-1) + \sum_{i=1}^n d_i$ whose vertices are colored by C. Then there exists a color c_i such that T has a sub-tree of height d_i in which all internal vertices are colored by c_i.*

Proof. We will prove by induction on $\sum_{i=1}^n d_i$. If $d_i = 1$ for all i, then the height of T becomes 1, and the statement holds trivially. Now suppose the lemma holds for any d_i's whose summation is less than N and let T have the height $N - n + 1$. Without loss of generality, we may assume that the root node x_0 is colored by c_1. We consider two sub-trees T_1, T_2 of height $N-n$ whose root nodes are children of x_0. Let $e_1 = d_1 - 1$ and $e_i = d_i$ for $i > 1$. Since $\sum_{i=1}^n e_i = N - 1$, by the inductive assumption each T_j has a sub-tree of height e_{i_j} in which all internal vertices are colored by c_{i_j}. If $i_j \neq 1$ for some j, then we are done because $e_{i_j} = d_{i_j}$. If $i_j = 1$ for all $j = 1, 2$, then merging these two trees with the node x_0 forms a sub-tree of height $e_1 + 1 = d_1$ of color c_1. This completes the inductive argument. □

Now we are ready to prove Theorem 8.

Theorem 8 (restated)**.** *Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ and $\mathcal{F} \subset [-1,1]^{\mathcal{X}}$ be multi-class and regression hypothesis classes, respectively.*

1. *If* $\mathrm{Ldim}_{2\tau}(\mathcal{H}) \geq d$*, then* $\mathcal{H}$ *contains* $\lfloor \frac{\log_K d}{K^2} \rfloor$ *thresholds with a gap* τ*.*

2. *If* $\mathrm{fat}_\gamma(\mathcal{F}) \geq d$*, then* $\mathcal{F}$ *contains* $\lfloor \frac{\gamma^2}{10^4} \log_{100/\gamma} d \rfloor$ *thresholds with a margin* $\frac{\gamma}{5}$*.*

Proof. We begin with the multi-class setting. Suppose $d = K^{K^2 t}$. It suffices to show $\mathcal{H}$ contains t thresholds. Let T be a shattered binary tree of height d and tolerance 2τ. Letting $\mathcal{H}_0 = \mathcal{H}$ and $T_0 = T$, we iteratively apply ColorAndChoose (Algorithm 2). Namely, we write

$$k_n, k'_n, h_n, x_n, \mathcal{H}_n, T_n = \textsc{ColorAndChoose}(\mathcal{H}_{n-1}, T_{n-1}, 2\tau). \tag{2}$$

Observe that for all n, we can infer $h_n(x_n) = h_n(x) = k_n$ for all internal vertices x of T_n ($\because$ line 4 of Algorithm 2) and $h(x_n) = k'_n$ for all $h \in \mathcal{H}_n$ ($\because$ line 8 of Algorithm 2).

Additionally, Lemma 16 ensures that the height of T_n is no less than $\frac{1}{K}$ times the height of T_{n-1}. This means that the iterative step (2) can be repeated $K^2 t$ times since $d = K^{K^2 t}$. Then there exist k, k' and indices $\{n_i\}_{i=1}^t$ such that $k_{n_i} = k$ and $k'_{n_i} = k'$ for all i.

It is not hard to check that the functions $\{h_{n_i}\}_{1:t}$ and the arguments $\{x_{n_i}\}_{1:t}$ form thresholds with labels k, k'. Since $|k - k'| > \tau$ ($\because$ line 6 of Algorithm 2), this completes the proof.

Now we move on to the regression setting. Proposition 5 implies that $\mathrm{Ldim}_{20}([\mathcal{F}]_{\gamma/50}) \geq \mathrm{Ldim}_{24}([\mathcal{F}]_{\gamma/50}) \geq d$. Then using the previous result in the multi-class setting, we can deduce that $[\mathcal{F}]_{\gamma/50}$ contains $n := \lfloor \frac{\gamma^2}{10^4} \log_{100/\gamma} d \rfloor$ thresholds with a gap 10. This means that there exist $k, k' \in [\frac{100}{\gamma}]$, $\{x_i\}_{1:n} \subset \mathcal{X}$, and $\{[f_i]_{\gamma/50}\}_{1:n} \subset \mathcal{H}$ such that $|k - k'| \geq 10$ and

$$[f_i]_{\gamma/50}(x_j) = \begin{cases} k & \text{if } i \leq j \\ k' & \text{if } i > j \end{cases}.$$

Let u, u' be the middles points of the intervals that correspond to the labels k, k'. Then it is easy to check that $|u - u'| \geq \gamma/5$ and

$$f_i(x_j) \in \begin{cases} [u - \frac{\gamma}{100}, u + \frac{\gamma}{100}) & \text{if } i \leq j \\ [u' - \frac{\gamma}{100}, u' + \frac{\gamma}{100}) & \text{if } i > j \end{cases}.$$

