Proceedings of Machine Learning Research vol 313:1–27, 2026 37th International Conference on Algorithmic Learning Theory

Privately Learning Decision Lists and a Differentially Private Winnow

Mark Bun MBUN@BU.EDU
Boston University

William Fang WFANG@BU.EDU
Boston University

Editors: Matus Telgarsky and Jonathan Ullman

Abstract

We give new differentially private algorithms for the classic problems of learning decision lists and large-margin halfspaces in the PAC and online models. In the PAC model, we give a computationally efficient algorithm for learning decision lists with minimal sample overhead over the best non-private algorithms. In the online model, we give a private analog of the influential Winnow algorithm for learning halfspaces with mistake bound polylogarithmic in the dimension and inverse polynomial in the margin. As an application, we describe how to privately learn decision lists in the online model, qualitatively matching state-of-the art non-private guarantees.

1. Introduction

Differential privacy (Dwork et al., 2006) is a standard mathematical guarantee of individual-level privacy for algorithmic data analysis tasks, including those that arise in machine learning. In this paper, we give new differentially private learning algorithms, in the canonical PAC and online learning models, for some of the most fundamental concept classes in computational learning theory: decision lists and large-margin halfspaces

1.1. Privately PAC Learning Decision Lists

Our first main contribution concerns privately learning decision lists in the PAC model. The PAC model is the standard statistical learning model for binary classification, in which a learner is presented random examples $(x_1, y_1), \ldots, (x_n, y_n)$ where the x_i-s are drawn i.i.d. from an arbitrary unknown distribution $\mathcal{D}$ over $\mathcal{X}$, and each $y_i = f(x_i)$ for some unknown target function $f \in \mathcal{C}$. The goal of the learner is to produce a hypothesis $h : \mathcal{X} \to \{-1, 1\}$ that well-approximates f with respect to the distribution $\mathcal{D}$.

A *decision list* is a sequence of "if-then-else" rules of the form "if ℓ_1 then b_1, else if ℓ_2 then b_2, $\ldots$, else b_r" where each ℓ_i is a literal and each b_i is a decision bit. In his influential paper introducing the model of decision lists, Rivest (1987) described a computationally efficient PAC learner for this class. Rivest showed that (arbitrary length) decision lists over d variables are PAC learnable using $\tilde{O}(d)$ samples in $\mathrm{poly}(d)$ time. More generally, if instead of constraining each ℓ_i to be a literal, one allows it to be a member of an arbitrary class $\mathcal{F}$ of "features", such generalized decision lists are learnable using $\tilde{O}(|\mathcal{F}|)$ samples in $\mathrm{poly}(|\mathcal{F}|)$ time. Rivest's algorithm is a simple and intuitive empirical risk minimization procedure for finding a decision list consistent with a given sample. It proceeds by building a decision list one term at a time by just choosing any candidate term that is consistent with the portion of the sample that has not yet been correctly classified.

In practice, decision lists are widely deployed in high-stakes domains such as healthcare (where they are used to provide risk assessments) and finance. Even in the era of modern machine learning, they hold appeal for simultaneously providing expressivity, efficient learnability, and human interpretability (Rudin et al., 2022). However, their compatibility with privacy constraints remains unexplored – a critical gap for sensitive applications. Furthermore, decision lists represent a computational frontier: non-privately, they are among the most expressive natural classes known to be efficiently PAC-learnable, yet seemingly slightly more general classes (e.g., poly-size DNFs) have resisted the design of efficient algorithms for decades. This raises a fundamental question: Does adding privacy constraints undermine the efficient learnability of such frontier classes?

We address this question for decision lists by giving a differentially private analog of Rivest's algorithm achieving essentially the same guarantees.

Theorem 1 (Informal Statement of Theorem 7) *Let $\mathcal{F}$ be a set of "features" and let $\mathcal{F}$-decision lists denote the class of decision lists where each conditional rule is a member of $\mathcal{F}$. Then there is an (ε, δ)-differentially private learner for $\mathcal{F}$-decision lists using $\tilde{O}(|\mathcal{F}| \cdot \log(1/\delta)/\varepsilon)$ samples and running in $\mathrm{poly}(|\mathcal{F}|)$ time.*

Techniques. A baseline strategy for privately learning any finite concept class is to use the Exponential Mechanism (McSherry and Talwar, 2007) to perform approximate empirical risk minimization (Kasiviswanathan et al., 2008). This works to learn $\mathcal{F}$-decision lists using $\tilde{O}(|\mathcal{F}|)$ samples, but the runtime of a straightforward implementation is exponential in this quantity.

To design a computationally efficient algorithm, we follow Rivest's iterative algorithm, but instead of choosing each new term deterministically, we use the Exponential Mechanism to select a term that has low classification error with respect to the portion of the sample not yet correctly classified. This gives an efficient algorithm, but a naïve privacy analysis based on standard composition theorems yield a learner that seems to require $\tilde{O}(|\mathcal{F}|^{3/2})$ samples. Fortunately, we show that the structure of our iterative Exponential Mechanism-based learner follows the template of earlier approximate Set Cover algorithms (Gupta et al., 2009; Kaplan et al., 2019) for which a miraculously sharper privacy analysis is known.

1.2. Private Online Learning of Halfspaces via Winnow

For our second main contribution, we turn our attention to the more challenging *online mistake-bound* model of learning (Littlestone, 1988). This model captures learning from examples under minimal assumptions about how those examples are generated. When applied to binary classification, learning is modeled as a sequential T-round game between a learner and an adversary, where in each round t, the adversary presents an example $x_t \in \mathcal{X}$, the learner responds with a hypothesis $h_t : \mathcal{X} \to \{-1, 1\}$ and then the adversary reveals the true label $y_t \in \{-1, 1\}$. The learner's goal is produce hypotheses that are competitive in accuracy (as measured by the number of mistakes $h_t(x_t) \neq y_t$) with the best fixed function from a known class $\mathcal{C}$ of Boolean functions $f : \mathcal{X} \to \{-1, 1\}$. In the *realizable* case of online learning, true labels are constrained to be completely consistent with some $f \in \mathcal{C}$ and an ideal learner may aim to predict f with perfect accuracy after making a bounded number of mistakes.

A differentially private online learner is one whose entire sequence of hypotheses $h_t, \ldots, h_T$ is differentially private with respect to changing any labeled example (x_t, y_t). Most work on private online learning has focused on the problems of prediction from expert advice and online convex optimization more generally (Dwork et al., 2010a; Jain et al., 2012; Smith and Thakurta, 2013;

Jain and Thakurta, 2014; Agarwal and Singh, 2017; Kairouz et al., 2021; Asi et al., 2023a,b). Recently, Golowich and Livni (2021) honed in on the setting of binary classification described above, giving sublinear (in T) mistake bounds for learning any class of finite Littlestone dimension. Here, the Littlestone dimension is a combinatorial parameter that precisely characterizes non-private mistake bound learnability. Golowich and Livni's algorithm is a (technically sophisticated) analog of Littlestone's "Standard Optimal Algorithm" for non-private online learning (Littlestone, 1988), enhanced with techniques previously used to establish the private PAC learnability of these classes.

With the goal of designing a private online learner for decision lists, we actually tackle the more general problem of learning large-margin halfspaces (a.k.a. linear threshold functions). A halfspace with weight vector $v \in \mathbb{R}^d$, where $\|v\|_1 = 1$, is the function $\mathrm{sgn}(\langle v, x\rangle)$, and we say it has margin ρ if $|\langle v, x\rangle| \geq \rho$ for every $x \in \{-1, 1\}^d$. The other seminal algorithm from Littlestone's original paper, called Winnow, (non-privately) learns a halfspace by maintaining an estimate of the unknown weight vector v, updating the weights multiplicatively whenever it makes a prediction error. Littlestone showed that regardless of T, the Winnow algorithm is guaranteed to make at most $O(\log d/\rho^2)$ mistakes when learning any ρ-margin halfspace. Thus, the algorithm especially shines at attribute-efficient learning, where the ambient dimension d is much larger than the number of variables actually relevant to classification. For example, when learning sparse disjunctions of k out of d variables, for $k \ll d$, the Winnow algorithm described above has mistake bound $O(k^2 \log d)$ with a variant tailored to this class achieving an improved mistake bound of $O(k \log d)$. The Winnow algorithm learns length-r decision lists with mistake bound $2^{O(r)} \log n$, with an improvement to $O(r^{2D} \log n)$ when it is further promised that the decision bits alternate between $+1$ and -1 at most D times.

Our second main contribution gives the following differentially private analog of the Winnow algorithm.

Theorem 2 (Informal Statement of Theorem 12) *Let $\mathcal{C}$ be the class of halfspaces over $\{-1, 1\}$ with margin ρ. There is an (ε, δ)-differentially private online learner for $\mathcal{C}$ incurring regret $O(\mathrm{polylog}(d, T, 1/\delta)/\rho^6\varepsilon^4)$ against an oblivious adversary.*

Our differentially private Winnow has the same qualitative behavior (polylogarithmic in d and inverse polynomial in ρ) as its non-private counterpart, and thus similar behavior as Winnow for sparse disjunctions and short decision lists. Note that an explicit dependence of $\log T$ on the time horizon is generally necessary for differentially private online learning, even for the simplest of concept classes (Cohen et al., 2024; Dmitriev et al., 2024). Section 4.4 discusses how this lower bound applies to our setting.

Techniques. In a bit more detail, the non-private Winnow algorithm maintains as its hypothesis the halfspace $\mathrm{sgn}\langle w, \cdot\rangle$ where w is an estimate of the unknown weight vector v. On examples where it predicts correctly, it reuses the w it reported in the previous round. On examples (x, y) where it predicts incorrectly, it multiplies each coordinate w_j by a factor of $e^{\eta y x_j}$ for some learning rate η so as to push w in the direction of v in terms of its prediction on x.

The learned weight vectors w directly encode the given examples, so our private analog of Winnow releases randomized approximations of these instead. The first idea toward this comes from a beautiful observation from work using multiplicative weights updates for private query release (Gaboardi et al., 2014b): A random sample from a (suitably defined) multiplicative weights distribution may be viewed as an instantiation of the differentially private Exponential Mechanism (McSherry and Talwar, 2007). Thus, we can obtain a differentially private estimate of w by approximating it from a small number of random samples. This idea was also used more recently by Asi et al. (2023b) to give a private online experts algorithm.

