4-14-26

Today we'll see another approach to lower bounds for DP learning. Unlike packing and fingerprinting/score attack, what we'll see today is heavily tailored to a particular task: returning any value that's in the "interior" of the dataset.

We'll prove (most of) a surprising lower bound for this task. Although we won't go much beyond this, today's material provides a good starting point for understanding DP PAC learning.

Let X be a totally ordered domain, eg

$$X \subseteq [0,1] \quad \text{or} \quad X = \{0,1,\dots,|X|-1\}$$

Def An algorithm $A : X^n \to X$ solves the interior point problem on X with error probability β if $\forall\, X \in X^n$

$$\Pr\left[\min_{i\in[n]} x_i \le A(x) \le \max_{i\in[n]} x_i\right] \ge 1-\beta$$

Very simple task (not to be confused with "interior point methods" in optimization).

Without privacy, anything works: output any data point. Median is robust solution.

Theorem [Bun, Nissim, Stemmer, Vadhan 2015]

Fix $\varepsilon \in (0, 1/4)$ and $\delta(n) = \frac{1}{50n^2}$. For every $n \in \mathbb{N}$, solving the interior point problem on X with error $\le 1/4$ & $(\varepsilon, \delta(n))$-DP requires

$$n = \Omega\left(\log^*(|X|)\right).$$

Here $\log^*(z)$ is the <u>iterated logarithm</u>, the minimum number k such that:

$$\underbrace{\log(\log(\cdots \log(z))}_{k \text{ times}} \text{ is less than } 1$$

This is an extremely slow-growing function:
for $z \in (16, 65536]$, $\log^*(z) = 4$, and
for $z \in (65536, 2^{65536}]$, $\log^*(z) = 5$.

Before the proof, let's discuss the theorem.

- On the one hand, it is extremely weak: if you are willing to believe all your datapoints can be represented using a computer the size of the known universe, you don't need many examples!
- On the other hand, it is extremely strong: without additional assumptions (about, eg, an underlying data distribution), many

tasks for real-valued statistics are impossible under DP.

- The lower bound has quite a different flavor from the other techniques we've seen, and it's a interesting direction to understand how these two views talk to each other.

- There is an interesting line of work on upper-bounds for this & related problems.

- BNSV studied the interior point problem in the context of private PAC learning. This is the central setting in learning theory. This work sits in a larger body of literature on DP-PAC learning. the highlight of which is the equivalence

between

① DP-PAC learnability

② Online learnability (without privacy)

We won't cover this in any more detail, but it's one of the most notable results in learning theory in the last several years.

The theorem states $\delta(n) = \frac{1}{50n^2}$, but the proof technique can yield the same result for larger δ. Different approaches (as described in Mark Bun's dissertation) get the optimal $\delta(n) \approx \frac{1}{n}$.

Why is this optimal? Consider the following $(0, \delta)$-DP algorithm.

Alg: Name and Shame

Input: $x_1, \ldots, x_n$; $\delta > 0$

$S \leftarrow \emptyset$

For $i = 1, \ldots, n$

  wp δ, add x_i to S

End

Release S

Compare with Leaky $RR_{\varepsilon,\delta}$ & Leaky Inputs from Lecture 11

Recall this will allow us to give an interior point if $S \neq \emptyset$, which happens wp:

$$\Pr[S \neq \emptyset] = 1 - \Pr[S = \emptyset]$$

$$= 1 - \prod_{i=1}^{n} \Pr[x_i \notin S]$$

$$= 1 - \prod_{i=1}^{n} (1 - \delta)$$

$$= 1 - (1-\delta)^n \approx \delta n, \quad \text{when } \delta n \text{ small-ish}$$

So if $\delta \approx \frac{1}{n}$, Name-and-Shame will solve the interior point problem with constant probability.

Main Lemma & Plan for Proof

We will closely follow the presentation in BNSV15, except we will draw some pictures and skip some steps.

Our main lemma will require some notation. Let

$$P_n \triangleq \frac{e^\varepsilon}{e^\varepsilon+1} + (e^\varepsilon+1)\sum_{j=1}^{n} \delta(j) \le 1-\beta$$

$$b(n) \triangleq 1/\delta(n)$$

$s(1) = 2$ and $s(n+1) = b(n)^{s(n)}$, defined recursively

Lemma (Main Lemma) For every $n \in \mathbb{N}$, there exists a distribution D_n over datasets $[S(n)]^n = \{0, 1, \ldots, S(n)-1\}^n$ such that for every $(\varepsilon, S(n))$-DP algorithm A,

$$\Pr_{X \sim D_n}\left[\min X \le A(X) \le \max X\right] \le P_n$$

The distribution D_n will <u>not</u> be a product distribution: the samples will be correlated.