This proves the theorem. □

B.2 Proof of Theorem 9

Theorem 9 (restated)**.** *Let* $\mathcal{F} = \{f_i\}_{1:n} \subset [-1, 1]^{\mathcal{X}}$ *be a set of threshold functions with a margin* γ *on a domain* $\{x_i\}_{1:n} \subset \mathcal{X}$ *along with bounds* $u, u' \in [-1, 1]$*. Suppose* $\mathcal{A}$ *is a* $(\frac{\gamma}{200}, \frac{\gamma}{200})$*-accurate learning algorithm for* $\mathcal{F}$ *with sample complexity* m*. If* $\mathcal{A}$ *is* (ϵ, δ)*-DP with* $\epsilon = 0.1$ *and* $\delta = O(\frac{1}{m^2 \log m})$*, then it can be shown that* $m \geq \Omega(\log^* n)$*.*

Proof. The proof consists of two main lemmas. Lemma 19 proves that there is a large homogeneous set (see Definition 17). Then Lemma 21 yields the lower bound of the sample complexity when there exists a large homogeneous set. In particular, from these two lemmas, we can deduce that

$$\frac{\log^{(m)} n}{2^{O(m \log m)}} \leq 2^{O(m^2 \log^{(2)} m)}.$$

This means that there exists a constant c such that

$$\log^{(m)} n \leq e^{cm^2 \log m}.$$

Observing that $\log^* \left(\log^{(m)} n \right) \geq \left(\log^* n \right) - m$ and $\log^* \left(2^{O(m^2 \log^{(2)} m)} \right) = O(\log^* m)$, we can check the desired inequality $m \geq \Omega(\log^* n)$. □

B.2.1 Existence of a large homogenous set

Suppose $\mathcal{A}$ is a learning algorithm over a finite domain D. The hypothesis class consists of threshold functions over D with bounds u, u'. According to Definition 7, u and u' can be in an arbitrary order as long as $|u - u'| > \gamma$. But for simpler presentation, without loss of generality, we will assume $u > u'$. Also, let $\bar{u} = \frac{u+u'}{2}$. We define the following quantity:

$$\mathcal{A}_S(x) = \mathbb{P}_{f \sim \mathcal{A}(S)}\big(f(x) \geq \bar{u}\big).$$

The definition of homogenous sets (Definition 17) and Lemma 19 are adopted from Alon et al. [4]. Assume that $\mathcal{X}$ is linearly ordered. Given a training set $S = \big((x_i, y_i)\big)_{1:m}$, we say S is *increasing* if $x_1 \leq \cdots \leq x_m$. Additionally, we say S is *balanced* if $y_i = u'$ for all $i \leq \frac{m}{2}$ and $y_i = u$ for all $i > \frac{m}{2}$. Given $x \in \mathcal{X}$, we define $\operatorname{ord}_S(x) = \big|\{i \mid x_i \leq x\}\big|$. Lastly, we use $S_{\mathcal{X}}$ to denote $(x_i)_{1:m}$.

Definition 17 (m-homogeneous set)**.** *A set $D' \subset D$ is m-homogeneous with respect to a learning algorithm $\mathcal{A}$ if there are numbers $p_i \in [0,1]$ for $0 \leq i \leq m$ such that for every increasing balanced sample $S \in (D' \times \{u, u'\})^m$ and for every $x \in D' \setminus S_{\mathcal{X}}$*

$$|\mathcal{A}_S(x) - p_i| \leq \frac{1}{100m},$$

where $i = \operatorname{ord}_S(x)$.

The following theorem is a well-known result in Ramsey theory. It was originally introduced by Erdos and Rado [15] and rephrased by Alon et al. [4].

Theorem 18 (Alon et al. [4, Theorem 11])**.** *Let $s > t \geq 2$ and q be integers, and let $N \geq \operatorname{twr}_t(3sq \log q)$. Then for every coloring of the subsets of size t of a universe of size N using q colors, there is a homogeneous subset [2] of size s.*

The next lemma states that we can find a large homogeneous set.

Lemma 19 (Existence of a large homogeneous set)**.** *Let $\mathcal{A}$ be a learning algorithm over a domain D with $|D| = n$. Then there exists a set $D' \subset D$ which is m-homogeneous with respect to $\mathcal{A}$ such that*

$$|D'| \geq \frac{\log^{(m)} n}{2^{O(m \log m)}}.$$

Proof. We first define a coloring on the $(m+1)$-subsets of D. Let $B = \{x_1 < x_2 < \cdots < x_{m+1}\}$ be an $(m+1)$-subset. For each $i \in [m+1]$, let $B^{(i)} = B \setminus \{x_i\}$. Then by labeling the first half of $B^{(i)}$ by u' and the second half by u, we get a balanced increasing training set $S^{(i)}$. Then we compute p_i that is of the form $\frac{t}{100m}$ and closest to $\mathcal{A}_{S^{(i)}}(x_i)$ (in case of ties, choose the smaller one). Then we color B by the tuple $(p_i)_{1:m+1}$.

This scheme includes $(100m+1)^{m+1}$ colors, and Theorem 18 provides that there exists a set D' of size larger than

$$\frac{\log^{(m)} n}{3(100m+1)^{m+1}(m+1)\log(100m+1)} = \frac{\log^{(m)} n}{2^{O(m \log m)}}$$

such that all $(m+1)$-subsets of D' have the same color. It is easy to verify that this set is indeed m-homogeneous with respect to $\mathcal{A}$ according to Definition 17. □

B.2.2 Large homogeneous set implies the lower bound

Recall that PAC learning is defined with respect to $\operatorname{loss}_{\mathcal{D}}$ (see Definition 1). When $\operatorname{loss}_{\mathcal{D}}$ is replaced by loss_S, we say an algorithm $\mathcal{A}$ *empirically learns* a training set S. Bun et al. [9, Lemma 5.9] prove that if a hypothesis class is PAC learnable, then there exists an empirical learner as well.