There are two main challenges in turning this idea into a private online learner. The first is in ensuring that these approximate weight vectors are still suitable for learning. To argue that this is the case, we introduce a new (non-private) variant of the Winnow algorithm called ConfidentWinnow which is only guaranteed to stay with its current weight vector w when its prediction is highly confident and correct, i.e., $y\langle w, x\rangle > c\rho$ and $\operatorname{sgn}\langle w, x\rangle = y$ for some constant c. These confident correct predictions – ones where the inner product is large and has the right sign – are robust to random sampling, in the sense that a small random sample from w will, with high probability, yield the same prediction. We show that even in the face of an adversary who selects the rounds in which ConfidentWinnow updates its weights when it fails to make confident predictions, this algorithm updates not much more often than the original Winnow algorithm would have. This observation that we can demand the Winnow algorithm to make confident predictions without much overhead may be useful in contexts outside of differential privacy.

The second challenge is in ensuring privacy of the identities of the rounds in which an update occurs. That is, it is possible for neighboring sequences of examples to yield very different sequences of update rounds. To handle this, instead of deterministically updating in each round in which a prediction error occurs, we use the Sparse Vector technique (Dwork and Roth, 2014) to update only after a small, privately identifiable number of errors has occurred. The resulting algorithm (Algorithm 3), which interleaves applications of the Sparse Vector technique with the sampling procedure described above while maintaining private internal state, is unwieldy to perform a privacy analysis of directly. Fortunately, since it is possible to reconstruct the private weight vector from just the observed examples and the publicly released predictions, one can analyze the algorithm as though it were just a sequential composition of these standard private components.

Related Work. To our knowledge, our work is the first to explicitly study the problems of privately learning large-margin halfspaces or decision lists in the online model. Nevertheless, one could replace the 0-1 loss with a convex surrogate (e.g., hinge loss) to make these problems amenable to general techniques for private online convex optimization (Jain et al., 2012; Smith and Thakurta, 2013; Jain and Thakurta, 2014; Agarwal and Singh, 2017; Kairouz et al., 2021; Asi et al., 2023a,b). These techniques, however, incur regret bounds that scale polynomially in the dimension d, whereas our target is to design algorithms with only a polylogarithmic dependence. Moreover, most early work on private online learning focused on the general non-realizable case, where a polynomial dependence on T is necessary, and which cannot immediately be improved to $\operatorname{polylog}(T)$ under a realizability assumption.

Several more recent papers have studied the realizable setting more specifically. The aforementioned work of Golowich and Livni (2021) showed that private online classifiability, in the realizable setting, is characterized by the Littlestone dimension of the concept class. Recent work of Lyu (2025) showed that the dependence of the mistake bound on the Littlestone dimension can be improved significantly, from doubly exponential to polynomial, while preserving the logarithmic dependence on the time horizon T. Complementary work of Li et al. (2025a) developed improved private online learning algorithms for Littlestone classes against adaptive adversaries, and in the agnostic learning setting. As for lower bounds, Cohen et al. (2024) proved that mistake bound $\Omega(\log T)$ is necessary to privately learn point functions; Dmitriev et al. (2024) proved a similar result against a restricted class of learners. Li et al. (2025b) established lower bounds for more general concept classes under pure differential privacy and under approximate differential privacy with small δ.

Asi et al. (2023b) designed a private algorithm for learning from d experts incurring $\mathrm{polylog}(d, T)$ regret under the realizability assumption that there is a zero-loss expert. They extended this to realizable online convex optimization by building a cover over the concept class, and interpreting each element of the cover as an expert. This immediately gives a $\mathrm{poly}(k, \log d, \log T)$-regret algorithm for the problem of learning k-sparse disjunctions described above, but does not give polylogarithmic regret for general halfspaces, as the size of the required cover is too large.

The multiplicative weights technique is pervasive in differentially private algorithm design. In particular, the private multiplicative weights update is prevalent in the design of query release and synthetic data generation algorithms (Hardt and Rothblum, 2010; Hardt et al., 2012; Hsu et al., 2013; Gaboardi et al., 2014a; Ghazi et al., 2025) and differentially private boosting for PAC learning (Dwork et al., 2010b; Bun et al., 2020).

Finally, dimension-independent algorithms for private learning large-margin halfspaces in the PAC model are known via dimensionality reduction (Nguyễn et al., 2020) and boosting (Bun et al., 2020), though these results do not have direct consequences for the online setting.

2. Preliminaries

Decision Lists. A monotone decision list of length r is sequence of the form $(\ell_1, b_1), (\ell_2, b_2), \ldots$ $(\ell_{r-1}, b_{r-1}), b_r$ where each ℓ_i is a variable and each $b_i \in \{-1, 1\}$. It computes the function "if ℓ_1 then b_1, else if ℓ_2 then b_2, . . . , else b_r." For a class of Boolean functions $\mathcal{F}$, an $\mathcal{F}$-decision list is defined in the same way except each ℓ_i may be a function in $\mathcal{F}$. If the constant "true" function T is included in $\mathcal{F}$, we can replace the final "else" rule with (T, b_r) to allow for more concise notation.

Large-Margin Halfspaces. For a margin parameter $\rho > 0$, a ρ-margin halfspace is specified by a weight vector $v \in \mathbb{R}^d$ with $\|v\|_1 = 1$. It computes the function $\mathrm{sgn}(\langle v, x\rangle)$ and satisfies $|\langle v, x\rangle| \geq \rho$ for all $x \in \{-1, 1\}^d$.

Exponential Mechanism. Let $q : \mathcal{Z}^n \times H \to \mathbb{R}$ be a sensitivity-1 score function, in the sense that $|q(S, h) - q(S', h)| \leq 1$ for all neigboring datasets S, S'. The Exponential Mechanism $\mathcal{M}_E(S, q, \varepsilon)$ of McSherry and Talwar (2007) identifies an $h \in H$ which approximately maximizes $q(S, \cdot)$ by sampling each h with probability proportional to $\exp(-\varepsilon q(S, h)/2)$.

Proposition 3 *The Exponential Mechanism $\mathcal{M}_E$ is ε-differentially private. Moreover, for every $\tau > 0$, the Exponential Mechanism outputs a solution h that satisfies $q(S, h) \geq \max_{f\in H}(q(S, f)) - \frac{2}{\varepsilon}(\log(|H|) + \tau)$ with probability at least $1 - e^{-\tau}$ for any deviation term $\tau > 0$.*

3. Private PAC Learner for Decision Lists

In this section, we present DP-GreedyCover, a computationally and sample efficient PAC learner for decision lists. We describe our algorithm as directly handling $\mathcal{F}$-decision lists for some class of feature functions $\mathcal{F}$. Our algorithm will be differentially private and an accurate PAC learner according to the following definitions.

Definition 4 (Dwork et al. (2006); Kasiviswanathan et al. (2008)) *A learning algorithm $L : (\mathcal{X} \times \{0,1\})^n$ is (ε, δ)-differentially private if, for all neighboring datasets $S, S' \in (\mathcal{X} \times \{0,1\})^n$ (i.e., differing in a single example) and all sets of hypotheses R,*

$$\Pr[L(S) \in R] \leq e^{\varepsilon} \Pr[L(S') \in R] + \delta.$$

Definition 5 *Let $\mathcal{D}$ be a distribution over a domain $\mathcal{X}$, and let $\mathcal{C}$ be a concept class of Boolean functions over $\mathcal{X}$. Given $c, h : \mathcal{X} \to \{0,1\}$, define*

$$err_{\mathcal{D}}(c,h) = \Pr_{x \sim \mathcal{D}}[c(x) \neq h(x)].$$

An algorithm L is an (α, β)-PAC learner with sample complexity n if, for every distribution $\mathcal{D}$ and every target concept $c \in \mathcal{C}$,

$$\Pr_{h \leftarrow L(S)}[err_{\mathcal{D}}(c,h) \leq \alpha] \geq 1 - \beta,$$

where the probability is taken over $S = \{(x_i, c(x_i)\}_{i=1}^n$ where the x_i-s are drawn i.i.d. from $\mathcal{D}$, as well as any internal randomness of L.

3.1. Algorithm Overview

DP-GreedyCover operates on a sample S consisting of samples of the form (x_i, σ_i), where each $\sigma_i = c^*(x_i)$ for some unknown decision list c^*. The algorithm iteratively maintains a set S_j of unclassified samples and applies the Exponential Mechanism to select a function-label pair (f_j, b_j) with high score with respect to a quality function $q(S_j, (f_j, b_j))$ that guarantees that (f_j, b_j) errs on few examples in S_j. Then, the algorithm continues to the next iteration with S_{j+1}, the subset of the sample S_j that contains examples on which f_j evaluates to 0, and $\mathcal{F}_{j+1}$, the subset of $\mathcal{F}_j$ without f_j.

Algorithm 1: DP-GreedyCover

Input: Labeled sample $S = \{(x_i, \sigma_i)\}_{i=1}^n \in (\{0,1\}^d \times \{0,1\})^n$, set of feature functions $\mathcal{F}$
Output: Decision list $h_{fin} = [(f_1, b_1), (f_2, b_2), ..., (f_M, b_M)]$

Initialize: Let $\mathcal{F}_1 = \mathcal{F}$ together with the constant true function T, and let $S_1 = S$.
for $j = 1, 2, \ldots, M$ **do**
  Let S_j^1 and S_j^0 denote the positive and negative examples in S_j, respectively.
  for $f \in \mathcal{F}_j$ **do**
    Let $\#_{f \to 1}(S_j^1)$ and $\#_{f \to 1}(S_j^0)$ denote the number of positive and negative examples in S_j, respectively, that function f labels as 1. That is,
      $\#_{f \to 1}(S_j^1) = |\{x \in S_j^1 : f(x) = 1\}|$
      $\#_{f \to 1}(S_j^0) = |\{x \in S_j^0 : f(x) = 1\}|$
  end
  for $f \in \mathcal{F}_j$, $b \in \{0,1\}$ **do**
    Define the quality function $q(S_j, (f,b)) = -\#_{f \to 1}(S_j^{1-b})$
  end
  Let $(f_j, b_j) \leftarrow \mathcal{M}_E(S_j, q, \varepsilon)$
  Let $h_j \leftarrow (f_j, b_j)$, $S_{j+1} = \{x \in S_j | f_j(x) = 0\}$, and $\mathcal{F}_{j+1} = \mathcal{F}_j - \{f_j\}$
end
Return the hypothesis $h_{fin} = [h_1, h_2, ..., h_M]$

The success of the iterative selection procedure depends on the answer to the following question: Is there always a candidate function within $\mathcal{F}_j$ capable of accurately learning the remaining samples, especially as the subsets S_j become progressively smaller?

