Yale University CPSC 683
Stability in Machine Learning: Generalization, Privacy & Replicability

Lecture 6: Private PAC Learning Implies Online Learning

Instructor: Alkis Kalavasis, alkis.kalavasis@yale.edu

Acknowledgement Thanks to Argyris Mouzakis for improving the first version of this draft.

1 Which concept classes can we learn privately?

In Lecture 5, we focused on pure differential privacy. Here we relax the notion of privacy.

Definition 1: Approximate Differential Privacy

A randomized algorithm A is (ϵ, δ)-differentially private if for all neighboring databases S, S', and for all sets Y of outputs of A, it holds that

$$\Pr_{h \sim A(S)}[h \in Y] \leq \exp(\epsilon) \cdot \Pr_{h \sim A(S')}[h \in Y] + \delta\,.$$

The probability is taken over the internal randomness (i.e., the random coins) of A.

We are interested in understanding which concept classes are PAC learnable with (ϵ, δ)-DP constraints.[1] We will show a surprising result, shown by Alon, Livni, Malliaris and Moran in Alon et al. (2019):

(ϵ, δ)-Private PAC learning implies finite Littlestone dimension.

2 Learning Thresholds with Privacy Constraints

Learning thresholds is one of the most basic problems in machine learning. This problem consists of an unknown threshold function $c : \mathbb{R} \to \{-, +\}$, an unknown distribution $\mathcal{D}$ over $\mathbb{R}$, and the goal is to output a hypothesis $h : \mathbb{R} \to \{-, +\}$ that is close to c (in terms of classification loss), given access to a labeled examples $(x_1, c(x_1)), ..., (x_m, c(x_m))$, where the x_i's are drawn independently from $\mathcal{D}$.

Standard PAC learning of thresholds without privacy constraints is known to be easy and can be done using a constant number of examples (since the VC dimension is 1). In contrast, whether thresholds can be learned privately turned out to be more challenging:

- Based on Lecture 5, the result of Kasiviswanathan et al. (2011) implies a pure differentially private algorithm that learns thresholds over a finite $X \subseteq \mathbb{R}$ of size n with $O(\log n)$ examples. Feldman and Xiao (2014) showed a matching lower bound for any pure differentially private algorithm. Beimel et al. (2013) showed that by relaxing the privacy constraint to approximate differential privacy, one can significantly improve the upper bound to $2^{O(\log^* n)}$ samples. Bun et al. (2015) gave a lower bound of $\Omega(\log^* n)$ that applies for any proper learning algorithm.

[1] Using standard boosting arguments, we can focus on the regime $\epsilon = 0.1$ and $\delta = o(1/m)$ for sample size m.

Exercise 1

Read the algorithm of Beimel et al. (2013) for singletons and thresholds based on quasi-concave promise problems.

3 Some Deep Connections: Thresholds and Littlestone Classes

It turns out that there is an close connection between thresholds and the Littlestone dimension, which goes back to Shelah (1972):

A class $\mathcal{H}$ has a finite Littlestone dimension if and only if it does not embed thresholds as a subclass.

More formally, Shelah's result (which is part of the so-called Unstable Formula Theorem) (see (Malliaris and Moran, 2022) for a more modern reference) provides a simple and elegant connection between Littlestone dimension and th class of thresholds:

Theorem 1: Littlestone Dimension and Thresholds (Shelah, 1972)

Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class. The following hold true:

- If $\mathrm{Ldim}(\mathcal{H}) \geq d$, then $\mathcal{H}$ contains $\lfloor \log d \rfloor$ thresholds.
- If $\mathcal{H}$ contains d thresholds then $\mathrm{Ldim}(\mathcal{H}) \geq \lfloor \log d \rfloor$.

The second part of the statement is relatively easy. If $\mathcal{H}$ contains 2^t thresholds then there are $h_i \in \mathcal{H}$ for $0 \leq i < 2^t$ and there are x_j for $0 \leq j < 2^t - 1$ such that $h_i(x_j) = 0$ for $j < i$ and $h_i(x_j) = 1$ for $j \geq i$. Now we can define a labeled binary tree of height t corresponding to the binary search process that will witness the Littlestone dimension bound. For the first part, we refer to Alon et al. (2019) for a combinatorial proof.