Lemma 20 (Empirical learner)**.** *Suppose $\mathcal{A}$ is an (ϵ, δ)-DP PAC learner for a hypothesis class $\mathcal{H}$ that is (α, β)-accurate and has sample complexity m. Then there is an (ϵ, δ)-DP and (α, β)-accurate empirical learner for $\mathcal{H}$ with sample complexity $9m$.*

[2] A subset of the universe is homogeneous if all of its t-subsets have the same color.

The next is the main lemma.

Lemma 21 (Large homogeneous sets imply lower bounds on sample complexity)**.** *Suppose a learning algorithm $\mathcal{A}$ is (ϵ, δ)-DP with sample complexity m. Let $X = [N]$ be m-homogeneous with respect to $\mathcal{A}$. If $\epsilon = 0.1$, $\delta \leq \frac{1}{1000m^2 \log m}$, and $\mathcal{A}$ empirically learns the threshold functions with a margin γ over X with $(\frac{\gamma}{200}, \frac{\gamma}{200})$-accuracy, then*

$$N \leq 2^{O(m^2 \log^{(2)} m)}.$$

Proof. The proof is done by combining Lemma 22 and Lemma 23, which come below. □

This is the first helper lemma to prove Lemma 21. It adopts Alon et al. [4, Lemma 12].

Lemma 22. *Let $\mathcal{A}, X, m, N$ as in Lemma 21 and assume $N > 2m$. Then there exists a family $\mathcal{P} = \{P_i\}_{1:N-m}$ of distributions over $\{-1, 1\}^{N-m}$ that satisfies the following two properties.*

1. *P_i and P_j are (ϵ, δ)-indistinguishable for all $i \neq j$.*
2. *There exists $r \in [0, 1]$ such that for all $i, j \in [N - m]$,*

$$\mathbb{P}_{v \sim P_i}(v_j = 1) \begin{cases} \leq r - \frac{1}{10m} & \text{if } j < i \\ \geq r + \frac{1}{10m} & \text{if } j > i \end{cases}.$$

Proof. Let $(p_i)_{0:m}$ be the probability list associated with m-homogeneous set $X = [N]$. We first prove that there exists i^* such that $p_{i^*} - p_{i^*-1} \geq \frac{1}{4m}$. Fix an increasing balanced training set $S := \big((x_i, y_i)\big)_{1:m} \in \big(X \times \{u, u'\}\big)^m$ such that $x_i - x_{i-1} \geq 2$ for all i, which is possible by the assumption $N > 2m$. By the definition of threshold functions with a margin γ, we can infer

$$\min_f \text{loss}_S(f) \leq \frac{\gamma}{20} = 0.05\gamma,$$

where the minimum is taken over the threshold functions with a margin γ.

Furthermore, since $\mathcal{A}$ is an $(\alpha = \frac{\gamma}{200}, \beta = \frac{\gamma}{200})$-accurate empirical learner, we can bound the expected loss of $\mathcal{A}(S)$ as

$$\mathbb{E}_{f \sim \mathcal{A}(S)} \text{loss}_S(f) \leq \alpha + \beta + \min_f \text{loss}_S(f) \leq 0.06\gamma. \tag{3}$$

Also, we can lower bound the expected empirical loss by using the quantity $\mathcal{A}_S(x_i)$ as follows (recall that we assumed $u > u'$)

$$\mathbb{E}_{f \sim \mathcal{A}(S)} \text{loss}_S(h) \geq \frac{1}{m} \cdot \frac{\gamma}{2} \left(\sum_{i=1}^{m/2} [\mathcal{A}_S(x_i)] + \sum_{i=m/2+1}^{m} [1 - \mathcal{A}_S(x_i)] \right). \tag{4}$$

Combining (3) and (4), we can show that there exists $j \leq \frac{m}{2}$ such that $\mathcal{A}_S(x_j) \leq \frac{1}{4}$. Let $S' = (S \setminus \{(x_j, y_j)\}) \cup \{(x_j + 1, y_j)\}$. Since $\mathcal{A}$ is $(\epsilon = 0.1, \delta \leq \frac{1}{1000m^2 \log m})$-DP, we have

$$p_{j-1} - \frac{1}{100m} \leq \mathcal{A}_{S'}(x_j) \leq \frac{1}{4} e^{\epsilon} + \delta \leq 0.3,$$

which implies that $p_{j-1} \leq 0.3 + \frac{1}{100m} \leq \frac{1}{3}$. Similarly, we can find $k > \frac{m}{2}$ such that $p_{k+1} \geq \frac{2}{3}$. Then we can find $i^* \in [j, k + 1]$ such that $p_{i^*} - p_{i^*-1} \geq \frac{1}{4m}$, which proves our assertion.