We show that this is the case in Proposition 6, which confirms that decision lists can be learned "consistently" by a covering algorithm – any remaining subset of the sample S labeled by a known decision list c^* can indeed be perfectly learned by a decision list comprised of the candidate functions. At the end of iteration M, the algorithm will have exhausted all available functions and all examples will have been classified.

Proposition 6 *Let $S = \{(x_i, \sigma_i)\}_{i=1}^n \in (\{0,1\}^d \times \{0,1\})^n$ be a sample that is labeled by a decision list c^* in the general form of $[(f_1, b_1), (f_2, b_2), ..., (f_m, b_m), (T, b_{m+1})]$ where $m \leq M$. For every iteration of running the* DP-GreedyCover *over S, there exists a decision list c_j over the remaining set of candidate functions $\mathcal{F}_j$ that is consistent with the remaining sample S_j.*

Proof It suffices to show that by induction over iteration j of DP-GreedyCover's execution over S, there exists a c_j in $\mathcal{F}_j$ which is consistent with the sample S_j.

For $j = 1$, we set $c_j = c^*$. For $j \geq 2$, suppose that there exists a c_j over $\mathcal{F}_j$ that is consistent with the sample S_j and fix an arbitrary $f_z \in \mathcal{F}_j$. Consider the following case analysis:

1. Assume f_z does not appear in c_j. Then we may take $c_{j+1} = c_j$. To see this, since $S'_j \subseteq S_j$ and c_j is a decision list over $\mathcal{F}_j \backslash \{f_z\}$ that is consistent with S_j, c_j is also consistent with S'_j.
2. Assume f_z appears in $c_j = [(f_1, b_1), (f_2, b_2), ..., (f_z, b_z), ..., (f_m, b_m), (T, b_{m+1})]$. Then we may take $c_{j+1} = [(f_1, b_1), ...(f_{z-1}, b_{z-1}), (f_{z+1}, b_{z+1}), ..., (T, b_{m+1})]$, which is c_j without the term (f_z, b_z).

Observe that the subset S_j that has been classified up to rule f_{z-1} is the same as the subset S'_j that has been classified up to function f_{z-1}. Since $S_j = S'_j$, for the terms (f_{z+1}, b_{z+1}) and after, the residual subset S_j is classified in the same pattern as S'_j. ■

3.2. Algorithm Analysis

Theorem 7 summarizes the privacy and accuracy guarantees of DP-GreedyCover. The complete proof is provided in Appendix A.

Theorem 7 *Let $M = |\mathcal{F}|$. There exists a (ε, δ)-DP (α, β)-PAC learner for the concept class $\mathcal{C}$ of $\mathcal{F}$-decision lists with sample complexity*

$$n \geq \max\left(\frac{64}{\alpha}(VC(\mathcal{C})\log(\frac{64}{\alpha}) + \log(\frac{16}{\beta})), \frac{8M\log(\frac{2M}{\sqrt{\beta}})(2\log(\frac{1}{\delta}) + \frac{3}{2})}{\alpha\varepsilon}\right)$$

and runtime $\mathrm{poly}(M)$.

Note that the VC dimension of $\mathcal{C}$ is at most the logarithm of its cardinality, which is also $\tilde{O}(M)$.

Our algorithm makes essential use of realizability. Even non-privately, the prospect of extending these techniques to the agnostic setting faces fundamental computational barriers Feldman et al. (2009). Specifically, efficient agnostic learning of even disjunctions would imply a breakthrough result for PAC learning poly-size DNF, suggesting very different techniques will be needed.

4. Differentially Private Winnow for Online Learning Halfspaces

Recall that we model online learning as a sequential T-round game between a learner and an adversary. In this paper, we focus on *oblivious* adversaries, which must choose a sequence of labeled examples $(x_1, y_1), \ldots, (x_T, y_T)$ in advance, but presents them to the learner in an online fashion. In each round t of the game, the adversary reveals unlabeled example x_t, the learner responds with a hypothesis $h_t : \mathcal{X} \to \{-1, 1\}$, and then the adversary reveals the label y_t.

Definition 8 (Differentially Private Learning Against an Oblivious Adversary) *Let S denote a sequence of labeled examples of the form $(x_1, y_1), \ldots, (x_T, y_T)$. Denote by $M(S)$ the sequence $h_1, \ldots, h_T$ of hypotheses produced by an online learner M playing the above game against an adversary choosing the sequence S. We say that learner M is (ε, δ)-differentially private if, for every pair S, S' differing in a single example and every set R of sequences of hypotheses, we have* $\Pr[M(S) \in R] \leq e^{\varepsilon} \Pr[M(S') \in R] + \delta$.

4.1. ConfidentWinnow

In this section, we present ConfidentWinnow, a non-private online learning algorithm for learning large-margin halfspaces. This algorithm serves as the foundational subroutine for its private analog, detailed in Section 4.2. The pseudocode of the algorithm is given in Algorithm 2. Unlike the traditional Winnow Algorithm, ConfidentWinnow is guaranteed not to update its weight vector only when it predicts both correctly and confidently.

Algorithm 2: ConfidentWinnow

Input: Time horizon T, confidence parameter c, margin parameter ρ
Output: Sequence of weight vectors $w^{(1)}, \ldots, w^{(T)}$ that represent halfspaces

Initialize: $w^{(0)} = (\frac{1}{d}, \ldots, \frac{1}{d})$
for $t = 1, 2, \ldots, T$ **do**
    $w^{(t)} \leftarrow w^{(t-1)}$
    Adversary presents example $x^{(t)}$
    Learner outputs $w^{(t)}$
    Adversary reveals $y^{(t)}$ and chooses bit $b^{(t)} \in \{0, 1\}$
    if $\mathrm{sgn}(\langle w^{(t)}, x^{(t)} \rangle) \neq y^{(t)}$ *or* ($-c\rho < \langle w^{(t)}, x^{(t)} \rangle < c\rho$ *and* $b^{(t)} = 1$) **then**
        // Perform multiplicative weight update on $w^{(t)}$ using $(x^{(t)}, y^{(t)})$
        for $j = 1$ *to* d **do**
            $w_j^{(t)} \leftarrow \frac{\exp(\eta y^{(t)} x_j^{(t)})}{\sum_{k=1}^{d} w_k^{(t-1)} \exp(\eta y^{(t)} x_k^{(t)})} \cdot w_j^{(t-1)}$
        end
    end
end

Lemma 9 below summarizes the guarantees of ConfidentWinnow. Our analysis closely follows the standard analysis of Winnow as presented by Mohri et al. (2018) and appears in Appendix B.

Lemma 9 *Let $x^{(1)}, \ldots, x^{(T)} \in \{-1, 1\}^d$ be an arbitrary sequence of T points and $b^{(1)}, \ldots, b^{(T)}$ and arbitrary sequence of update instructions. Assume that there exists a weight vector $v \in \mathbb{R}^d$*

where $v \geq 0$ and $\|v\|_1 = 1$ and a margin $\rho \geq 0$ such that for every $t \in [T]$, $y^{(t)}(v \cdot x^{(t)}) \geq \rho$. Then, when run with learning rate η and confidence ratio $c < 1/2$, the number of updates made by ConfidentWinnow *is at most $\frac{\log d}{(1-c)\eta\rho - \eta^2/2}$.*

4.2. DP-Winnow

In this section, we describe DP-Winnow (Algorithm 3), our main algorithm for learning large-margin halfspaces in the realizable setting with an oblivious adversary.

Sparse Vector Technique. Our differentially private Winnow algorithm relies on the sparse vector technique (Dwork and Roth, 2014). The technique rests on an interactive algorithm called AboveThreshold that, given a threshold parameter L and a sequence of sensitivity-1 queries $q^{(1)}, q^{(2)}, \ldots, q^{(T)}$ presented in a online fashion, approximately determines the first query whose value exceeds L. Following Asi et al. (2023b), we describe the interface to AboveThreshold via the following three functions:

1. InitializeAboveThr(ε, L, β): Initialize a new instance of AboveThreshold with privacy parameter ε, threshold L, and failure probability β. This returns a data structure Q supporting the following two functions.
2. Q.AddQuery(q): Adds a new sensitivity-1 query $q : \mathcal{Z}^n \to \mathbb{R}$ to Q.
3. Q.TestAboveThr$()$: Test if the most recent query added to Q was (approximately) above the threshold L. If that is the case, the data structure ceases to accept further queries.

Lemma 10 *(Dwork and Roth, 2014) There is an ε-differentially private interactive algorithm* AboveThreshold *with the interface described above. For every threshold parameter L and failure probability β, and every dataset $D \in \mathcal{Z}^n$, with probability at least $1 - \beta$, the following holds. Let $q^{(1)}, \ldots, q^{(T)}$ be the sequence of queries added to Q, and let k be the index of the first query for which* Q.TestAboveThr$()$ *would return* True *(or $T + 1$ if no such query exists). Then for $\alpha = \frac{8\log(2T/\beta)}{\varepsilon}$,*

1. *For all $t < k$, we have $q^{(t)}(D) \leq L + \alpha$, and*
2. *Either $q^{(k)}(D) \geq L - \alpha$ or $k = T + 1$.*

We leverage the sparse vector technique to efficiently and privately decide update rounds. In classical Winnow and ConfidentWinnow, updates only occur on rounds in which a prediction error occurred or if the prediction is "unconfident." By using AboveThreshold to conditionally trigger updates when errors exceed a noisy threshold, we ensure the privacy cost scales with the number of updates $(\mathrm{polylog}(T))$ rather than the total number of rounds T.