In this Lecture, we will discuss two theorems. We focus on the standard DP regime where $\epsilon = 0.1$ and $\delta = o(1/m)$, where m is the number of samples.[2]

Theorem 2: Thresholds are not privately learnable (Alon et al., 2019)

Let $\mathcal{X} \subseteq \mathbb{R}$ with $|\mathcal{X}| = n$ and let $\mathcal{A}$ be an $(1/16, 1/16)$-accurate learning algorithm for the class of thresholds over $\mathcal{X}$ with sample complexity m which satisfies (ϵ, δ)-differential privacy with $\epsilon = 0.1$ and $\delta = O(\frac{1}{m^2 \log m})$. Then

$$m \geq \Omega(\log^* n)\,.$$

In particular, the class of thresholds over an infinite $\mathcal{X}$ cannot be learned privately.

The above is a strong negative result showing that thresholds over infinite domains are not privately PAC learnable (even in constant accuracy).

Definition 2: Iterated Logarithm

We define $\log^{(0)} x = \log x$ and $\log^{(m)} x = 1 + \log^{(m-1)} \log x$. The function $\log^* x$ equals the number of times the iterated logarithm must be applied before the result is less than or equal to 1. We define

$$\log^* x = (1 + \log^* \log x)\mathbf{1}\{x > 1\}\,.$$

[2]The studied privacy regime is the standard one appearing in practice. A natural interpretation of the negligible $\delta > 0$ parameter is that there is a very small probability of a very bad event (in which even all the input data is leaked) but otherwise the algorithm satisfies pure differential privacy.

The function $\log^* x$ equals the number of times the iterated logarithm must be applied before the result is less than or equal to 1.

Combining the above result with the above deep connection between thresholds and Littlestone dimension, we get the following:

Theorem 3: Private Learning implies finite Littlestone dimension (Alon et al., 2019)

Let $\mathcal{H}$ be a hypothesis class with Littlestone dimension $d \in \mathbb{N} \cup \{\infty\}$. Let $\mathcal{A}$ be an $(1/16, 1/16)$-accurate learning algorithm for $\mathcal{H}$ with sample complexity m which satisfies (ϵ, δ)-differential privacy with $\epsilon = 0.1$ and $\delta = O(\frac{1}{m^2 \log m})$. Then

$$m \geq \Omega(\log^* d)\,.$$

In particular, any class that is privately learnable has a finite Littlestone dimension.

To see this, $\mathcal{H}$ contains $c = \lfloor \log d \rfloor$ thresholds, i.e., there exist $x_1, ..., x_c$ and $h_1, ..., h_c \in \mathcal{H}$ such that $h_i(x_j) = 1$ for all $j \geq i$. But then, Theorem 2 implies a lower bound of $m \geq \Omega(\log^* c) = \Omega(\log^* d)$.

4 Ramsey Theory and Sketch of the Reduction

It suffices to prove Theorem 2. We begin by considering an arbitrary differentially private algorithm A that learns the class of thresholds over an ordered domain $\mathcal{X}$ of size n. Our goal is to show a lower bound of order $\Omega(\log^* n)$ on the sample complexity of A.

Before the work of Alon et al. (2019), we already knew of lower bounds for proper algorithms. The central challenge in the proof is the potential *improper* nature of the learner A, i.e., A may output arbitrary hypotheses (which is in contrast with proving impossibility results for proper algorithms where the structure of the learned class can be exploited).

Some rough ideas about the proof are as follows:

1. Construct a collection of hard datasets (which are algorithm-dependent)
2. These datasets will exploit some structure of the algorithm, making use of the fact that it learns thresholds.

The core property that the hard datasets will have is the following: they will come from a critical subset of the domain on which the learner behaves *in a highly regular way*. How do we find some a regular structure?

Insight 1: Ramsey Theory

Ramsey theory is a branch of the mathematical field of combinatorics that focuses on the appearance of **order in a substructure given a structure of a known size**. We use Ramsey theory to deal with the fact that the algorithm is improper by finding some structure in its behavior.

Ramsey theory handles the above challenge by showing that for any algorithm (in fact, for any mapping that takes input samples to output hypotheses) there is a large subset of the domain that is *homogeneous with respect to the algorithm.* This notion of homogeneity places useful restrictions on the algorithm when restricting it to the homogeneous set.

We will need the following definition.

Definition 3: Increasing and Balanced Dataset

A sample $S = (x_1, y_1), ..., (x_m, y_m)$ of an even size is realizable (by thresholds), balanced, and increasing if and only if $x_1 < x_2 < ... < x_m$ and the first half of y_i's is -1 and the second one $+1$.