Now we construct $\mathcal{P} = \{P_i\}_{1:N-m}$. Given i, let

$$B^{(i)} = \{1, \cdots, i^* - 1\} \cup \{i^* + i\} \cup \{i^* + N - m + 1, \cdots, N\} \subset X.$$

Observe that $B^{(i)}$ and $B^{(j)}$ only differ by one item at the position i^*. Then define $S^{(i)}$ to be the balanced increasing training set built upon $B^{(i)}$. Given a hypothesis f, we can compute a $N - m$ dimensional binary vector $v \in \{-1, 1\}^{N-m}$ such that

$$v_j = \mathbb{I}\left(f(i^* - 1 + j) \geq \bar{u}\right), \text{ where } \bar{u} = \frac{u + u'}{2}.$$

This mapping induces a distribution over $\{-1, 1\}^{N-m}$ from $\mathcal{A}(S^{(i)})$, which we define to be P_i.

Due to DP property of $\mathcal{A}$, P_i and P_j are (ϵ, δ)-indistinguishable. Furthermore, our construction of i^* ensures the second property with $r = \frac{p_{i-1} + p_i}{2}$. This completes the proof. □

The second helper lemma is shown by Alon et al. [4, Lemma 13].

Lemma 23. *Suppose the family $\mathcal{P}$ as in Lemma 22 exists. Then $N - m \leq 2^{1000m^2 \log^{(2)} m}$.*

C Section 5 details

We provide details omitted in Section 5.

C.1 Proof of Theorem 13

Let $\mathcal{H}$ be a multi-class hypothesis class with $\mathrm{Ldim}(\mathcal{H}) = d$ and $\mathcal{D}$ be a realizable distribution over examples $(x, c(x))$ where $c \in \mathcal{H}$ is an unknown target hypothesis. The globally-stable (GS) leaner G for $\mathcal{H}$ will make use of the Standard Optimal Algorithm (SOA_0, Algorithm 1).

SOA_0 can be simply extended to non-realizable sequences as follows.

Definition 24 (Extending the SOA_0 to non-realizable sequences)**.** *Consider a run of SOA_0 on examples $\big((x_i, y_i)\big)_{1:m}$, and let h_t denote the predictor used by the SOA_0 after observing the first t examples. Then after observing (x_{t+1}, y_{t+1}), proceed as below.*

- *If $\big((x_i, y_i)\big)_{1:t+1}$ is realizable by some $h \in \mathcal{H}$, then apply the usual update rule of the SOA_0 to obtain h_{t+1}.*
- *Else, set h_{t+1} as $h_{t+1}(x_{t+1}) = y_{t+1}$, and $h_{t+1}(x) = h_t(x)$ for every $x \neq x_{t+1}$. That is to say, h_{t+1} no longer belongs to $\mathcal{H}$.*

This update rule keeps updating the predictor h_t to agree with the last example while observing the sequences which are not necessarily realized by a hypothesis in $\mathcal{H}$. Due to this extension, our resulting algorithm possibly becomes improper.

The finite Littlestone class is online learnable by SOA_0 (Algorithm 1) with at most d mistakes on any realizable sequence. Prior to building a GS learner G, we define a distribution $\mathcal{D}_k$ as in Algorithm 3.

Algorithm 3 Distribution $\mathcal{D}_k$

1: $\mathcal{D}_0$: output an empty set with probability 1
2: Let $k \geq 1$. If there exists an f satisfying $\mathbb{P}_{S\sim\mathcal{D}_{k-1}, T\sim\mathcal{D}^n}\big(\mathrm{SOA}_0(S \circ T) = f\big) \geq K^{-d}$, or if $\mathcal{D}_{k-1}$ is undefined, then $\mathcal{D}_k$ is undefined
3: Else, $\mathcal{D}_k$ is defined recursively as follows
4: (i) Randomly sample $S_0, S_1 \sim \mathcal{D}_{k-1}$ and $T_0, T_1 \sim \mathcal{D}^n$
5: (ii) Let $f_0 = \mathrm{SOA}_0(S_0 \circ T_0)$ and $f_1 = \mathrm{SOA}_0(S_1 \circ T_1)$
6: (iii) If $f_0 = f_1$, go back to step (i)
7: (iv) Else, pick $x \in \{x \mid f_0(x) \neq f_1(x)\}$ and sample $y \sim [K]$ uniformly at random
8: (v) If $f_0(x) \neq y$, output $S_0 \circ T_0 \circ (x, y)$ and $S_1 \circ T_1 \circ (x, y)$ otherwise

Let k be such that $\mathcal{D}_k$ is well-defined and consider a sample S drawn from $\mathcal{D}_k$. The size of $\mathcal{D}_k$ is $k \cdot (n + 1)$, and they consist of $k \cdot n$ instances randomly drawn from $\mathcal{D}$ and k examples generated in Item 3(iv) of Algorithm 3. We call these k examples *tournament examples*. Due to the construction of $\mathcal{D}_k$, SOA_0 always errs in tournament rounds, which means that SOA_0 makes at least k mistakes when run on $S \circ T$ where $S \sim \mathcal{D}_k, T \sim \mathcal{D}^n$.

A natural way to obtain a GS learning algorithm G is to run the SOA_0 on this carefully chosen sample $S \circ T$. In fact, the output enjoys both global stability in multi-class learning and good generalization as follows.