Algorithm 3: DP-Winnow

Input: Time horizon T, margin parameter ρ, privacy parameter $\hat{\varepsilon}$, learning rate η, threshold parameter L, normalization parameter m, failure probability β, switching bound K, Examples $(x^{(1)}, y^{(1)}), \ldots, (x^{(T)}, y^{(T)})$

Output: Sequence of weight vectors $\tilde{w}^{(1)}, \ldots, \tilde{w}^{(T)}$ representing halfspaces

Initialize: $k \leftarrow 0$, $t_p \leftarrow 0$, $C \leftarrow \emptyset$, $w^{(1)} \leftarrow \left(\frac{1}{d}, \frac{1}{d}, \ldots, \frac{1}{d}\right)$, $\tilde{w}^{(1)} \leftarrow \left(\frac{1}{d}, \frac{1}{d}, \ldots, \frac{1}{d}\right)$
$Q \leftarrow \mathsf{InitializeAboveThr}(\hat{\varepsilon}, L, \beta/T)$
for $t = 1, 2, \ldots, T$ **do**
  Receive unlabeled example $x^{(t)}$
  Output hypothesis $\tilde{w}^{(t)}$
  Receive label $y^{(t)}$
  if $\mathrm{sgn}(\langle \tilde{w}^{(t)}, x^{(t)} \rangle) \neq y^{(t)}$ **then**
    $C \leftarrow C \cup \{(x^{(t)}, y^{(t)})\}$
  end
  Define query $q^{(t)} \leftarrow \sum_{z=t_p}^{t} 1[\mathrm{sgn}(\langle \tilde{w}^{(z)}, x^{(z)} \rangle) \neq y^{(z)}]$
  $Q.\mathsf{AddQuery}(q^{(t)})$
  if $Q.\mathsf{TestAboveThr}()$ *and* $k < K$ **then**
    Let $(\hat{x}, \hat{y})$ be the first example cached in C
    // Perform multiplicative weight update on $w^{(t)}$ using $(\hat{x}, \hat{y})$
    for $j = 1, 2, \ldots, d$ **do**
      $w_j^{(t)} \leftarrow \frac{\exp(\eta \hat{y} \hat{x}_j)}{\sum_{k=1}^{d} w_k^{(t-1)} \exp(\eta \hat{y} \hat{x}_k)} \cdot w_j^{(t-1)}$
    end
    // Update approximate weight vector $\tilde{w}^{(t)}$ by sampling from $w^{(t)}$
    $j_1, \ldots, j_m \leftarrow [d]$ i.i.d. where for each i, $\Pr[j_i = j] = w_j^{(t)}$
    for $j = 1, 2, \ldots, d$ **do**
      $\tilde{w}_j^{(t)} \leftarrow \frac{1}{m} \cdot \#\{i : j_i = j\}$
    end
    Update $k \leftarrow k + 1$, $C \leftarrow \emptyset$, $t_p \leftarrow t + 1$
    $Q \leftarrow \mathsf{InitializeAboveThr}(\hat{\varepsilon}, L, \beta/T)$
  end
end

4.2.1. PRIVACY ANALYSIS

Theorem 11 *Let $\varepsilon, \delta \in (0,1)$. If $\hat{\varepsilon} = \varepsilon/(4\sqrt{2K\log(2/\delta)})$ and $\eta = \varepsilon/(8\sqrt{2mK\log(2/\delta)})$, then algorithm* DP-Winnow *is (ε, δ)-differentially private.*

Our privacy proof appears in Appendix C.

4.2.2. REGRET ANALYSIS

In this section, we summarize the performance of DP-Winnow (Algorithm 3) via Theorem 12.

Theorem 12 *When invoked using parameters $K = \frac{2\log d}{\eta\rho - \eta^2}$, $\hat{\varepsilon} = \varepsilon/(4\sqrt{2K\log(2/\delta)})$, $\eta = \varepsilon/(8\sqrt{2mK\log(4K/\delta)})$, $L = \frac{8\log(2T/\beta)}{\hat{\varepsilon}}$ and $m = 2\log(2T/\beta)/\rho^2$, Algorithm* DP-Winnow *is (ε, δ)-differentially private and makes at most*

$$\tilde{O}\left(\frac{\log^3 d \cdot \log^{5/2}(T/\beta) \cdot \log^2(1/\delta)}{\varepsilon^4\rho^6}\right)$$

prediction errors with probability at least $1 - 2\beta$.

We analyze the regret bound stated in Theorem 12 as follows.

1. **Regret Bound of** ConfidentWinnow: We previously showed that ConfidentWinnow learns halfspaces with finite mistake bound as detailed in Lemma 9.

2. **Consistency of Weight Vectors**: We demonstrate in Lemma 13 that, with high probability, the shadow weight vectors and approximate weight vectors used by ConfidentWinnow make equivalent predictions when the predictions are confident.

3. **Equivalence of Regret and Update Bounds**: We show that the number of mistakes made by ConfidentWinnow is asymptotically equivalent to DP-Winnow's update bound in Lemma 14, bridging the gap between the non-private and private learning algorithms.

4. **Cost Efficiency of the Sparse Vector Technique**: By applying the properties of the Sparse Vector Technique described in Lemma 10, we establish that DP-Winnow, with high probability, incurs a polylogarithmic number of mistakes per update.

Lemma 13 demonstrates that with high probability, the predictions made by the shadow weight vectors $w^{(t)}$ and the approximate weight vectors $\tilde{w}^{(t)}$ are consistent across all rounds $t \in [T]$.

Lemma 13 *Let $x \in \{-1, 1\}^d$ be an unlabeled example and let w be a weight vector with $\|w\|_1 = 1$ such that $|\langle w, x\rangle| \geq c\rho$. Construct $\tilde{w}$ according to the following process:*

- *Define a probability distribution over the index set $[d]$ where each coordinate j is sampled with probability w_j.*

- *For each $i \in \{1, \ldots, m\}$, independently sample indices j_i from this distribution.*

- *Set $\tilde{w}_j$ for each $j \in [d]$ to $\frac{1}{m}$ multiplied by the count of j_i's equal to j.*

Let $\beta > 0$ and $m \geq \frac{2}{c^2\rho^2}\log(\frac{2T}{\beta})$. Then, with probability at least $1 - \beta/T$,

$$\operatorname{sgn}(\langle \tilde{w}, x\rangle) = \operatorname{sgn}(\langle w, x\rangle).$$

The proof appears in Appendix D.

Lemma 14 below shows that that the upper bound on the number of updates required by ConfidentWinnow is asymptotically equivalent to the switching bound of DP-Winnow.

Lemma 14 *For any (obliviously chosen) sequence of examples consistent with a ρ-margin halfspace, with probability at least* $1-\beta$*, algorithm* DP-Winnow *performs fewer than* $\log d/((1-c)\eta\rho - \eta^2/2)$ *updates. Thus, if we set* $K = \log d/((1-c)\eta\rho - \eta^2/2)$*, the condition "*$k < K$*" in line 12 will never be violated.*

Proof

Fix a sequence of examples consistent with a ρ-margin halfspace. We compare a run of DP-Winnow (denoted here by $\mathcal{A}_1$) with a run of ConfidentWinnow (denoted here by $\mathcal{A}_2$) on the subsequence of examples on which $\mathcal{A}_1$ performs an update. Simplifying our notation to index these examples, let us denote them by $(x^{(1)}, y^{(1)}), \ldots, (x^{(B)}, y^{(B)})$.

By construction, these are only candidates for updates because each $\mathrm{sgn}(\langle \tilde{w}^{(t)}, x^{(t)} \rangle) \neq y^{(t)}$. By Lemma 13, with probability at least $1-\beta$, we must have $\langle w^{(t)}, x^{(t)} \rangle \cdot y^{(t)} \leq c\rho$ for all of these candidates. Thus, if $\mathcal{A}_2$ is instructed to update its weights when presented with each of these examples, it maintains the same sequence of weights $w^{(1)}, \ldots, w^{(B)}$ as $\mathcal{A}_1$. By Lemma 9, the number of such updates is upper bounded by the stated quantity. ■

Proof [Proof of Theorem 12]

By Lemma 14, with probability $1-\beta$, Algorithm DP-Winnow performs at most $K = \frac{\log d}{(1-c)\eta\rho - \eta^2/2}$ updates, and only incurs prediction errors between those updates. The number of such prediction errors between updates is governed by the accuracy of AboveThreshold which, by Lemma 10, guarantees at most $\frac{16\log(2T^2/\beta)}{\hat{\varepsilon}}$ prediction errors per update with total probability $1-\beta$. Hence, we have that with probability $1-2\beta$, the total number of prediction errors is at most $\frac{16\log(2T^2/\beta)}{\hat{\varepsilon}} \cdot K$.

By Theorem 11, it suffices to take $\hat{\varepsilon} = \varepsilon/(4\sqrt{2K\log(2/\delta)})$ and $\eta = \varepsilon/(8\sqrt{2mK\log(4K/\delta)})$ to guarantee (ε, δ)-differential privacy overall. Fix $c = 1/2$ and recall that we are taking $m = 2\log(2T/\beta)/c^2\rho^2$, so $\eta < \rho/2$. Hence, $K \leq \frac{4\log d}{\eta\rho}$. Plugging in our setting of η gives

$$K \leq \tilde{O}\left(\frac{(\log^2 d) m \log(1/\delta)}{\varepsilon^2\rho^2}\right).$$

Thus, with probability at least $1-2\beta$, the total number of prediction errors is at most

$$\tilde{O}\left(\frac{\log^3 d \cdot \log^{5/2}(T/\beta) \cdot \log^2(1/\delta)}{\varepsilon^4\rho^6}\right).$$

■

While our analysis makes essential use of the realizability assumption, as does the classic Winnow algorithm, an interesting direction for future work would be to extend it to learn a drifting target as studied, e.g., by Blum (1998).

4.3. Application: Privately Learning Decision Lists

In this section, we explain how our Algorithm 3 for learning large-margin halfspaces yields a private algorithm for learning decision lists. This is based on a classic connection observed in work of Blum and Singh (1990), Dhagat and Hellerstein (1994), Valiant (1999), and Nevo and El-Yaniv (2003).

In general, if a decision list has length r and its output bits alternate between $+1$ and -1 at most D times, then it is represented by a halfspace with large margin.

Lemma 15 *(Blum and Singh, 1990; Dhagat and Hellerstein, 1994; Valiant, 1999) Every monotone 1-decision list over d variables with length r and D alternations can be represented by a halfspace over $d+1$ variables with margin $\Omega(1/r^D)$.*

However, the weight vector for the resulting halfspace might have negative entries. For our DP-Winnow algorithm to apply, we apply a simple transformation to convert a halfspace over d dimensions (with possibly negative entries in its weight vector) into one over $2d$ dimensions $z_1, \ldots, z_{2d}$ with only nonnegative entries by setting

$$w'_i = \begin{cases} |w_i| & \text{if } i \le d, w_i \ge 0 \\ |w_{i-d}| & \text{if } i > d, w_i < 0 \\ 0 & \text{otherwise.} \end{cases} \qquad z_i = \begin{cases} x_i & \text{if } i \le d \\ -x_{i-d} & \text{otherwise.} \end{cases}$$

Note that the resulting halfspace has the same margin as the original halfspace.

