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

Lecture 7: Online Learning Implies Private PAC Learning

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

In the previous lecture, we proved that private PAC learning requires a finite Littlestone dimension. In Lecture 7, we will sketch the ideas for the breakthrough result of Bun, Livni and Moran (Bun et al., 2020), showing that finite Littlestone dimension is also sufficient for learning with differential privacy.

This establishes a qualitative equivalence between online learning and PAC learning with approximate differential privacy.

1 Differential Privacy, Online Learning, and Stability

At a first glance it may seem that online learning (which is characterized by the finiteness of the Littlestone dimension) and differentially private PAC learning have little to do with one another: e.g., in online learning there is no underlying distribution while in PAC learning, each learning task corresponds to some probability distribution over labeled examples.

However, a long line of works has revealed a tight connection between the two. For instance, in Lecture 6, we showed that online learnability is necessary for private PAC learning.

This connection was culminated in the work of Bun et al. (2020), where Bun, Livni, and Moran proved that every binary concept class with finite Littlestone dimension can be learned by an approximate differentially private algorithm. This result, together with the main result of the previous Lecture, demonstrate an equivalence between online and private classification.

At a high-level, this connection seems to boil down to the notion of *stability*, which plays a key role in both topics. On one hand, the definition of differential privacy is itself a form of stability; it requires that the distribution over outputs of the learning algorithm is "stable" with respect to small changes of the training set. On the other hand, stability also arises as a central motif in online learning paradigms, where an online learner eventually "stabilizes" to some hypothesis given feedback from the environment and stops making mistakes.

In their monograph, Dwork and Roth (Dwork et al., 2014) identified stability as a common factor of learning and differential privacy: "*Differential privacy is enabled by stability and ensures stability...we observe a tantalizing moral equivalence between learnability, differential privacy, and stability.*"

Essentially, the works of Alon et al. (2019) and Bun et al. (2020) formalize the above intuitive statement.

2 Privacy Reminder

Our main object of interest is approximate differential privacy which we recall below.

Definition 1: Approximate Differential Privacy

A randomized algorithm A is (ϵ, δ)-differentially private if for all neighboring datasets 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.

3 Online Learning Implies DP PAC Learning

The main result of Bun et al. (2015) is that Littlestone classes can be PAC learning with approximate differential privacy.

Theorem 1: Bun et al. (2020)

Let $\mathcal{H}$ be a class of Littlestone dimension d, let $\epsilon, \delta \in (0,1)$ be privacy parameters and let $\alpha, \beta \in (0, 1/2)$ be accuracy parameters. For

$$n = \frac{2^{\widetilde{O}(2^d)} + \log 1/\beta\delta}{\alpha\epsilon},$$

there exists an (ϵ, δ)-DP algorithm A such that for any realizable distribution $\mathcal{D}$, given as input $S \sim \mathcal{D}^n$ samples, the output hypothesis $f \sim A(S)$ satisfies $R_{\mathcal{D}}(f) \leq \alpha$ with probability $1 - \beta$, where the randomness is over S and A and $R_{\mathcal{D}}$ is the population error under $\mathcal{D}$.

Combining this result with the main Theorem of the previous lecture, one gets the following surprising equivalence.

Theorem 2: Alon et al. (2022)

For a class $\mathcal{H} \subseteq \{-1, 1\}^{\mathcal{X}}$, $\mathcal{H}$ is online learnable if and only if $\mathcal{H}$ is (ϵ, δ)-DP PAC learnable.

4 The Proof

The starting point of the proof is the following key definition.

Definition 2: Global Stability (Bun et al., 2020)

For $\eta > 0$ and $n \in \mathbb{N}$, a learning algorithm A is (n, η)-globally stable with respect to a distribution $\mathcal{D}$ over examples if there is a hypothesis h whose frequency as an output is at least η, i.e.,

$$\Pr_{S \sim \mathcal{D}^n}[h = A(S)] \geq \eta.$$

The word *global* highlights a difference with other forms of algorithmic stability. Indeed, previous forms of stability such as differential privacy (Dwork et al., 2014) and uniform stability (Bousquet and Elisseeff, 2002) are local in the sense that they require output robustness subject to local changes in the input. However, the property required by global stability captures stability with respect to resampling the entire input.

For instance, every learner using n examples which produces a hypothesis in a finite hypothesis class $\mathcal{H}$ is $(n, 1/|\mathcal{H}|)$-globally stable.