Our proof will use induction and have 3 main steps. First is the

(i) <u>base case</u>: we will write down D_1 & prove the claim for it.

Then the induction step: we assume the claim holds for D_n and use that to construct D_{n+1}. This has two parts:

- Construction: given D_n, how do we define D_{n+1}?
- Reduction: we have to show that D_{n+1} is hard, which we will do using the assumption that D_n is hard.

A bit more before we dive in. When we draw a dataset from D_n, we get n items, where each item is an integer in $[S(n)] = \{0, 1, \ldots, S(n)-1\}$, where $S(n)$ is defined above.

So for $n=1$, D_1 is a distribution over single-item datasets where

each point lives in $[s(1)] = [2] = \{0, 1\}$.

Now, D_{n+1} gives $n+1$ items in $[s(n+1)]$, where $s(n+1) = b(n)^{s(n)}$. We will talk about items in this dataset as follows:

$$(y_0, y_1, \ldots, y_n) \sim D_{n+1}$$

$$y_i = \boxed{y_{i0}}\boxed{y_{i1}}\boxed{y_{i2}}\cdots\boxed{y_{i\,s(n)-1}}$$

each digit is base $b(n)$, in $\{0, 1, \ldots, b(n-1)\}$.

$\longleftarrow s(n)$ digits $\longrightarrow$

Thus the length of datapoints from D_{n+1}, when written in base $b(n)$, is equal to the domain of datapoints from D_n.

That means the domain size gets exponentiated every time we move from D_n to D_{n+1}. This tower of exponentials is exactly the inverse of the iterated logarithm that shows up in the theorem.

Base Case

Take $D_1 = \text{Unif}(\{0,1\})$. Let A be $(\varepsilon, \delta(1))$-DP. A gets a single bit x and solves the interior point problem if $A(x) = x$.

We apply the DP definition:

$$\Pr[A(x) = x] \leq e^{\varepsilon} \Pr[A(\neg x) = x] + \delta(1)$$

$$= e^{\varepsilon}\left(1 - \Pr[A(x) = x]\right) + \delta(1)$$

$$= e^{\varepsilon} - e^{\varepsilon} \Pr[A(x) = x] + \delta(1).$$

We rearrange to get an upper bound on the probability of success:

$$(1 + e^{\varepsilon}) \Pr[A(x) = x] \leq e^{\varepsilon} + \delta(1)$$

$$\Pr[A(x) = x] \leq \frac{e^{\varepsilon}}{1+e^{\varepsilon}} + \frac{1}{1+e^{\varepsilon}} \delta(1)$$

$$\leq \frac{e^{\varepsilon}}{1+e^{\varepsilon}} + (1+e^{\varepsilon})\,\delta(1) = P_1$$

This finishes our base case.

Induction Step: Construction

Given a sample $(x_1, \dots, x_n) \sim D_n$, we define D_{n+1} as follows:

- generate $y_0 \sim \text{Unif}([S(n+1)])$

$$y_0 = \boxed{y_{00}}\,\boxed{y_{01}}\,\boxed{y_{02}}\,\cdots\,\boxed{y_{0,S(n)-1}}$$

↑ all uniform & indep

- For each $i = 1, \dots, n$ independently:

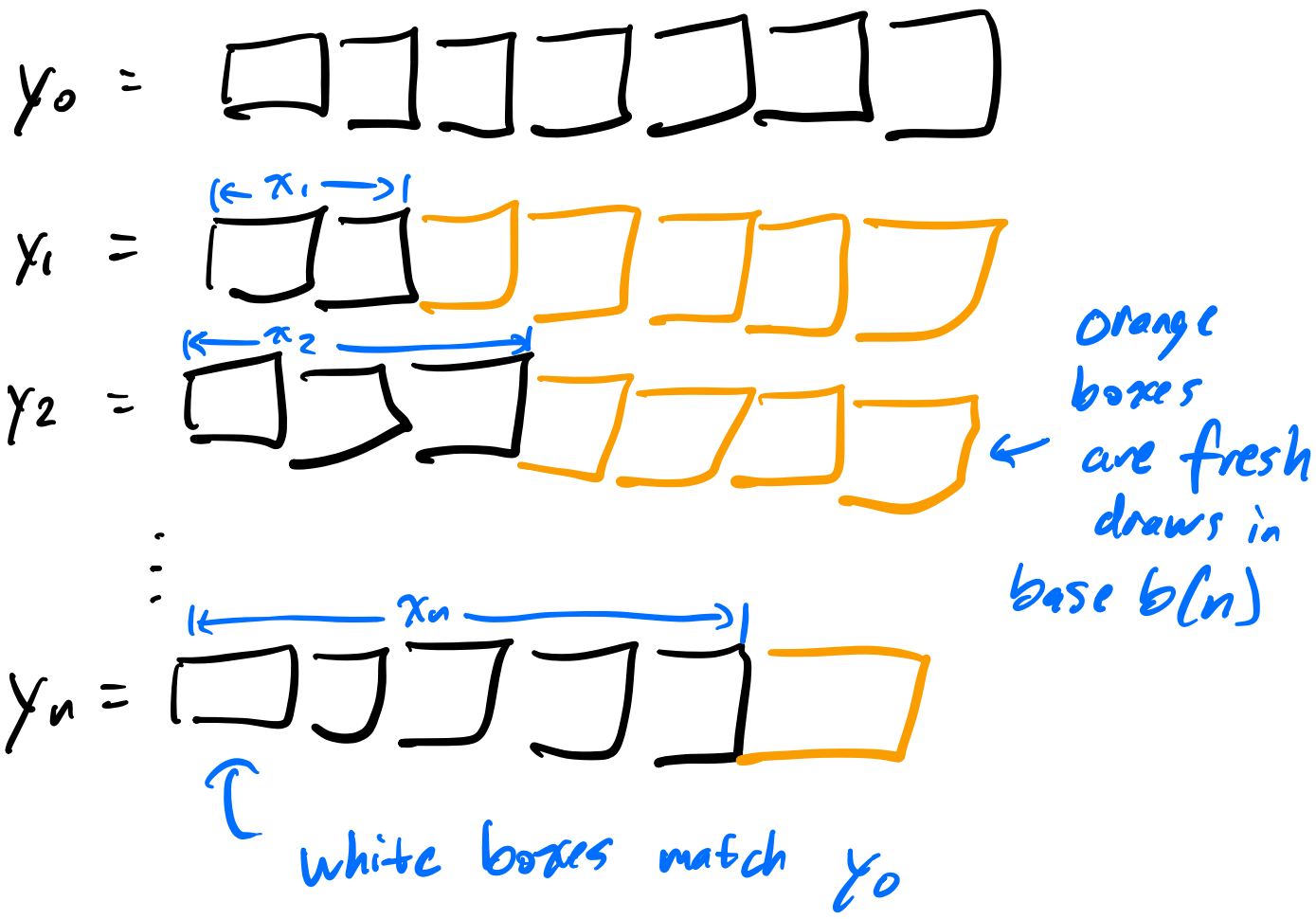
$$y_i = \boxed{y_{00}}\,\boxed{y_{01}}\,\cdots\,\boxed{y_{0x_i}}\,\boxed{y_{i,x_i+1}}\,\cdots\,\boxed{y_{i,S(n)-1}}$$

match y_0 on first x_i digits

↑ rest random & fresh

So drawing $(y_0, y_1, \dots, y_n) \sim D_{n+1}$ is defined in this way.

If we order $x_1 \le x_2 \le \cdots \le x_n$, then the picture is:



Induction Step: Reduction

Assume that any $(\varepsilon, \delta(n))$-DP algorithm on inputs from D_n has success probability at most P_n.

Now consider an algorithm A which is $(\varepsilon, \delta(n+1))$-DP and operates on datasets in $[S(n+1)]^n$. We will show it cannot solve the interior point problem on samples from D_{n+1} with probability greater than P_{n+1}.

To do this, we will use A to construct an algorithm $\hat{A}$ which works for D_n.