Theorem 16 *Algorithm* DP-Winnow *learns the class of length-r monotone 1-decision lists with D alternations while making at most*

$$\tilde{O}\left(\frac{r^{6D} \cdot \log^3 d \cdot \log^{5/2}(T/\beta) \cdot \log^2(1/\delta)}{\varepsilon^4}\right)$$

prediction errors with probability at least $1 - O(\beta)$.

In general, if instead of taking each ℓ_i to be a variable, one takes it to be a member of some feature space $\mathcal{F}$, then we can use the same algorithm to obtain

Theorem 17 *Algorithm* DP-Winnow *learns the class of length-r $\mathcal{F}$-decision lists with D alternations while making at most*

$$\tilde{O}\left(\frac{r^{6D} \cdot \log^3 |\mathcal{F}| \cdot \log^{5/2}(T/\beta) \cdot \log^2(1/\delta)}{\varepsilon^4}\right)$$

prediction errors with probability at least $1 - O(\beta)$.

4.4. Lower Bound

Let $\mathcal{P} = \{p_t : [T] \to \{-1, 1\}\}$ be the class of point functions over a domain $[T]$ defined by $p_t(x) = 1 \iff x = t$. Cohen et al. (2024) showed that every $(\varepsilon = 1/2, \delta = 1/(20 \log T))$-differentially private online learner for $\mathcal{P}$ incurs at least $\Omega(\log T)$ mistakes on some sequence of T examples.

We now argue that this implies a lower bound of $\Omega(\min\{\log d, \log T\})$ for privately learning halfspaces with margin $\rho = 1$. In particular, this implies that for $T \ll d$, a polylogarithmic dependence on T as in the statement of Theorem 12 is necessary. To see this, observe that we can embed point functions into the class of margin-1 halfspaces in dimension $d = T$ by mapping each domain point $x \in [T]$ to the point $(1, 1, \ldots, 1, -1, 1, \ldots, 1)$, where the -1 appears in the x'th coordinate, and mapping each function p_t to the halfspace whose weight vector is the t'th standard basis vector $(0, 0, \ldots, 0, 1, 0, \ldots 0)$.

Acknowledgments

MB was supported by NSF CNS-2046425. WF thanks Debanuj Nayak and Satchit Sivakumar for helpful discussions.

References

Naman Agarwal and Karan Singh. The price of differential privacy for online learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 32–40. PMLR, 2017. URL http://proceedings.mlr.press/v70/agarwal17a.html.

Hilal Asi, Vitaly Feldman, Tomer Koren, and Kunal Talwar. Private online prediction from experts: Separations and faster rates. In Gergely Neu and Lorenzo Rosasco, editors, *The Thirty Sixth Annual Conference on Learning Theory, COLT 2023, 12-15 July 2023, Bangalore, India*, volume 195 of *Proceedings of Machine Learning Research*, pages 674–699. PMLR, 2023a.

Hilal Asi, Vitaly Feldman, Tomer Koren, and Kunal Talwar. Near-optimal algorithms for private online optimization in the realizable regime, 2023b.

Avrim Blum. *On-Line Algorithms in Machine Learning*, pages 306–325. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. ISBN 978-3-540-68311-7. doi: 10.1007/BFb0029575. URL https://doi.org/10.1007/BFb0029575.

Avrim Blum and Mona Singh. Learning functions of k terms. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, COLT '90, page 144–153, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc. ISBN 1558601465.

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the vapnik-chervonenkis dimension. *J. ACM*, 36(4):929–965, 1989. doi: 10.1145/76359.76371. URL https://doi.org/10.1145/76359.76371.

Mark Bun, Marco Leandro Carmosino, and Jessica Sorrell. Efficient, noise-tolerant, and private learning via boosting. In Jacob Abernethy and Shivani Agarwal, editors, *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 1031–1077. PMLR, 09–12 Jul 2020. URL https://proceedings.mlr.press/v125/bun20a.html.

Edith Cohen, Xin Lyu, Jelani Nelson, Tamás Sarlós, and Uri Stemmer. Lower bounds for differential privacy under continual observation and online threshold queries. In Shipra Agrawal and Aaron Roth, editors, *Proceedings of Thirty Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pages 1200–1222. PMLR, 30 Jun–03 Jul 2024. URL https://proceedings.mlr.press/v247/cohen24b.html.

Aditi Dhagat and Lisa Hellerstein. Pac learning with irrelevant attributes. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 64–74. IEEE, 1994.

Daniil Dmitriev, Kristóf Szabó, and Amartya Sanyal. On the growth of mistakes in differentially private online learning: A lower bound perspective. In Shipra Agrawal and Aaron Roth, editors, *Proceedings of Thirty Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pages 1379–1398. PMLR, 30 Jun–03 Jul 2024. URL https://proceedings.mlr.press/v247/dmitriev24a.html.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, aug 2014. ISSN 1551-305X. doi: 10.1561/0400000042. URL https://doi.org/10.1561/0400000042.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC'06, page 265–284, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3540327312. doi: 10.1007/11681878_14. URL https://doi.org/10.1007/11681878_14.

Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, STOC '10, page 715–724, New York, NY, USA, 2010a. Association for Computing Machinery. ISBN 9781450300506. doi: 10.1145/1806689.1806787. URL https://doi.org/10.1145/1806689.1806787.

Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60, 2010b. doi: 10.1109/FOCS.2010.12.

Vitaly Feldman, Venkatesan Guruswami, Prasad Raghavendra, and Yi Wu. Agnostic learning of monomials by halfspaces is hard. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 385–394, 2009. doi: 10.1109/FOCS.2009.26.

Marco Gaboardi, Emilio Jesus Gallego Arias, Justin Hsu, Aaron Roth, and Zhiwei Steven Wu. Dual query: Practical private query release for high dimensional data. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1170–1178, Bejing, China, 22–24 Jun 2014a. PMLR. URL https://proceedings.mlr.press/v32/gaboardi14.html.

Marco Gaboardi, Emilio Jesus Gallego Arias, Justin Hsu, Aaron Roth, and Zhiwei Steven Wu. Dual query: Practical private query release for high dimensional data. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1170–1178, Bejing, China, 22–24 Jun 2014b. PMLR. URL https://proceedings.mlr.press/v32/gaboardi14.html.

Badih Ghazi, Cristóbal Guzmán, Pritish Kamath, Alexander Knop, Ravi Kumar, Pasin Manurangsi, and Sushant Sachdeva. Prem: Privately answering statistical queries with relative error, 02 2025.

Noah Golowich and Roi Livni. Littlestone classes are privately online learnable. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc. ISBN 9781713845393.

Anupam Gupta, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. Differentially private combinatorial optimization, 2009.

Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 61–70, 2010. doi: 10.1109/FOCS.2010.85.

Moritz Hardt, Katrina Ligett, and Frank Mcsherry. A simple and practical algorithm for differentially private data release. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/208e43f0e45c4c78cafadb83d2888cb6-Paper.pdf.

Justin Hsu, Aaron Roth, and Jonathan Ullman. Differential privacy for the analyst via private equilibrium computation. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '13, page 341–350, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450320290. doi: 10.1145/2488608.2488651. URL https://doi.org/10.1145/2488608.2488651.

Prateek Jain and Abhradeep Guha Thakurta. (near) dimension independent risk bounds for differentially private learning. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 476–484. JMLR.org, 2014. URL http://proceedings.mlr.press/v32/jain14.html.

Prateek Jain, Pravesh Kothari, and Abhradeep Thakurta. Differentially private online learning. In Shie Mannor, Nathan Srebro, and Robert C. Williamson, editors, *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland*, volume 23 of *JMLR Proceedings*, pages 24.1–24.34. JMLR.org, 2012.

Peter Kairouz, Brendan Mcmahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and private (deep) learning without sampling or shuffling. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5213–5225. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/kairouz21b.html.

Haim Kaplan, Yishay Mansour, Yossi Matias, and Uri Stemmer. Differentially private learning of geometric concepts. *CoRR*, abs/1902.05017, 2019. URL http://arxiv.org/abs/1902.05017.

Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 531–540, 2008. doi: 10.1109/FOCS.2008.27.

Bo Li, Wei Wang, and Peng Ye. Private online learning against an adaptive adversary: Realizable and agnostic settings. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a. URL https://openreview.net/forum?id=gcWJyQlFlz.

Bo Li, Wei Wang, and Peng Ye. The limits of differential privacy in online learning, 2025b. URL https://arxiv.org/abs/2411.05483.

Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Mach. Learn.*, 2(4):285–318, April 1988. ISSN 0885-6125. doi: 10.1023/A:1022869011914. URL https://doi.org/10.1023/A:1022869011914.

Xin Lyu. Private Learning of Littlestone Classes, Revisited. *arXiv e-prints*, art. arXiv:2510.00076, September 2025. doi: 10.48550/arXiv.2510.00076.

Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103, 2007. doi: 10.1109/FOCS.2007.66.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning*. The MIT Press, 2nd edition, 2018. ISBN 0262039400.

Ziv Nevo and Ran El-Yaniv. On online learning of decision lists. *J. Mach. Learn. Res.*, 3(null): 271–301, March 2003. ISSN 1532-4435. doi: 10.1162/153244303765208395. URL https://doi.org/10.1162/153244303765208395.

Huy Lê Nguyễn, Jonathan Ullman, and Lydia Zakynthinou. Efficient Private Algorithms for Learning Large-Margin Halfspaces. In Aryeh Kontorovich and Gergely Neu, editors, *Proceedings of the 31st International Conference on Algorithmic Learning Theory*, volume 117 of *Proceedings of Machine Learning Research*, pages 704–724. PMLR, 08 Feb–11 Feb 2020. URL https://proceedings.mlr.press/v117/nguy-en20a.html.

Ronald L. Rivest. Learning decision lists. *Machine Learning*, 2(3):229–246, 1987. doi: 10.1007/BF00058680.

Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistic Surveys*, 16:1–85, 2022.

Adam D. Smith and Abhradeep Guha Thakurta. (nearly) optimal algorithms for private online learning in full-information and bandit settings. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*, pages 2733–2741, 2013.

Salil Vadhan and Wanrong Zhang. Concurrent composition theorems for differential privacy. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, page 507–519, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450399135. doi: 10.1145/3564246.3585241. URL https://doi.org/10.1145/3564246.3585241.