Lemma 25 (Global Stability)**.** *There exist $k \leq d$ and a hypothesis $f : \mathcal{X} \to [K]$ such that*

$$\mathbb{P}_{S\sim\mathcal{D}_k, T\sim\mathcal{D}^n}\big(\mathrm{SOA}_0(S \circ T) = f\big) \geq K^{-d}.$$

Proof. Assume for contradiction that $\mathcal{D}_d$ is well-defined and for every f,

$$\mathbb{P}_{S\sim\mathcal{D}_k, T\sim\mathcal{D}^n}\big(\mathrm{SOA}_0(S \circ T) = f\big) < K^{-d}.$$

In each tournament example (x_i, y_i), the label y_i is drawn uniformly at random from $[K]$. Accordingly, with probability K^{-d} over $S \sim \mathcal{D}_d$, all d tournament examples are consistent with the true labeling function c and thus $S \circ T$ becomes consistent with c. Since the number of total mistakes of SOA_0 should be no more than d, we can deduce that $\mathrm{SOA}_0(S \circ T) = c$. This implies that

$$\mathbb{P}_{S \sim \mathcal{D}_k, T \sim \mathcal{D}^n}\big(\mathrm{SOA}_0(S \circ T) = c\big) \geq K^{-d},$$

which is a contradiction, and hence completes the proof. □

Lemma 26 (Generalization)**.** *Let k be such that $\mathcal{D}_k$ is well-defined. Then for every f such that*

$$\mathbb{P}_{S \sim \mathcal{D}_k, T \sim \mathcal{D}^n}\big(\mathrm{SOA}_0(S \circ T) = f\big) \geq K^{-d}$$

satisfies $\mathrm{loss}_{\mathcal{D}}(f) \leq \frac{d \log K}{n}$.

Proof. Let f be such hypothesis and let $\alpha = \mathrm{loss}_{\mathcal{D}}(f)$. We will argue that $K^{-d} \leq (1-\alpha)^n$. Then the following result is derived, $\alpha \leq \frac{d \log K}{n}$ using the fact that $(1-\alpha)^n \leq e^{-n\alpha}$.

By the property of SOA_0, $\mathrm{SOA}_0(S \circ T)$ is consistent with T. Thus, if $\mathrm{SOA}_0(S \circ T) = f$, then it must be the case that f is consistent with T. By assumption, $\mathrm{SOA}_0(S \circ T) = f$ holds with probability at least K^{-d} and f is consistent with T with probability $(1-\alpha)^n$ where n is the size of T. This gives the desired inequality. □

One challenge associated with the distribution $\mathcal{D}_k$ is computational limitation. It may require an unbounded number of samples from the target distribution $\mathcal{D}$, since during generation of tournament examples the number of samples drawn from $\mathcal{D}$ depends on how many times Item 3(i)-(iii) will be repeated. To handle this practical issue, we suggest a Monte-Carlo Variant of $\mathcal{D}_k$, $\tilde{\mathcal{D}}_k$, by setting an upper bound N of random samples drawn from $\mathcal{D}$ as an input parameter. Algorithm 4 summarizes how we construct the distribution $\tilde{\mathcal{D}}_k$.

Algorithm 4 Distribution $\tilde{\mathcal{D}}_k$

1: Let n be the auxiliary sample size and N be an upper bound on the number of samples from $\mathcal{D}$
2: $\tilde{\mathcal{D}}_0$: output an empty set with probability 1
3: Let $k \geq 1$. $\tilde{\mathcal{D}}_k$ is defined recursively by the following processes
4: ($\star$) Throughout the process, if more than N examples are drawn from $\mathcal{D}$, then output "Fail"
5: (i) Randomly sample $S_0, S_1 \sim \tilde{\mathcal{D}}_{k-1}$ and $T_0, T_1 \sim \mathcal{D}^n$
6: (ii) Let $f_0 = \mathrm{SOA}_0(S_0 \circ T_0)$ and $f_1 = \mathrm{SOA}_0(S_1 \circ T_1)$
7: (iii) If $f_0 = f_1$, go back to step (i)
8: (iv) Else, pick $x \in \{x \mid f_0(x) \neq f_1(x)\}$ and sample $y \sim [K]$ uniformly at random
9: (v) If $f_0(x) \neq y$, output $S_0 \circ T_0 \circ (x, y)$ and $S_1 \circ T_1 \circ (x, y)$ otherwise

The next step is to specify the upper bound N. The following lemma characterizes the expected sample complexity of sampling from $\mathcal{D}_k$.

Lemma 27 (Expected sample complexity of sampling from $\mathcal{D}_k$)**.** *Let k be such that $\mathcal{D}_k$ is well-defined and M_k be the number of samples from $\mathcal{D}$ when generating $S \sim \mathcal{D}_k$. Then we have $\mathbb{E}M_k \leq 4^{k+1} \cdot n$.*

Proof. Initially, $\mathbb{E}M_0 = 0$ since $\mathcal{D}_0$ outputs an empty set with probability 1. It suffices to show that for all $0 < i < k$, $\mathbb{E}M_{i+1} \leq 4\mathbb{E}M_i + 4n$ to conclude the desired inequality by induction.