Having this definition at hand, it is a natural first step to ask: which concept classes can be learned by globally stable algorithms?

Proof Outline. The proof consists of two parts. First, we will see that every concept class with a finite Littlestone dimension can be learned by a globally stable algorithm. Next, we have to transform the globally stable learner to a DP one (using mostly standard tools from differential privacy).

4.1 Littlestone Classes can be Learned by Globally Stable Learners

Let $\mathcal{H}$ be a concept class with Littlestone dimension d. Global stability is quite strong of a notion. Our goal is to give an (n, η)-globally-stable learning algorithm for $\mathcal{H}$ with (very small) stability parameter $\eta = 2^{-2^{O(d)}}$ and (very large) sample complexity $n = 2^{2^{O(d)}}$.

Let $\mathcal{D}$ be the unknown realizable learning distribution. So, $\mathcal{D}$ is a distribution over examples $(x, c(x))$ where $c \in \mathcal{H}$ is an unknown target concept.

Since $d < \infty$, we know that $\mathcal{H}$ is online learnable in the realizable setting with at most d mistakes. Let A be the algorithm that online learns $\mathcal{H}$ with mistake bound d. Consider an execution of that online algorithm to the training set $(x_1, c(x_1)), ..., (x_n, c(x_n))$ drawn i.i.d. from $\mathcal{D}^n$.

This execution induces a random variable M, which corresponds to the number of mistakes that A makes. By definition of the algorithm (and online learning), we know that

$$0 \leq M \leq d.$$

Hence, we have that there exists $i \in \{0, 1, ..., d\}$ such that

$$\mathbf{Pr}[M = i] \geq \frac{1}{d+1}.$$

The first observation is that this index can be identified from samples: with high probability, an i with $\mathbf{Pr}[M = i] \geq 1/(2d)$ can be found by independently running A on $O(d)$ traces from $\mathcal{D}^n$.

This index i has $d + 1$ possible values. Let us first assume that $i = d$, i.e.,

$$\mathbf{Pr}[M = d] \geq \frac{1}{2d}.$$

This is a good scenario for our learning algorithm because, after making d mistakes, we have *identified* the true hypothesis c (since otherwise the adversary could force us make more mistakes). Hence, if $i = d$, then $A(S) = c$ with probability at least $1/(2d)$. In this case, the samples were very informative and we could identify the target hypothesis. What if $i < d$?

Let us assume that $i = d - 1$. To give some intuition, we plan to follow an inductive argument "over a Littlestone tree of depth d" and this is why we get double-exponential dependencies on the parameters.

The issue with $i = d - 1$ is that A can output many different hypotheses before making the d-th mistake. Hence, we need to find a way to make the algorithm stable: we should artificially force the learner make one more mistake.

Let us draw two samples $S_1, S_2 \sim \mathcal{D}^n$ independently and let $f_i = A(S_i)$. Consider the event that the number of mistakes made by A on each S_i is exactly $d - 1$. We know that this event occurs with probability $1/(2d)^2$. Let us condition on that event. There are two cases: either $\mathbf{Pr}[f_1 = f_2] < 1/4$ or $\geq 1/4$.

If the collision occurs with probability at least $1/4$, then this means that there exists an h such that

$$\mathbf{Pr}[A(S) = h] \geq \frac{1}{4} \cdot \frac{1}{(2d)^2}$$

and we are done. The difficult case is when the collision is rare. For that case, we have to run a "random contest" between S_1, S_2:

- Pick x such that $f_1(x) \neq f_2(x)$ and draw $y \sim \{-1, 1\}$ uniformly at random.
- If $f_1(x) \neq y$, then the output of the contest is $A(S_1 \circ (x, y))$ (since then A makes d mistakes in the online game)
- If $f_2(x) \neq y$, the output of the contest is $A(S_2 \circ (x, y))$.

Adding the ghost example (x, y) forces A to make d mistakes on $S_i \circ (x, y)$. Now if $y = c(x)$, then by the mistake bound of A, it will hold that $A(S_i \circ (x, y)) = c$. Crucially, since y was picked at random, it will hold that $\mathbf{Pr}_y[c(x) = y] = 1/2$. Hence,

$$\mathop{\mathbf{Pr}}_{S_1, S_2, y}[A(S_i \circ (x, y)) = c] \geq \frac{1}{(2d)^2} \cdot 3/4 \cdot 1/2 = \Omega(1/d^2)\,.$$

Now, we can use induction for the other cases. But now, we need to guess more labels, to enforce mistakes on the algorithm. As we guess more labels the success rate reduces, nevertheless we never need to make more then 2^d such guesses.