Salil P. Vadhan and Tianhao Wang. Concurrent composition of differential privacy. In Kobbi Nissim and Brent Waters, editors, *Theory of Cryptography - 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8-11, 2021, Proceedings, Part II*, volume 13043 of *Lecture Notes in*

Computer Science, pages 582–604. Springer, 2021. doi: 10.1007/978-3-030-90453-1_20. URL https://doi.org/10.1007/978-3-030-90453-1_20.

Leslie G. Valiant. Projection learning. *Machine Learning*, 37(2):115–130, jan 1999. doi: 10.1023/a:1007678005361. URL https://doi.org/10.1023/a:1007678005361.

Appendix A. Proof of Theorem 7

Our analysis underlying Theorem 7 goes by way of the following roadmap.

1. EMCover **Framework**: We introduce EMCover, a generic learning framework that DP-GreedyCover instantiates. EMCover employs the Exponential Mechanism to iteratively select hypotheses, with each selection dependent on previous rounds. We state the precise privacy guarantees in Lemma 18. The pseudocode of EMCover is shown in Algorithm 4.

2. **Martingale Argument for Privacy Analysis**: To prove the privacy claim established in Lemma 18, we invoke the martingale argument of Lemma 19 (detailed in Gupta et al. (2009); Kaplan et al. (2019)).

3. **Empirical Error of** DP-GreedyCover: In Lemma 20, we show that DP-GreedyCover does not incur too much error with respect to its given sample.

4. **Empirical Error to Generalization Error**: Theorem 21 shows that when given a sample of size above the VC dimension, our algorithm's small empirical error translates into small generalization error.

A.1. Privacy Analysis

We consider the following general framework for covering algorithms based on the Exponential Mechanism. This framework abstracts and generalizes a presentation of Kaplan et al. (2019), who studied a similar algorithm for learning sparse disjunctions and conjunctions. Let Z be a data universe, let H be a set of candidates, and let Q be a family of score functions of the form $q : Z^* \times H \to \mathbb{R}$. For each $z \in Z$, let H_z be the subset of H of candidates that "covers" data point z. We are interested in EM-based cover algorithms that take the following form:

Algorithm 4: EMCover

Input: Dataset $S \in Z^*$, privacy parameter ε

Output: List of functions $h_1, \dots, h_T$

Initialize: Let $\mathcal{F}_1 = \mathcal{F}$ together with the constant true

for $t = 1, \dots, T$ **do**

  Select a score function q_t from Q (arbitrarily, and possibly adaptively, based on the outcomes of previous iterations)

  Invoke the Exponential Mechanism to obtain $h_t \leftarrow \mathcal{M}_E(q_t, S, \varepsilon)$

  Update S by removing all elements z that are covered by h_t

end

When Q is structured, this algorithm gives privacy guarantees that are much stronger than what one might expect based on standard composition theorems.

Lemma 18 *If the family of quality functions Q has the property that for every $q \in Q$, every $h \in H$, and every $z \in Z$:*

1. *If h covers z, then for every dataset S, we have $q(S,h) - 1 \le q(S \cup \{z\}, h) \le q(S,h) + 1$, and*
2. *If h does not cover z, then for every dataset S, we have $q(S \cup \{z\}, h) = q(S,h)$,*

then for every $\delta > 0$, algorithm EMCover *is $(2\varepsilon(\ln(1/\delta) + \frac{3}{2}), \delta)$-differentially private.*

Proof Let S be any dataset and let $S' = S \cup \{z\}$ for some $z \in Z$. Fix a sequence of candidate outputs $h_1, \dots, h_T$. Let t be the smallest index for which h_t covers z (or $t = T$ otherwise).

Case (a)

$$\begin{aligned}
\frac{\Pr[\mathsf{EMCover}(S) = (h_1, \dots, h_T)]}{\Pr[\mathsf{EMCover}(S') = (h_1, \dots, h_T)]} &= \frac{e^{\varepsilon q_t(S,h_t))}}{e^{\varepsilon q_t(S',h_t))}} \cdot \prod_{i=1}^{t} \frac{\sum_{h \in H} e^{\varepsilon q_i(S',h)}}{\sum_{h \in H} e^{\varepsilon q_i(S,h)}} \\
&\le e^{\varepsilon} \cdot \prod_{i=1}^{t} \frac{\sum_{h \in H_z} e^{\varepsilon} \cdot e^{\varepsilon q_i(S,h)} + \sum_{h \notin H_z} e^{\varepsilon q_i(S,h)}}{\sum_{h \in H} e^{\varepsilon q_i(S,h)}} \\
&= e^{\varepsilon} \cdot \prod_{i=1}^{t} (1 + (e^{\varepsilon} - 1) p_i(S;z)) \\
&\le e^{\varepsilon} \cdot \exp\left(\sum_{i=1}^{t} 2\varepsilon p_i(S;z) \right).
\end{aligned}$$

The first equality uses the facts that i) after candidate h_t is realized, the datasets consisting of uncovered elements in the two executions are identical, and ii) before the realization of h_t, by condition (2), scores for both datasets are the same. The first inequality is an immediate result of condition (1). Next, let $p_i(S;z)$ be the probability that the next candidate drawn covers z when the dataset is S, conditioned on realizing $h_1, \dots, h_{i-1}$ in the previous iterations. Then the second inequality stems from the fact that for every $\varepsilon > 0$ and $x \in [0,1]$, $1 + (e^{\varepsilon} - 1)x \le e^{2\varepsilon x}$.

For some dataset S and an example z, we call an output $\vec{h} = (h_1, h_2, ..., h_T)$ λ-bad if $\sum_{i=1}^{T} p_i(S;z) \cdot \mathbb{1}[\forall j \le i, h_j \text{ does not cover } z] > \lambda$. Otherwise, the output $\vec{h}$ is λ-good. We first consider the case when the output $\vec{h}$ is $\ln(\frac{1}{\delta})$-good. Then

$$\sum_{i=1}^{t-1} p_i(S;z) \le \sum_{i=1}^{T} p_i(S;z) \cdot \mathbb{1}[\forall j \le i, h_j \text{ does not cover } z] \le \ln(\frac{1}{\delta}).$$

Thus, for every $\ln(\frac{1}{\delta})$-good output h, we have

$$\begin{aligned}
\frac{\Pr[\mathsf{EMCover}(S) = (h_1, \dots, h_T)]}{\Pr[\mathsf{EMCover}(S') = (h_1, \dots, h_T)]} &\le e^{\varepsilon} \cdot e^{2\varepsilon \cdot (\ln(\frac{1}{\delta}) + p_t(S;z))} \\
&\le e^{\varepsilon} \cdot e^{2\varepsilon \cdot (\ln(\frac{1}{\delta}) + 1)} \\
&\le e^{2\varepsilon \cdot (\ln(\frac{1}{\delta}) + \frac{3}{2})}.
\end{aligned}$$

Next, by invoking a martingale argument of Gupta et al. (2009), we show that the probability that EMSetCover outputs a $\ln(\frac{1}{\delta})$-bad output is bounded by at most δ.

Lemma 19 *(Gupta et al., 2009) Consider the T-round random process where an adversary chooses $p_i \in [0,1]$ in each round possibly adaptively based on the previous $i-1$ rounds, and a coin is tossed with probability p_i of realizing heads. Let N_i be the indicator for the event that no heads are realized within the first i steps. Defining the random variable $Y = \sum_{i=1}^{T} p_i N_i$, we have $\Pr[Y > y] \leq \exp(-y)$ for every $y > 0$.*

We map the setting of Lemma 19 to the execution of EMCover as follows. When selecting a candidate h_i in round i, the algorithm first tosses a coin with heads probability of $p_i(S,z)$ to decide whether to pick a candidate that covers data point z or not. Then, EMCover uses a second source of randomness to determine the output h_i, sampling it according to the right conditional probability based on the coin's outcome.

Therefore, for all sets of outputs R,

$$\begin{aligned}
\Pr[\mathsf{EMCover}(S) \in R] &= \sum_{\vec{h} \in R} \Pr[\mathsf{EMCover}(S) = \vec{h}] \\
&= \sum_{\substack{\vec{h} \in R, \\ \ln(\frac{1}{\delta})-\text{good for } S}} \Pr[\mathsf{EMCover}(S) = \vec{h}] + \sum_{\substack{\vec{h} \in R, \\ \ln(\frac{1}{\delta})-\text{bad for } S}} \Pr[\mathsf{EMCover}(S) = \vec{h}] \\
&\leq \sum_{\vec{h} \in R,\ \ln(\frac{1}{\delta})-\text{good for } S} e^{2\varepsilon \cdot (\ln(\frac{1}{\delta}) + \frac{3}{2})} \Pr[\mathsf{EMCover}(S') = \vec{h}] + \delta \\
&\leq e^{2\varepsilon \cdot (\ln(\frac{1}{\delta}) + \frac{3}{2})} \Pr[\mathsf{EMCover}(S') \in R] + \delta.
\end{aligned}$$

Studying the opposite ratio proceeds similarly.

Case (b)

$$\begin{aligned}
\frac{\Pr[\mathsf{EMCover}(S') = (h_1, \ldots, h_T)]}{\Pr[\mathsf{EMCover}(S) = (h_1, \ldots, h_T)]} &= \frac{e^{\varepsilon q_t(S', h_t))}}{e^{\varepsilon q_t(S, h_t))}} \cdot \prod_{i=1}^{t} \frac{\sum_{h \in H} e^{\varepsilon q_i(S,h)}}{\sum_{h \in H} e^{\varepsilon q_i(S',h)}} \\
&\leq e^{\varepsilon} \cdot \prod_{i=1}^{t} \frac{\sum_{h \in H_z} e^{\varepsilon} \cdot e^{\varepsilon q_i(S',h)} + \sum_{h \notin H_z} e^{\varepsilon q_i(S',h)}}{\sum_{h \in H} e^{\varepsilon q_i(S',h)}} \\
&= e^{\varepsilon} \cdot \prod_{i=1}^{t} (1 + (e^{\varepsilon} - 1) p_i(S'; z)) \\
&\leq e^{\varepsilon} \cdot e^{\sum_{i=1}^{t} 2\varepsilon p_i(S'; z)}
\end{aligned}$$

Similar to Case (a), the first equality uses the facts that i) after candidate h_t is realized, the datasets of uncovered elements in the two executions are identical, and ii) before the realization of h_t, by condition (2), scores for both datasets are the same. The first inequality is derived from the double inequality for quality scores from (1). Then, we denote the probability that the next candidate drawn covers z when the dataset is S' conditioned on realizing $h_1, \ldots, h_{i-1}$ in the previous iterations as $p_i(S'; z)$. Again using the fact that for every $\varepsilon > 0$ and $x \in [0,1]$, $1 + (e^{\varepsilon} - 1)x \leq e^{2\varepsilon x}$, we obtain the second inequality.