Let R be the number of times Item 3(i) was executed during generation of $S \sim \mathcal{D}_{i+1}$, and R is distributed geometrically with a success probability θ, where

$$\begin{aligned}
\theta &= 1 - \mathbb{P}_{S_0,S_1,T_0,T_1}\big(\mathrm{SOA}_0(S_0 \circ T_0) = \mathrm{SOA}_0(S_1 \circ T_1)\big) \\
&= 1 - \sum_f \big(\mathbb{P}_{S,T}\big(\mathrm{SOA}_0(S \circ T) = f\big)\big)^2 \\
&\geq 1 - K^{-d}.
\end{aligned}$$

The last inequality holds because $i < k$ and hence $\mathcal{D}_i$ is well-defined, which implies that $\mathbb{P}_{S,T}\big(\mathrm{SOA}_0(S \circ T) = f\big) \leq K^{-d}$ for all f.

Let M_{i+1} be a random variable expressed as $M_{i+1} = \sum_{j=1}^{\infty} M_{i+1}^{(j)}$ where

$$M_{i+1}^{(j)} = \begin{cases} 0, & \text{if } R < j \\ \text{the number of examples from } \mathcal{D} \text{ in the } j\text{-th execution of Item 3(i)}, & \text{if } R \geq j \end{cases}.$$

Thus, we have

$$\begin{aligned} \mathbb{E}M_{i+1} &= \sum_{j=1}^{\infty} \mathbb{E}M_{i+1}^{(j)} = \sum_{j=1}^{\infty} (1-\theta)^{j-1} \cdot (2\mathbb{E}M_i + 2n) \\ &= \frac{1}{\theta} \cdot (2\mathbb{E}M_i + 2n) \leq 4\mathbb{E}M_i + 4n, \end{aligned}$$

where the last inequality holds since $\theta \geq 1 - K^{-d} \geq 1/2$ since $K \geq 2$ and $d \geq 1$. □

Equipped with Lemma 25,26, and 27, we are ready to prove Theorem 13.

Theorem 13 (restated)**.** *Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ be a MC hypothesis class with $\mathrm{Ldim}(\mathcal{H}) = d$. Let $\alpha > 0$, and $m = \big((4K)^{d+1} + 1\big) \times [\frac{d \log K}{\alpha}]$. Then there exists a randomized algorithm $G : (\mathcal{X} \times [K])^m \to [K]^{\mathcal{X}}$ such that for a realizable distribution $\mathcal{D}$ and an input sample $S \sim \mathcal{D}^m$, there exists a h such that*

$$\mathbb{P}\big(G(S) = h\big) \geq \frac{K-1}{(d+1)K^{d+1}} \quad \textit{and} \quad loss_{\mathcal{D}}(h) \leq \alpha.$$

Proof. The globally-stable algorithm G is defined in Algorithm 5.

Algorithm 5 Algorithm G

1: **Input :** target distribution $\tilde{\mathcal{D}}_k$, auxiliary sample size $n = [\frac{d \log K}{\alpha}]$, and the sample complexity upper bound $N = (4K)^{d+1} \cdot n$
2: Draw $k \in \{0, 1, \cdots, d\}$ uniformly at random
3: **Output :** $h = \mathrm{SOA}_0(S \circ T)$, where $T \sim \mathcal{D}^n, S \sim \tilde{\mathcal{D}}_k$

The sample complexity of G is $|S| + |T| \leq N + n = \big((4K)^{d+1} + 1\big) \times [\frac{d \log K}{n}]$. By Lemma 25 and 26, there exists $k^\star \leq d$ and $f^\star$ such that

$$\mathbb{P}_{S \sim \mathcal{D}_{k^\star}, T \sim \mathcal{D}^n}\big(\mathrm{SOA}(S \circ T) = f^\star\big) \geq \frac{1}{K^d}, \quad \mathrm{loss}_{\mathcal{D}}(f^\star) \leq \frac{d \log K}{n} \leq \alpha.$$

Let $M_{k^\star}$ denote the number of random examples from $\mathcal{D}$ during generation of $S \sim \mathcal{D}_{k^\star}$. We obtain the following inequality from Lemma 27 and Markov's inequality,

$$\mathbb{P}\big(M_{k^\star} > (4K)^{d+1} \cdot n\big) \leq \mathbb{P}\big(M_{k^\star} > K^{d+1} \cdot 4^{k^\star+1} \cdot n\big) \leq K^{-(d+1)}.$$

Accordingly,

$$\begin{aligned} &\mathbb{P}_{S \sim \tilde{\mathcal{D}}_{k^\star}, T \sim \mathcal{D}^n}\big(\mathrm{SOA}_0(S \circ T) = f^\star\big) \\ &\quad \geq \mathbb{P}_{S \sim \mathcal{D}_{k^\star}, T \sim \mathcal{D}^n}\big(\mathrm{SOA}_0(S \circ T) = f^\star \text{ and } M_{k^\star} \leq (4K)^{d+1} \cdot n\big) \\ &\quad \geq \mathbb{P}_{S \sim \mathcal{D}_{k^\star}, T \sim \mathcal{D}^n}\big(\mathrm{SOA}_0(S \circ T) = f^\star\big) - \mathbb{P}\big(M_{k^\star} > (4K)^{d+1} \cdot n\big) \\ &\quad \geq K^{-d} - K^{-(d+1)} = (K-1)K^{-(d+1)} \end{aligned}$$

Since $k = k^\star$ with probability $\frac{1}{d+1}$, G outputs $f^\star$ with probability at least $\frac{K-1}{(d+1)K^{d+1}}$. □

C.2 Globally-stable learning implies private multi-class learning

In this section, we utilize the GS algorithm from the previous section to derive a DP learning algorithm with a finite sample complexity. Theorem 11 establishes that online multi-class learnability implies private multi-class learnability, which can be proved by combining Theorem 13 and Theorem 28.