Summary. We can get global stability using the above inductive argument with stability parameter doubly exponentially small in the depth d.

Theorem 3: Bun et al. (2020)

Let $\mathcal{H}$ be a concept class with Littlestone dimension d. Let $n = \text{poly}(2^{2^d}/\alpha)$. There exists an algorithm A that uses n samples $S \sim \mathcal{D}^n$ from the realizable distribution $\mathcal{D}$ such that

$$\mathbf{Pr}[A(S) = f] \geq 1/2^{2^d}$$

for some hypothesis f and

$$R_{\mathcal{D}}(f) \leq \alpha.$$

We finally add the actual algorithm for completeness. For the details, we refer to Bun et al. (2020).

The Distributions $\tilde{\mathcal{D}}_k$ (a Monte-Carlo variant of $\mathcal{D}_k$)

1. Let n be the auxiliary sample size and N be an upper bound on the number of examples drawn from $\mathcal{D}$.
2. $\tilde{\mathcal{D}}_0$: output the empty sample $\emptyset$ with probability 1.
3. For $k > 0$, define $\tilde{\mathcal{D}}_k$ recursively by the following process:
   - (*) **Throughout the process, if more than N examples from $\mathcal{D}$ are drawn (including examples drawn in the recursive calls), then output "Fail".**
   - (i) Draw $S_0, S_1 \sim \tilde{\mathcal{D}}_{k-1}$ and $T_0, T_1 \sim \mathcal{D}^n$ independently.
   - (ii) Let $f_0 = \mathsf{SOA}(S_0 \circ T_0)$, $f_1 = \mathsf{SOA}(S_1 \circ T_1)$.
   - (iii) If $f_0 = f_1$ then go back to step (i).
   - (iv) Else, pick $x \in \{x : f_0(x) \neq f_1(x)\}$ and sample $y \sim \{\pm 1\}$ uniformly.
   - (v) If $f_0(x) \neq y$ then output $S_0 \circ T_0 \circ ((x, y))$ and else output $S_1 \circ T_1 \circ ((x, y))$.

Figure 1: Taken from Bun et al. (2020). Tournament Process: For $k = 1$, we run A in two different datasets T_0, T_1 until ot outputs two different hypotheses f_0, f_1. Then we force the algorithm to make a mistake and output the extended dataset.

4.2 From Global Stability to Differential Privacy

The above argument gives a way to design a globally stable algorithm for Littlestone classes. We now have to convert it into a differentially private one. This step is not quite difficult.

If A is a (η, n)-globally stable learner with respect to a distribution $\mathcal{D}$, we obtain a DP learner using roughly n/η samples from $\mathcal{D}$ as follows.

Algorithm G

1. Consider the distribution $\tilde{\mathcal{D}}_k$, where the auxiliary sample size is set to $n = \lceil \frac{2^{d+2}}{\alpha} \rceil$ and the sample complexity upper bound is set to $N = 2^{2^{d+2}+1} 4^{d+1} \cdot n$.
2. Draw $k \in \{0, 1, \ldots, d\}$ uniformly at random.
3. Output $h = \mathsf{SOA}(S \circ T)$, where $T \sim \mathcal{D}^n$ and $S \sim \tilde{\mathcal{D}}_k$.

Figure 2: Taken from Bun et al. (2020). Globally Stable Algorithm: Given this algorithm G, which randomly draws the index k of the number of mistakes, one can show that there exists a hypothesis f such that $\mathbf{Pr}[G(S) = f] > \frac{1}{d+1} 2^{-2^{d+2}}$ and $\text{loss}_{\mathcal{D}}(f) < \alpha$.

- First, run A on $k \approx 1/\eta$ independent batches, non-privately producing a list of k hypotheses.
- We next use the 'Stable Histograms' algorithm to this list. This algorithm privately publishes a short list of hypotheses that appear with frequency $\Omega(\eta)$. It privately finds the heavy elements of the list.
- Global stability of the learner A guarantees that with high probability, this list contains some hypothesis h with small population loss.
- Think of this small list as a finite class. We can then apply a generic differentially-private learner (based on the exponential mechanism) on a fresh set of examples to identify such an accurate hypothesis from this short list.