Next, we perform an analogous case analysis on whether the output $\vec{h}$ is $\ln(\frac{1}{\delta})$-good and bad.

If $\vec{h} = (h_1, \dots, h_T)$ is $\ln(\frac{1}{\delta})$-good, we have

$$\begin{aligned}\frac{\Pr[\mathsf{EMCover}(S') = (h_1, \dots, h_T)]}{\Pr[\mathsf{EMCover}(S) = (h_1, \dots, h_T)]} &\leq e^{\varepsilon} \cdot e^{2\varepsilon \cdot (\ln(\frac{1}{\delta}) + p_t(S';z))} \\ &\leq e^{\varepsilon} \cdot e^{2\varepsilon \cdot (\ln(\frac{1}{\delta}) + 1])} \\ &\leq e^{2\varepsilon \cdot (\ln(\frac{1}{\delta}) + \frac{3}{2})}\end{aligned}$$

Otherwise, using Lemma 19, we have $\sum_{\vec{h} \in R,\ \ln(\frac{1}{\delta}) - \text{bad for } S'} \Pr[\mathsf{EMCover}(S') = \vec{h}] \leq \delta$. Thus, for every set of outcomes R,

$$\begin{aligned}\sum_{\vec{h} \in R} \Pr[\mathsf{EMCover}(S') = \vec{h}] &= \sum_{\vec{h} \in R,\ \ln(\frac{1}{\delta}) - \text{good for } S'} \Pr[\mathsf{EMCover}(S') = \vec{h}] \\ &\quad + \sum_{\vec{h} \in R,\ \ln(\frac{1}{\delta}) - \text{bad for } S'} \Pr[\mathsf{EMCover}(S') = \vec{h}] \\ &\quad \sum_{\vec{h} \in R,\ \ln(\frac{1}{\delta}) - \text{good for } S'} e^{2\varepsilon \cdot (\ln(\frac{1}{\delta}) + \frac{3}{2})} \Pr[\mathsf{EMCover}(S) = \vec{h}] + \delta \\ &\leq e^{2\varepsilon \cdot (\ln(\frac{1}{\delta}) + \frac{3}{2})} \Pr[\mathsf{EMCover}(S) \in R] + \delta.\end{aligned}$$

■

A.2. Utility Analysis

Lemma 20 describes the empirical error bound for DP-GreedyCover.

Lemma 20 *Fix a target function $c^* \in \mathcal{F}$-decision lists with feature set size of M, and consider the execution of* DP-GreedyCover *on a sample $S = \{(x_i, \sigma_i)\}_{i=1}^n$. Assume $\beta > 0$, and for every iteration j. Then, with probability at least $1 - \beta$, the algorithm errs on at most $\frac{4M}{\varepsilon}(\log(\sqrt{\frac{2}{\beta}}M))$ examples in S.*

Proof By Proposition 6, every iteration maintains the consistency between the remaining sample S and some decision list with a feature set of size $M - 1$. This implies every iteration j there is a pair of Boolean function $f \in \mathcal{F}_j$ and Boolean b such that either $\#_{f \to 1}(S^1) = 0$ or $\#_{f \to 1}(S^0) = 0$. By Proposition 3, for every iteration j, the mechanism outputs a pair (f_j, b_j) that satisfies for all $t \geq 0$,

$$\Pr[q(S_j, (f_j, b_j)) \geq \max_{(f,b) \in \mathcal{F}_j \times \{0,1\}} \{q(S_j, (f, b))\} - \frac{2}{\varepsilon}(\log(|\mathcal{F}_j \times \{0, 1\}|) + t)] \leq e^{-t}.$$

Let $t = \log(\frac{M}{\beta})$. Then, by union-bounding over the M iterations and taking the complement, we have

$$
\begin{aligned}
&\Pr\left[\bigcap_{j=1}^{M}\left(q(S_j,(f_j,b_j)) \geq \max_{(f,b)\in\mathcal{F}_j\times\{0,1\}} q(S_j,(f,b)) - \frac{2}{\varepsilon}\left(\log\Big(|\mathcal{F}_j\times\{0,1\}|\Big) + \log\Big(\frac{M}{\beta}\Big)\right)\right)\right] \\
&\quad \geq 1 - M\, e^{-\log\left(\frac{M}{\beta}\right)} \\
&\quad = 1-\beta.
\end{aligned}
$$

Furthermore, since there exists a pair $(f,b)\in\mathcal{F}_j\times\{0,1\}$ such that either $\#_{f\to 1}(S^1)=0$ or $\#_{f\to 1}(S^0)=0$, we have

$$
\Pr[\bigcap_{j=1}^{M} -q(S_j,(f_j,b_j)) \leq \frac{2}{\varepsilon}(\log(|\mathcal{F}_j\times\{0,1\}|)+\log(\frac{M}{\beta}))] \geq 1-\beta
$$

$$
\implies \Pr[\bigcap_{j=1}^{M} \underset{f_j\to 1}{\#}(S_j^{1-b_j}) \leq \frac{4}{\varepsilon}(\log(\sqrt{\frac{2}{\beta}}M))] \geq 1-\beta
$$

In every iteration j, the selected pair of Boolean function and Boolean (f_j,b_j) misclassifies at most $\frac{4}{\varepsilon}(\log(\sqrt{\frac{2}{\beta}}M))$ examples from S_j with probability at least $1-\beta$. Furthermore, since the algorithm exhausts all candidate functions by the end of iteration M, all examples from S have been classified. Therefore, h_{fin} errs on at most $\frac{4M}{\varepsilon}(\log(\sqrt{\frac{2}{\beta}}M))$ examples of S with probability at least $1-\beta$. ■

Having established that our learner enjoys low empirical error with respect to its sample, we next quote the VC dimension generalization bound we need to translate this into low population error.

Theorem 21 *(VC-Dimension Generalization Bound (Blumer et al., 1989)) Let $\mathcal{D}$ and $\mathcal{C}$ be a distribution and concept class over domain $\mathcal{X}$ respectively, and let $c\in\mathcal{C}$. For sample $S=\{(x_i,c(x_i))\}_{i=1}^n$ of size $n\geq\frac{64}{\alpha}(VC(\mathcal{C})\log(\frac{64}{\alpha})+\log(\frac{16}{\beta}))$ where x_i are chosen i.i.d from $\mathcal{D}$, it holds that*

$$
\Pr_{S\sim\mathcal{D}}[\exists h\in\mathcal{C} \text{ s.t. } err_{\mathcal{D}}(h,c)>\alpha, err_S(h)\leq\frac{\alpha}{2}]\leq\beta
$$

Proof [Proof of Theorem 7]

Let algorithm A denote DP-GreedyCover with the Exponential Mechanism instantiated with $\hat{\varepsilon}=\frac{\varepsilon}{2(\log(\frac{1}{\delta})+\frac{3}{2})}$. By Claim 18, A is (ε,δ)-DP.

Assume that the number of samples n is at least $\frac{64}{\alpha}(VC(\mathcal{C})\log(\frac{64}{\alpha})+\log(\frac{16}{\beta}))$. By Lemma 20 DP-GreedyCover obtains generalization error at most $\frac{8M\log(\frac{2M}{\sqrt{\beta}})}{\varepsilon n}$ with probability at least $1-\beta$. Then, we compute the sample complexity bound

$$
n\geq\frac{8M\log(\frac{2M}{\sqrt{\beta}})}{\hat{\varepsilon}\alpha}=\frac{8M\log(\frac{2M}{\sqrt{\beta}})(2\log(\frac{1}{\delta})+\frac{3}{2})}{\alpha\varepsilon}.
$$

■

A.3. Application of DP-GreedyCover to Learning k-Decision Lists

In this section, we illustrate using DP-GreedyCover to learn the class of k-decision lists, where each term is a conjunction of at most k literals over d Boolean variables. We denote the set of conjunctions of at most k literals over d Boolean variables as $\mathcal{C}_d^k$.

Corollary 22 *There exists a (ε, δ)-DP (α, β) PAC learner for the concept class k-decision lists with sample complexity*

$$n = O\left(\frac{d^k \log(d^k/\beta) \cdot \log(\frac{1}{\delta})}{\alpha\varepsilon}\right)$$

To see this, it suffices to show that the number of conjunctions with at exactly k literals over d Boolean variables is upper bounded by $e^2 d^k$ in Proposition 23.

Proposition 23 $|\mathcal{C}_d^k| \leq e^2 d^k$.

Proof

$$\begin{aligned}
|\mathcal{C}_d^k| &= \sum_{j=0}^{k} \# \text{ conjunctions of width exactly} j \\
&= \sum_{j=0}^{k} \# \text{ ways to choose } j \text{ variables} \times \# \text{ ways to choose sign pattern} \\
&= \sum_{j=0}^{k} 2^j \binom{d}{j} \\
&\leq \sum_{j=0}^{k} \frac{2^j}{j!} (d) \cdot (d-1) \cdot \ldots \cdot (d-k+1) \\
&\leq d^k \sum_{j=0}^{k} \frac{2^j}{j!} \leq d^k \sum_{j=0}^{\infty} \frac{2^j}{j!} \leq e^2 d^k.
\end{aligned}$$

■

Appendix B. Proof of Lemma 9

Proof For every $t \in [T]$, define the potential function $\phi^{(t)}$ to be the relative entropy between the components of the normalized target vector v and the components of the hypothesis vector $w^{(t)}$:

$$\phi^{(t)} = \sum_{i=1}^{d} v_i \log\left(\frac{v_i}{w_i^{(t)}}\right).$$

Intuitively, this captures how close the hypothesis weight vector $w^{(t)}$ is to the target weight vector v.