Theorem 28 (Globally-stable learning implies private multi-class learning)**.** *Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ be a multi-class hypothesis class. Let $G : (\mathcal{X} \times [K])^m \to [K]^{\mathcal{X}}$ be a randomized algorithm such that for a realizable distribution $\mathcal{D}$ and $S \sim \mathcal{D}^m$, there exists a hypothesis h such that $\mathbb{P}\big(G(S) = h\big) \geq \eta$ and $\text{loss}_{\mathcal{D}}(h) \leq \alpha/2$. Then for some $n = O(\frac{m \log(1/\eta\beta\delta)}{\eta\epsilon} + \frac{\log(1/\eta\beta)}{\alpha\epsilon})$, there exists an (ϵ, δ)-DP algorithm M which for n i.i.d. samples from $\mathcal{D}$, outputs a hypothesis $\hat{h}$ such that $\text{loss}_{\mathcal{D}}(\hat{h}) \leq \alpha$ with probability at least $1 - \beta$.*

To construct a private learner M, we first introduce standard tools in the DP community such as *Stable Histogram* and *Generic Private Learner*.

Lemma 14 (Stable Histogram, restated)**.** *Let X be any data domain. For $n \geq O(\frac{\log(1/\eta\beta\delta)}{\eta\epsilon})$, there exists an (ϵ, δ)-DP algorithm* HIST *which with probability at least $1 - \beta$, on input $S = (x_1, \cdots, x_n)$ outputs a list $L \in X$ and a sequence of estimates $a \in [0, 1]^{|L|}$ such that*

1. *Every x with $\text{Freq}_S(x) \geq \eta$ appears in L, and*
2. *For every $x \in L$, the estimate a_x satisfies $|a_x - \text{Freq}_S(x)| \leq \eta$,*

where $\text{Freq}_S(x) = \big|\{i \in [n] \mid x_i = x\}\big|/n$.

Lemma 29 (Generic Private Learner, [10])**.** *Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ be a collection of multi-class hypotheses. For $n = O(\frac{\log|\mathcal{H}| + \log(1/\beta)}{\alpha\epsilon})$, there exists an $(\epsilon, 0)$-DP algorithm* GENERICLEARNER *: $(\mathcal{X} \times [K])^n \to \mathcal{H}$ satisfying the following; let $\mathcal{D}$ be a distribution over $\mathcal{X} \times [K]$ such that there exists an $h^\star \in \mathcal{H}$ with $\text{loss}_{\mathcal{D}}(h^\star) \leq \alpha$. Then on input $S \sim \mathcal{D}^n$,* GENERICLEARNER *outputs, with probability at least $1 - \beta$, a hypothesis $\hat{h} \in \mathcal{H}$ such that $\text{loss}_S(\hat{h}) \leq 2\alpha$.*

Now we are ready to prove Theorem 28.

Proof of Theorem 28. The learning algorithm M is built on top of the Stable Historgram and the Generic Private Learner as described in Algorithm 6. According to Lemma 14 and 29, we choose parameters

$$k = O\big(\frac{\log(1/\eta\beta\delta)}{\eta\epsilon}\big), \quad n' = O\big(\frac{\log(1/\eta\beta)}{\alpha\epsilon}\big).$$

Algorithm 6 Differentially-Private Learner M

1: Let $S_1, \cdots, S_k$ each consist of i.i.d. samples of size m from $\mathcal{D}$. Run G on each batch of samples producing $h_1 = G(S_1), \cdots, h_k = G(S_k)$
2: Run the Stable Histogram algorithm HIST on input $H = (h_1, \cdots, h_k)$ using privacy $(\epsilon/2, \delta)$ and accuracy $(\eta/8, \beta/3)$, publishing a list L of frequent hypotheses
3: Let S' consist of n' i.i.d. samples from $\mathcal{D}$. Run GENERICLEARNER(S') using L with privacy $\epsilon/2$ and accuracy $(\alpha/2, \beta/3)$ to output a hypothesis $\hat{h}$

We show that the algorithm M is (ϵ, δ)-DP. During the executions of $G(S_1), \cdots G(S_k)$, a change to one entry in a certain S_i changes at most one outcome $h_i \in H$. Thus, differential privacy for this step is observed by taking expectations over the coin tosses of all the executions of G. Then the differential privacy for overall algorithm holds by simple composition of differentially-private HIST and GENERICLEARNER.

Next, we prove that the algorithm M is accurate. By standard generalization arguments, we have with probability at least $1 - \beta/3$,

$$\big|\text{Freq}_H(h) - \mathbb{P}_{S \sim \mathcal{D}^m}\big(G(S) = h\big)\big| \leq \frac{\eta}{8}$$

for every $h \in [K]^{\mathcal{X}}$ as long as $k \geq O(\log(1/\beta)/\eta)$. Conditioned on this event, by accuracy of HIST, with probability $1 - \beta/2$, it produces a list L containing $h^\star$ together with a sequence of estimates that are accurate to within an additive error $\eta/8$. Then, $h^\star$ appears in L with an estimate $a_{h^\star} \geq \eta - \eta/8 - \eta/8 = 3\eta/4$.