Reduction to homogeneous set. As discussed above, the first step in the proof is about identifying a large homogeneous subset of the input domain $\mathcal{X}$ on which we can control the output of A:

Definition 4: Homogeneous Set

A subset $\mathcal{X}' \subseteq \mathcal{X}$ is called m-homogeneous with respect to A if there is a list of numbers $p_0, p_1, ..., p_m$ such that for every increasing balanced sample $S \in (\mathcal{X}' \times \{-1, 1\})^m$ of points from $\mathcal{X}'$ and every $x' \in \mathcal{X}'$ with $\text{ord}_S(x') = i$:

$$|A_S(x') - p_i| = O(1/m)\,.$$

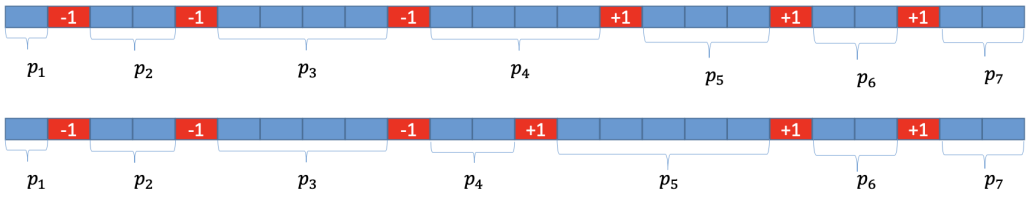

Figure 1: Depiction of two possible outputs of an algorithm over an homogeneous set, given two input samples from the set (marked in red). The number p_i denote, for a given point x, the probability that $h(x) = 1$, where $h \sim A(S)$ is the hypothesis h outputted by the algorithm on input sample S. These probabilities depends (up to a small additive error) only on the interval that x belongs to. In the figure above we changed in the input the fourth example – this only affects the interval and not the values of the p_i's (again, up to a small additive error).

Figure 1: Taken from Alon et al. (2019).

The intuition is the following: the probability for any x' in the homogeneous subset $\mathcal{X}' \setminus S$ to receive the label 1 by the algorithm A trained on S depends only on the relative ordering with respect to training examples S.

Existence of Large Homogeneous Sets. Do homogeneous sets exist? This is where Ramsey theory comes in.

Lemma 1: Every algorithm has a large homogeneous set

Let A be a (possibly randomized) algorithm that is defined over input samples of size m over a domain $\mathcal{X} \subseteq \mathbb{R}$ of size $|\mathcal{X}| = n$. Then, there is a set $\mathcal{X}' \subseteq \mathcal{X}$ that is m-homogeneous with respect to A of size

$$|\mathcal{X}'| \geq \frac{\log^{(m)} n}{2^{m \log m}}\,.$$

The intuition is the following: Consider the complete m-uniform hypergraph with n vertices. If n is appropriately large and we color the hyperedges with k colors, there must exist a large monochromatic hyperclique. Roughly speaking, in our case:

1. vertices are domain points.
2. m-uniform hyperedges are all datasets that are increasing and balanced.
3. colors are lists of probabilities.
4. monochromatic hyperclique is a homogeneous set.

Now Ramsey's Theorem implies that for any algorithm A there is a large subset $\mathcal{X}' \subseteq \mathcal{X}$ homogeneous with respect to A.

More typically, define a coloring on the $(m+1)$-subsets of $\mathcal{X}$ as follows. Let $D = \{x_1 < x_2 < ... < x_{m+1}\}$ be such a subset. For each $i \leq m+1$, define $D^{-i} = D \setminus \{x_i\}$ and let S^{-i} denote the balanced increasing sample on D^{-i}. Set p_i to be the element of $\{t/10m^2\}_t$ that is closest to $A_{S^{-i}}(x_i)$ ($10m^2+1$ choices). The coloring assigned is then the list $(p_1, p_2, ..., p_{m+1})$.

Theorem 4: Quantitative Version of Ramsey Theorem

Let $s > t \geq 2$ and q be integers and let

$$N \geq \text{twr}_t(3sq \log q)\,.$$

Then for every coloring of the subsets of size t of a universe of size N using q colors there is a homogeneous subset of size s (a subset of the universe is homogeneous if all of its t-subsets have the same color.).

Recall that $\text{twr}_1(x) = x$ and $\text{twr}_i(x) = 2^{\text{twr}_{i-1}(x)}$.