Suppose the algorithm performs an update in round $t+1$. Let us compute the difference between $\phi^{(t+1)}$ and $\phi^{(t)}$, i.e, how much the potential decreases as a result of this update.

$$
\begin{aligned}
\phi^{(t+1)} - \phi^{(t)} &= \sum_{i=1}^{d} v_i \left(\log\left(\frac{v_i}{w_i^{(t+1)}}\right) - \log\left(\frac{v_i}{w_i^{(t)}}\right) \right) \\
&= \sum_{i=1}^{d} v_i \log\left(\frac{w_i^{(t)}}{w_i^{(t+1)}}\right) \\
&= \sum_{i=1}^{d} v_i \log\left(\frac{\sum_{k=1}^{d} w_k^{(t)} e^{\eta y^{(t)} x_k^{(t)}}}{e^{\eta y^{(t)} x_i^{(t)}}}\right) \\
&= \sum_{i=1}^{d} v_i \log\left(\sum_{k=1}^{d} w_k^{(t)} e^{\eta y^{(t)} x_k^{(t)}}\right) + \sum_{i=1}^{d} v_i \log\left(\frac{1}{e^{\eta y^{(t)} x_i^{(t)}}}\right) \\
&= \log\left(\sum_{k=1}^{d} w_k^{(t)} e^{\eta y^{(t)} x_k^{(t)}}\right) - \sum_{i=1}^{d} v_i \eta y^{(t)} x_i^{(t)} \\
&\le \log\left(\sum_{k=1}^{d} w_k^{(t)} e^{\eta y^{(t)} x_k^{(t)}}\right) - \eta\rho \\
&= \log\left(\mathbb{E}_{k\sim w^{(t)}}[e^{\eta y^{(t)} x_k^{(t)}}]\right) - \eta\rho \\
&= \log\left(\mathbb{E}_{k\sim w^{(t)}}[\exp(\eta y^{(t)} x_k^{(t)} - \eta y^{(t)}\langle w^{(t)}, x^{(t)}\rangle + \eta y^{(t)}\langle w^{(t)}, x^{(t)}\rangle)]\right) - \eta\rho \\
&\le \log\left(\mathbb{E}_{k\sim w^{(t)}}[\exp(\eta^2/2)]\right) + \eta y^{(t)}\langle w^{(t)}, x^{(t)}\rangle - \eta\rho \\
&\le \frac{\eta^2}{2} + \eta\rho(c-1).
\end{aligned}
$$

The first inequality uses the margin assumption that $\sum_{i=1}^{d} y^{(t)} v_i x_i^{(t)} \ge \rho$. The following two inequalities are derived by expressing the sum as an expectation over the distribution defined by $w^{(t)}$. The second inequality follows by applying Hoeffding's lemma to the random variable $x_k^{(t)} \in \{-1, 1\}$ where $k \sim w^{(t)}$:

$$
\begin{aligned}
\mathbb{E}_{k\sim w^{(t)}}[\exp(\eta y^{(t)}(x_k^{(t)} - \langle w^{(t)}, x^{(t)}\rangle))] &= \mathbb{E}_{k\sim w^{(t)}}[\exp(\eta y^{(t)}(x_k^{(t)} - \mathbb{E}[x_k^{(t)}]))] \\
&= \exp\left(\frac{\eta^2 \cdot (1-(-1))^2}{8}\right) = \exp\left(\frac{\eta^2}{2}\right).
\end{aligned}
$$

Finally, the last inequality uses the fact that updates may only occur when the prediction is unconfident or incorrect, i.e., when $y^{(t)}\langle w^{(t)}, x^{(t)}\rangle \le c\rho$.

Let M be the total number of updates the algorithm performs. Then summing up over all of the rounds in which an update occurs, we have

$$
\phi_{T+1} - \phi_1 \le M\left(\frac{\eta^2}{2} + \eta\rho(c-1)\right).
$$

To derive an upper bound on M, observe that

1. $\phi_{T+1} \ge 0$ since the relative entropy between two distributions is always nonnegative, and

2. $\phi_1 = \sum_{i=1}^{d} v_i \log(\frac{v_i}{1/d}) \leq \log d$.

By combining these inequalities, we obtain the stated upper bound on M.

$$-\log d \leq M(\frac{\eta^2}{2} + \eta\rho(c-1))$$
$$\implies M \leq \frac{\log d}{(1-c)\eta\rho - \frac{\eta^2}{2}}.$$

■

Appendix C. Proof of Theorem 11

Our DP-Winnow algorithm is perhaps most naturally viewed as a concurrent composition between two interactive differentially private algorithms, described at a high-level as follows.

View 1: DP-Winnow as a concurrent composition. Run the following two algorithms concurrently:

1. Sequentially compose at most K invocations of the AboveThreshold algorithm to reveal the rounds in which an update to the weight vector should occur.
2. Maintain the weight vector w, and whenever AboveThreshold triggers an update, perform it and release m independent samples from the Exponential Mechanism.

This view of the algorithm allows one to use concurrent composition theorems for differential privacy (Vadhan and Wang, 2021; Vadhan and Zhang, 2023) to reason about its privacy guarantees. However, it is possible to give a simpler analysis by studying the following equivalent algorithm that just post-processes a sequential composition of AboveThreshold instances with the Exponential Mechanism.

View 2: Recompute weights from history Consider the following algorithm whose output distribution is identically distributed to that of Algorithm 3, but does so with the following changes:

1. At the end of each round t, it discards its current weights $w^{(t)}$.
2. If in a round t, the above-threshold check passes in line 15, it recomputes $w^{(t)}$ from scratch by rerunning the same multiplicative weight updates on the prefix $(x^{(1)}, y^{(1)}), \ldots, (x^{(t)}, y^{(t)})$ that DP-Winnow would have performed given the history of released approximate weight vectors $\tilde{w}^{(1)}, \ldots, \tilde{w}^{(t-1)}$.
3. It releases $\tilde{w}^{(t)}$ either by sampling from $w^{(t)}$ (i.e., using the Exponential Mechanism) or by keeping $\tilde{w}^{(t-1)}$, as appropriate.

Proof

Since this alternative implementation of the algorithm does not maintain private state between AboveThreshold and Exponential Mechanism invocations, it can be viewed as the sequential composition of at most K applications of AboveThreshold, each $(\hat{\varepsilon}, 0)$-DP, and mK Exponential Mechanism invocations, each (2η)-DP.

To see the latter, let us focus on the computation performed by DP-Winnow between lines 13 and 22, and fix a history of the algorithm's outputs $\tilde{w}^{(1)}, \ldots, \tilde{w}^{(t)}$ and updates. We claim that each sample j_i in line 19 corresponds to an invocation of the Exponential Mechanism on a sensitivity-1 score function with parameter η. To see this, observe that we can unroll the numerator of each entry w_j of the weight vector as $\exp(\sum_{i \in R} \eta \hat{y}^{(i)} \hat{x}_j^{(i)})$ where R is the set of indices of examples which were used to perform updates. For any fixed history of weights and updates, and for any pair of neighboring example sequences, there is at most one index in R and at most one corresponding term in the sum that differ. Thus, drawing a single sample j_i is (2η)-differentially private.

Let ε_{AT} and ε_{EM} be the privacy loss contributed by AboveThreshold and Exponential Mechanism calls. Applying the advanced composition theorem (Dwork and Roth, 2014) with failure probability split evenly between the two layers ($\delta/2$ each), one obtains

1. $\varepsilon_{AT} \leq \hat{\varepsilon}\sqrt{2K \log(2/\delta)} + K\hat{\varepsilon}^2$, and
2. $\varepsilon_{EM} \leq 2\eta\sqrt{2mK \log(2/\delta)} + 4mK\eta^2$.

Setting

$$\hat{\varepsilon} = \varepsilon/(4\sqrt{2K \log(2/\delta)}), \qquad \eta = \varepsilon/(8\sqrt{2mK \log(2/\delta)})$$

yields

$$\varepsilon_{AT} + \varepsilon_{EM} = \hat{\varepsilon}\sqrt{2K \log(2/\delta)} + 2\eta\sqrt{2mK \log(2/\delta)} + K\hat{\varepsilon}^2 + 4mK\eta^2$$
$$\leq \varepsilon/4 + \varepsilon/4 + \varepsilon^2/(16 \log(2/\delta)) < \varepsilon/2 + \varepsilon/8 < \varepsilon$$

for all $\varepsilon, \delta \in (0, 1)$ and gives the stated bound. ■

Appendix D. Proof of Lemma 13

Proof Fix a time step $t \in [T]$. Our goal is to show that if w is any shadow weight vector with $\|w\|_1 = 1$ and $x \in \{-1, 1\}^d$, then

$$\langle w, x\rangle \geq c\rho \implies \langle \tilde{w}, x\rangle > 0$$
$$\langle w, x\rangle \leq -c\rho \implies \langle \tilde{w}, x\rangle < 0$$

with probability at least $1 - \beta/T$ over the sampling of $\tilde{w}$.

For each $i \in [m]$, let J_i be the i'th index sampled. That is, for each index $j \in [d]$, we have $\Pr[J_i = j] = w_j$. For each $i \in [m]$, define the random variable $\chi_i = x_{J_i}$, which is ± 1-valued with $\mathbb{E}[\chi_i] = \sum_{j=1}^{d} x_j \Pr[J_i = j] = \langle w, x\rangle$.

For each $j \in [d]$, let $\tilde{w}_j = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}[J_i = j]$ be the fraction of times j is sampled, and let $\tilde{w} = (\tilde{w}_1, \ldots, \tilde{w}_d)$. Then

$$
\begin{aligned}
\langle \tilde{w}, x \rangle &= \frac{1}{m} \sum_{j=1}^{d} x_j \sum_{i=1}^{m} \mathbf{1}[J_i = j] \\
&= \frac{1}{m} \sum_{i=1}^{m} x_{J_i} \\
&= \frac{1}{m} \sum_{i=1}^{m} \chi_i.
\end{aligned}
$$

Thus, we can estimate the probability of the bad event where $\langle \tilde{w}, x \rangle$ disagrees in sign with $\langle w, x \rangle$ as

$$
\begin{aligned}
\Pr[|\langle \tilde{w}, x \rangle - \langle w, x \rangle| > |\langle w, x \rangle|] &\le \Pr[|\langle \tilde{w}, x \rangle - \langle w, x \rangle| > c\rho] \\
&= \Pr\left[\left|\frac{1}{m}\left(\sum_{i=1}^{m} \chi_i - \sum_{i=1}^{m} \mathbb{E}[\chi_i]\right)\right| > c\rho\right] \\
&= \Pr\left[\left|\sum_{i=1}^{m} \chi_i - \sum_{i=1}^{m} \mathbb{E}[\chi_i]\right| > mc\rho\right] \\
&\le 2\exp\left(-\frac{mc^2\rho^2}{2}\right) \\
&\le \frac{\beta}{T}
\end{aligned}
$$

The first inequality uses our confidence assumption that $|\langle w, x \rangle| \ge c\rho$. The second inequality follows from Hoeffding's inequality. We derive the last inequality by setting $m \ge \frac{2}{c^2\rho^2} \log(\frac{2T}{\beta})$. ■