Now remove from L every item h with $a_h \leq \frac{3\eta}{4}$. Since every estimate is accurate within $\eta/8$, h appears in L such that $\text{Freq}_H(h) \geq \frac{3\eta}{4} - \frac{\eta}{8} = \frac{5\eta}{8}$. Since sum of frequencies is less than 1, the number of list L should be less than $2/\eta$ (i.e. $|L| \leq 2/\eta$). This list contains $h^\star$ such that $\text{loss}_{\mathcal{D}}(h^\star) \leq \alpha$. Hence the GENERICLEARNER identifies $h^\star$ with $\text{loss}_{\mathcal{D}}(h^\star) \leq \alpha/2$ with probability at least $1 - \beta/3$. □

C.3 Extension to the Agnostic setting

Theorem 11 showed that online MC learnability continues to imply private MC learnability in the realizable setting. A similar result also holds even when the realizability assumption is violated, which is called *agnostic setting*.

Corollary 30 (Agnostic setting : Online MC learning implies private MC learning)**.** *Let $\mathcal{H} \subset [K]^{\mathcal{X}}$ be a MC hypothesis class with $\text{Ldim}(\mathcal{H}) = d$. Let $\epsilon, \delta \in (0,1)$ be privacy parameters and let $\alpha, \beta \in (0, 1/2)$ be accuracy parameters. For $n = O_d\big(\frac{\log(1/\beta\delta)}{\alpha^2\epsilon}\big)$, there exists (ϵ, δ)-DP learning algorithm such that for every distribution $\mathcal{D}$, given an input sample $S \sim \mathcal{D}^n$, the output hypothesis $f = \mathcal{A}(S)$ satisfies*

$$\textit{loss}_{\mathcal{D}}(f) \leq \min_{h \in \mathcal{H}} \textit{loss}_{\mathcal{D}}(h) + \alpha$$

with probability at least $1 - \beta$.

Proof. Alon et al. [5, Theorem 6] propose an algorithm, $\mathcal{A}_{PrivateAgnostic}$, which transforms a private learner in the realizable setting to a private learner that can operate in the agnostic setting. The main idea is based on the standard sub-sampling method, and as a result, the transformed agnostic learner has a larger sample complexity by a factor of $1/\epsilon$. Then Corollary 30 is shown by applying $\mathcal{A}_{PrivateAgnostic}$ to the realizable learner used in Theorem 11. □

C.4 Proof of Theorem 15

We complete the proof of Theorem 15. The proof for Condition 4 is given in the main body.

Theorem 15 (restated)**.** *Let $\mathcal{F} \subset \mathcal{Y}^{\mathcal{X}}$ be a real-valued function class such that $\text{fat}_\gamma(\mathcal{F}) < \infty$ for every $\gamma > 0$. If one of the following conditions holds, then $\mathcal{F}$ is privately learnable.*

1. *Either $\mathcal{F}$ or $\mathcal{X}$ is finite.*
2. *The range of $\mathcal{F}$ over $\mathcal{X}$ is finite (i.e., $\big|\{f(x) \mid f \in \mathcal{F}, x \in \mathcal{X}\}\big| < \infty$).*
3. *$\mathcal{F}$ has a finite cover with respect to the sup-norm at every scale.*
4. *$\mathcal{F}$ has a finite sequential Pollard Pseudo-dimension.*

Proof. 1. If $|\mathcal{F}| < \infty$, then for sample complexity $n = \mathcal{O}(\frac{\log|\mathcal{F}| + \log(1/\beta)}{\alpha\epsilon})$ we directly run the ϵ-DP Generic Private Learner to output with probability at least $1 - \beta$, a hypothesis $\hat{f} \in \mathcal{F}$ such that $\text{loss}_S(\hat{f}) \leq \alpha$. Next, assume that $\mathcal{X}$ is finite. The finiteness of $\mathcal{X}$ does not imply finite $|\mathcal{F}|$ because $\mathcal{Y}$ is continuous, but we can discretize $\mathcal{F}$ at some scale γ, which gives us a finite MC hypothesis class $[\mathcal{F}]_\gamma$. It is private-learnable by ϵ-DP Generic Private Learner, and then the original class $\mathcal{F}$ is also privately-learnable within accuracy γ.

2. Observe that this regression problem is essentially a MC problem. Furthermore, $\text{Ldim}(\mathcal{F})$ by considering it as a MC problem is bounded above by $\text{fat}_\gamma(\mathcal{F})$, where γ is the minimal gap between consecutive values in the range of $\mathcal{F}$ over $\mathcal{X}$. This means that $\text{Ldim}(\mathcal{F})$ is finite, and hence by the argument of Section 5.1, $\mathcal{F}$ is privately learnable.

3. Given an accuracy α, $\mathcal{F}$ has n finite covers with a radius $r < \alpha$. We construct a set of representative function as $\mathcal{F}' = \{f_1, \cdots, f_n\} \subset \mathcal{F}$ by arbitrarily choosing a representative f_i from the i-th cover, and then run ϵ-DP Generic Private Learner on $\mathcal{F}'$ to output a hypothesis $\hat{f} \in \mathcal{F}$ with a small population loss. □