In the above, we have $q = (10m^2+1)^{m+1}$ colors, $t = m+1$ and $N = n$. Solving for s in the above, we can show that there is a set $\mathcal{X}' \subseteq X$ such that $|\mathcal{X}'| \geq \frac{\log^{(m)} n}{2^{m \log m}}$ such that all $m+1$-subsets have the same color.

Implications of Homogeneous Sets. We can now use the structure of these sets for the next result.

Theorem 5: Large homogeneous sets imply lower bounds for private learning

Let A be an $O(0.1, \delta)$-differentially private algorithm with sample complexity m and $\delta \leq \frac{1}{1000m^2 \log m}$.
Let $\mathcal{X} = \{1, ..., k\}$ be m-homogeneous with respect to A.
Then if A *empirically* learns thresholds over $\mathcal{X}$ with $(1/16, 1/16)$-accuracy, then

$$m \geq \frac{\sqrt{\log k}}{\log \log k}\,.$$

Using Lemma 5.9 in Bun et al. (2015), we can transfer the above sample complexity lower bound from empirical learners to population ones (with the same sample complexity lower bound).

Combining the above with Lemma 1, we get that $m \geq \Omega(\log^* n)$. Let us see why:

1. Lemma 1 implies that $k \geq \log^{(m)} n / 2^{O(m \log m)}$
2. Theorem 5 implies that $k = 2^{O(m^2 \log^2 m)}$.

Combining the two bounds, we get that

$$\log^{(m)} n \leq 2^{c \cdot m^2 \log m}\,.$$

Apply the iterated logarithm t times where $t = \log^*(2^{O(m^2 \log^2 m)}) = \log^*(m) + O(1)$ to the above inequality and get that

$$\log^{(m+t)} n = \log^{(m+\log^* m + O(1))} n \leq 1 .$$

This means that $\log^* n \leq m + \log^* m + O(1)$. This implies that $m \geq \log^* n$ and gives the desired lower bound on the sample complexity.

To show Theorem 5, we need to obtain a large class of hard datasets. Roughly speaking, if A is an accurate and private empirical learner for thresholds over a homogeneous set, its list of probabilities $p_1, ..., p_{m+1}$ should have a gap, i.e., there is some i such that $|p_i - p_{i-1}| \geq 1/m$.

We will use this property to construct a hard family of distributions. Let's give some intuition: Let this index be i^* and construct hard datasets as follows: Pick an increasing realizable sample $S \in (\mathcal{X}' \times \{0,1\})^m$ of size m so that the interval $J \subseteq \mathcal{X}'$ between x_{i^*-1} and x_{i^*+1} is of size $k - m$. For every $x \in J$, let S_x be the neighboring sample of S that is obtained by replacing x_i with x.

> **Lemma 2: Hard Family (Informal)**
>
> Let m, k as in Theorem 5. Let $A, \mathcal{X}'$ be as above. Let $n = m - k$. There exists a family $\{P_i : i \leq n\}$ over $\{-1, 1\}^n$ such that (i) every P_i, P_j are $(0.1, \delta)$-indistinguishable and (ii) there exists r such that for all i : $\mathbf{Pr}_{v \sim P_i}[v(j) = 1] = (r - 1/m)1\{j < i\} + (r + 1/m)1\{j \geq i\}$.

Then one can show that $k - m = |\text{size of family}| \leq 2^{m^2 \log^2 m}$, which implies that $k = 2^{O(m^2 \log^2 m)}$.

Conclusion The main result of this Lecture is a lower bound on the sample complexity of private learning in terms of the Littlestone dimension.

In the next Lecture, we will see whether the opposite is true: can every class with finite Littlestone dimension be learned privately?

References

Alon, N., Livni, R., Malliaris, M., and Moran, S. (2019). Private pac learning implies finite littlestone dimension. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 852–860.

Beimel, A., Nissim, K., and Stemmer, U. (2013). Private learning and sanitization: Pure vs. approximate differential privacy. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 363–378. Springer.

Bun, M., Nissim, K., Stemmer, U., and Vadhan, S. (2015). Differentially private release and learning of threshold functions. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 634–649. IEEE.

Feldman, V. and Xiao, D. (2014). Sample complexity bounds on differentially private learning via communication complexity. In *Conference on Learning Theory*, pages 1000–1019. PMLR.

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. (2011). What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826.

Malliaris, M. and Moran, S. (2022). The unstable formula theorem revisited via algorithms. *arXiv preprint arXiv:2212.05050*.

Shelah, S. (1972). A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261.