Theorem 4: Stable Histograms Algorithm

Let $\mathcal{X}$ be any data domain. For

$$n \geq O\left(\frac{\log(1/(\eta\beta\delta))}{\eta\epsilon}\right)$$

there exists an $(\epsilon, \delta)-$differentially private algorithm which, with probability at least $1 - \beta$, on input $S = (x_1, ..., x_n)$ outputs a list $L \subseteq \mathcal{X}$ and a sequence of estimates $a \in [0, 1]^{|L|}$ such that

- For every x with $\text{freq}_S(x) \geq \eta$, it holds that $x \in L$.
- For every $x \in L$, it holds that $|a_x - \text{freq}_S(x)| \leq \eta$.

Roughly speaking the idea given the short list of hypotheses is to use the exponential mechanism, which takes in the following input: (i) a dataset $S \in Z^n$, (ii) a set of objects H and (iii) a score function $s : Z^n \times H \to \mathbb{R}$. The only private information is the dataset S. We define the sensitivity of the score function with respect to the dataset only:

$$\Delta_s = \max_{h \in H} \max_{S, S'} |s(S, h) - s(S', h)| .$$

The exponential mechanism outputs an object $h \in H$ with probability proportional to $\exp\left(\frac{\epsilon s(S,h)}{2\Delta_s}\right)$. This mechanism is ϵ-DP and the score of its output is close to $\max_{h \in H} s(S, h)$ with high probability

5 Differential Privacy meets Graph Theory

In this section, we are going to demonstrate a link (by Alon et al. (2024)) between private learnability (both pure and approximate) and *cliques* in certain graphs, which we call *contradiction graphs*.

Definition 3: Clique

A clique in a graph G is a set δ of vertices such that every pair of distinct vertices in δ is connected by an edge.

A fractional clique is a standard LP relaxation of a clique.

Definition 4: Fractional Clique

A function $\delta : V \to [0,1]$ is called a fractional clique if for every independent set $I \subseteq V$, it holds that $\sum_{v \in I} \delta(v) \leq 1$. The size of a fractional clique δ is the sum $\sum_{v \in V} \delta(v)$.

- The clique number of G, denoted $\omega(G)$, is the largest size of a clique in G.
- The fractional clique number of G, denoted $\omega^\star(G)$, is the largest size of a fractional clique in G.

Definition 5: Contradiction Graph

Let $\mathcal{H} \subseteq \{-1,1\}^{\mathcal{X}}$ be a concept class and let $m \in \mathbb{N}$. The contradiction graph of order m of $\mathcal{H}$ is an undirected graph G_m whose vertices are datasets of size m that are consistent with $\mathcal{H}$. Two datasets are connected by an edge whenever they contradict each other.

The vertices of the contradiction graph $G_m(\mathcal{H})$ are $\mathcal{H}$-realizable sequences of length m and $\{S, S'\}$ is an edge of the contradiction graph if there is an example $x \in \mathcal{X}$ such that $(x,0) \in S$ and $(x,1) \in S'$. Given the above, we can get graph-theoretic dimensions of $\mathcal{H}$ that characterize private learning.

Theorem 5: Alon et al. (2024)

The following hold:

- For every $\mathcal{H}$, either $\omega_m = 2^m$ for all m or $\omega_m \leq \text{poly}(m)$.
- For every $\mathcal{H}$, either $\omega^\star_m = 2^m$ for all m or $\omega^\star_m \leq \text{poly}(m)$.
- $\mathcal{H}$ is pure DP learnable if and only if $\omega^\star_m \leq \text{poly}(m)$. ("small" fractional cliques.)
- $\mathcal{H}$ is approximate DP learnable if and only if $\omega_m \leq \text{poly}(m)$. ("small" cliques.)

Observe that the clique and fractional clique dimensions satisfy a (polynomial vs. exponential) dichotomy similar to the Sauer-Shelah-Perles (SSP) dichotomy of the VC dimension.

It is an open problem to use such graph-theoretic notions to characterize other learning settings such as PAC learning.

References

Alon, N., Bun, M., Livni, R., Malliaris, M., and Moran, S. (2022). Private and online learnability are equivalent. *J. ACM*, 69(4).

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.

Alon, N., Moran, S., Schefler, H., and Yehudayoff, A. (2024). A unified characterization of private learnability via graph theory. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 94–129. PMLR.

Bousquet, O. and Elisseeff, A. (2002). Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526.

Bun, M., Livni, R., and Moran, S. (2020). An equivalence between private classification and online prediction. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 389–402. IEEE.

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.

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