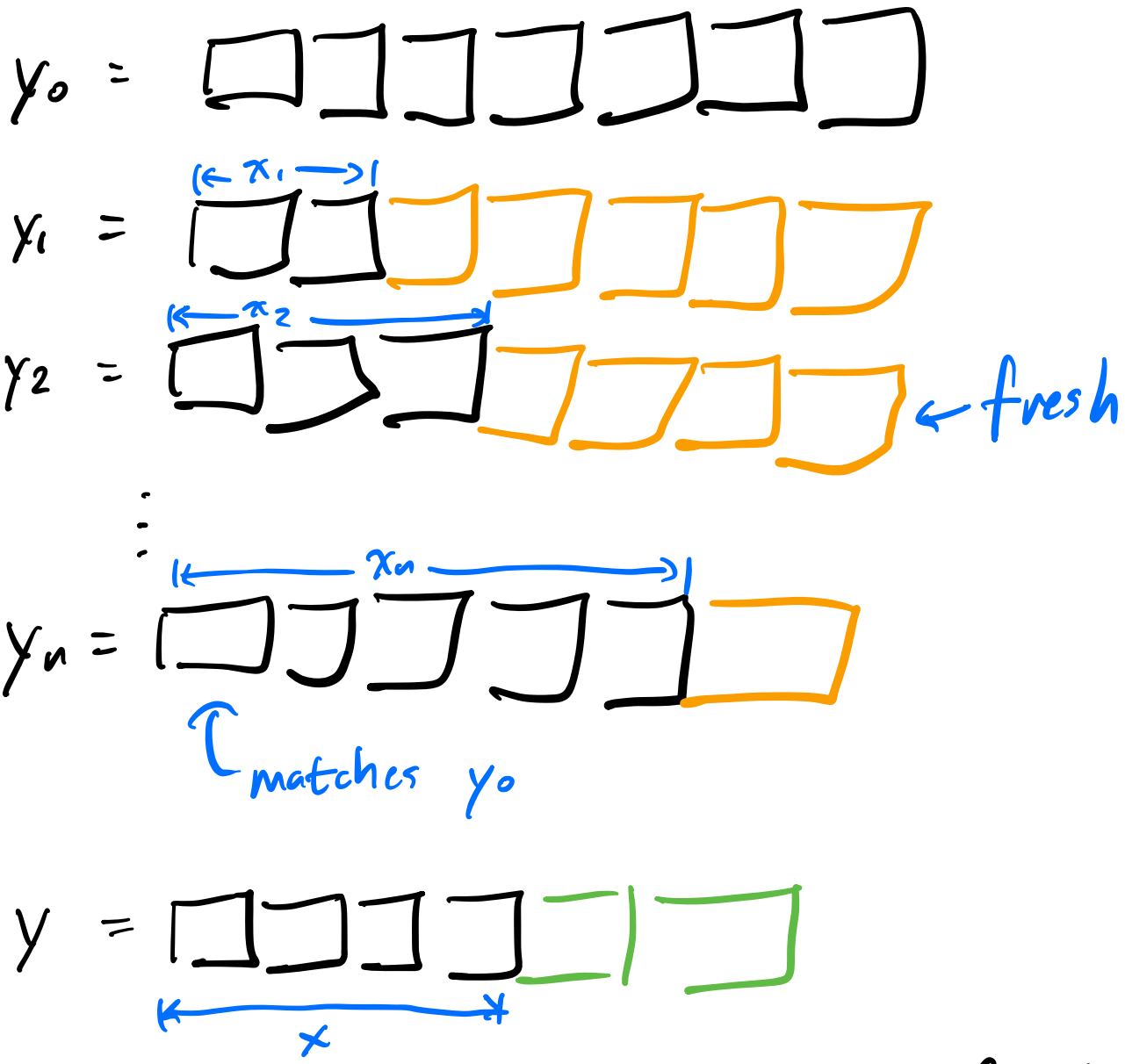
Alg: $\hat{A}$

Input: $X = (x_1, \dots, x_n) \in [S(n)]^n$

1. Construct $Y \in [S(n+1)]^{n+1}$ by sampling from D_{n+1} but starting with X. That is, sample y_0 uniformly and let y_i match y_0 on the first x_i digits (in base $b(n)$) and let the rest of y_i be random.
2. Let $y = A(y_0, y_1, \dots, y_n)$
3. Let $x \in [S(n)]$ be the length of the longest prefix (in base $b(n)$) which matches y_0
4. Return x.

In class, we skipped the proof that $\hat{A}$ is $(\varepsilon, \delta(n))$-DP. (It's a good exercise to check that you are following.)

We want to show that, when A returns some y that is an interior point of $(y_0, y_1, \ldots, y_n)$, then x is (almost always) an interior point of $(x_1, \ldots, x_n)$. Here's the picture:



y will match y_0 on some number of digits.

Claim: If y is an interior point of $(y_0, \dots, y_n)$, then $\min_{1 \le i \le n} x_i \le \chi$

Exercise: convince yourself of this.

In class we ran out of time, but the other side is slightly more involved.

We want to show that $\chi \le \max_i x_i$.

Idea: suppose $y = y_0$, ie they match at every digit. Then in particular they match on the last "digit", which is a random number in the range $\{0, 1, \dots, b(n)-1\}$.

If $\chi > \max_i x_i$, then this random number appeared **only** in y_0, ie only once in the dataset. Since A is DP, it cannot produce this digit with any probability much greater than random guessing, which here is $\frac{1}{b(n)} = \delta(n)$.

Hopefully this is enough to convice the reader that the probability A outputs an interior point on D_{n+1} is very similar to the probability $\hat{A}$ does it on D_n.

Recall that

$$P_{n+1} = \frac{e^\varepsilon}{1+e^\varepsilon} + (1+e^\varepsilon)\sum_{i=1}^{n+1} \delta(i)$$

$$= \frac{e^\varepsilon}{1+e^\varepsilon} + (1+e^\varepsilon)\sum_{i=1}^{n} \delta(i) + (1+e^\varepsilon)\,\delta(n+1)$$

$$= P_n + (1+e^\varepsilon)\,\delta(n+1).$$

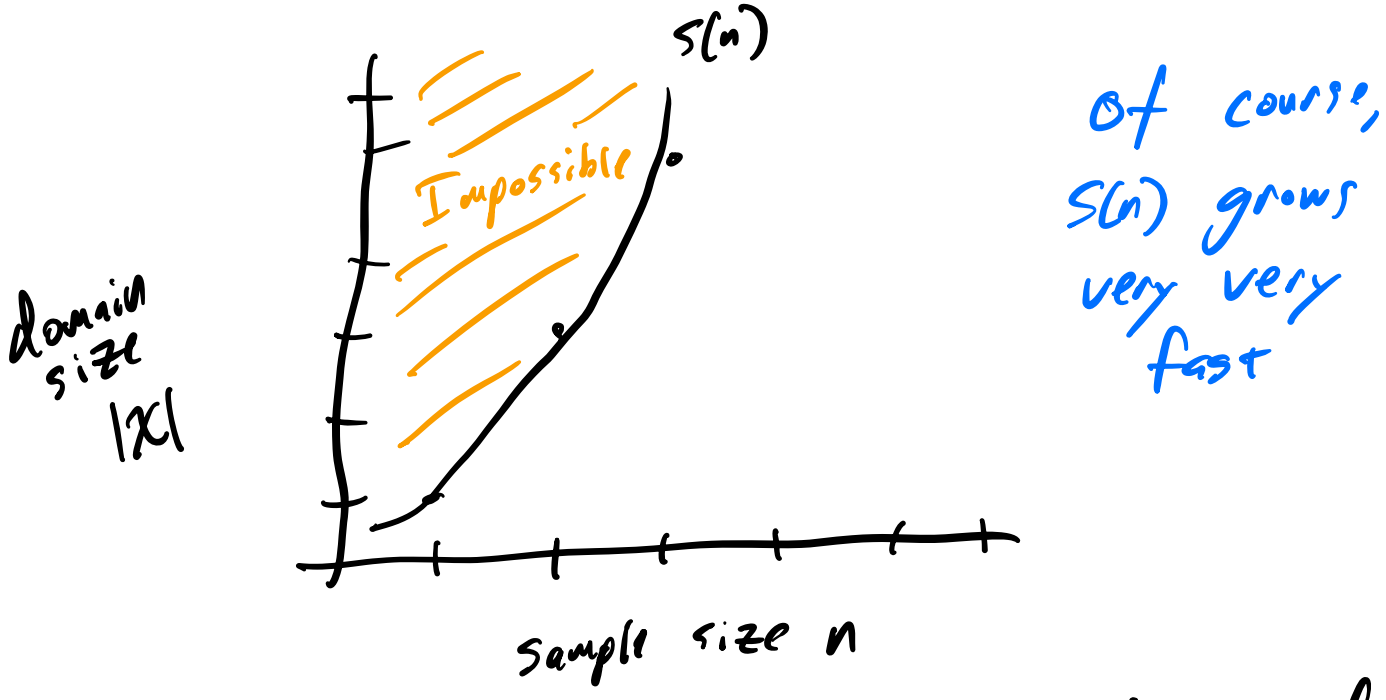
Refer to BNSV15 for the full details, and note I've switched $A \neq \hat{A}$ (sorry!)

This concludes our argument about the induction step.

Conclusions

In the setting of the theorem, we ask: for which domain sizes $|\mathcal{X}|$ and sample sizes n is the DP interior point problem solveable?

The main lemma allows us to answer that: for each n, it shows the problem is hard on the domain $[s(n)]$



of course, $s(n)$ grows very very fast

Anything above this line is impossible: if you give me a domain $\mathcal{X}$ with

$|X| \geq S(n)$, I can put the distribution D_n into that domain, showing you need more than n sampler.

References

BNSV 15 https://arxiv.org/abs/1504.07553

Mark Bun's dissertation

https://dash.harvard.edu/server/api/core/bitstreams/de9843de-2842-40e2-a316-15f4557c4c06/content

Equivalence of DP-PAC learnability & online learnability, Alon et al.

https://dl.acm.org/doi/10.1145/3526074