Lecture 4.2: PAC Learning & Differentially Private PAC Learning

Advanced Topics in Machine Learning

Amartya Sanyal

University of Copenhagen

Oct 1, 2024

Roadmap (90 min total)

- **Part I (30–45 min): Non-private PAC**
  - PAC setup & examples (conjunctions / rectangles)
  - Consistent learning for finite classes
  - VC dimension, uniform convergence (excess risk) $\Rightarrow$ PAC learnability
- **Part II (45–60 min): DP–PAC**
  - (ϵ, δ)-DP PAC definition
  - *Lower bound (packing)* for pure DP
  - *Upper bounds:* EM for $\textsc{Points}_d$; Representation Dimension
  - Pointers: approximate DP & Littlestone dimension

Setup and Notation

- Domain $\mathcal{X}$, labels $\mathcal{Y} = \{0, 1\}$, distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Y}$.
- Hypothesis class $\mathcal{H} \subseteq \{0, 1\}^{\mathcal{X}}$.
- Risk (0–1 loss): $\mathcal{R}\left(h; \mathcal{D}\right) := \Pr_{(x,y)\sim\mathcal{D}}[h(x) \neq y]$.
- Sample $S = ((x_i, y_i))_{i=1}^m \sim \mathcal{D}^m$; empirical risk $\mathcal{R}\left(h; S\right) := \frac{1}{m} \sum_{i=1}^m \mathbf{1}[h(x_i) \neq y_i]$.
- **PAC (realizable)**: $\exists\, h^\star \in \mathcal{H}$ with $\mathcal{R}\left(h^\star; \mathcal{D}\right) = 0$. Learner returns $\hat{h}$ s.t. with prob $\geq 1 - \beta$, $\mathcal{R}\left(\hat{h}; \mathcal{D}\right) \leq \alpha$.
- **Agnostic PAC**: with prob $\geq 1 - \beta$, $\mathcal{R}\left(\hat{h}; \mathcal{D}\right) \leq \inf_{h\in\mathcal{H}} \mathcal{R}\left(h; \mathcal{D}\right) + \alpha$.

Warm-up Example: Conjunctions on $\{0,1\}^d$

Hypotheses: conjunctions of literals x_j or $\neg x_j$ (or absent). Size $|\mathcal{H}| = 3^d$.

Simple algorithm (realizable):

1. Start with all $2d$ literals.
2. For each positive example, delete any literal falsified by it.
3. Output the remaining conjunction.

Consistency: Holds in the realizable case.

Generalization (finite-class bound):

$$m \ \geq \ \frac{1}{\alpha}\Big(\ln|\mathcal{H}| + \ln\tfrac{1}{\beta}\Big) \ = \ \frac{1}{\alpha}\Big(d\ln 3 + \ln\tfrac{1}{\beta}\Big).$$

Example: Axis-Aligned Rectangles in $\mathbb{R}^2$ (Realizable)

Class: $\mathcal{H} = \{h_{[a,b]\times[c,d]}\}$, with $h(x) = 1$ iff $x \in [a, b] \times [c, d]$.

Algorithm: Minimal enclosing rectangle (MER) of positives; predict 1 inside, 0 outside. If no positives, output all-zeros.
Why MER is consistent (realizable): True rectangle contains all positives; MER $\subseteq$ true rectangle; negatives remain outside.
Sample complexity (sketch): If error mass $> \alpha$, one of four "margin corners" has mass $\gtrsim \alpha/4$; with $m = \Theta\big(\frac{1}{\alpha} \log \frac{1}{\beta}\big)$ a point lands there and forces correction.

Consistent Learning for Finite Classes

Consistent ERM. Return h with $\mathcal{R}(h; S) = 0$ whenever such $h \in \mathcal{H}$ exists.

Theorem (realizable finite-class). If $|\mathcal{H}| < \infty$ and the learner is consistent, then

$$\Pr\left[\exists h \in \mathcal{H} :\ \mathcal{R}(h; S) = 0 \,\&\, \mathcal{R}(h; \mathcal{D}) > \alpha\right] \ \le\ |\mathcal{H}|\,(1-\alpha)^m \ \le\ e^{\ln|\mathcal{H}| - \alpha m}.$$

Thus it suffices that

$$m \ \ge\ \frac{1}{\alpha}\Big(\ln|\mathcal{H}| + \ln\tfrac{1}{\beta}\Big).$$

VC Dimension: $\Pi_{\mathcal{C}}(S)$, Shattering, $\mathrm{VCD}(\mathcal{C})$

Labeling set. For $S = \{x_1, \ldots, x_m\} \subseteq \mathcal{X}$ and $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$,

$$\Pi_{\mathcal{C}}(S) \;:=\; \{(h(x_1), \ldots, h(x_m)) : \; h \in \mathcal{C}\}.$$

Shattering. S is *shattered* by $\mathcal{C}$ if $\Pi_{\mathcal{C}}(S) = \{0,1\}^m$.

VC dimension.
$\mathrm{VCD}(\mathcal{C}) := \max\{m : \; \exists S, |S| = m, \; S \text{ shattered}\}$ [Vap13].

Example. Intervals on $\mathbb{R}$ have $\mathrm{VCD}(\cdot) = 2$ (cannot realize 1-0-1 on three ordered points).

Exercise. For $\Delta_c(\mathcal{C}) = \{x \mapsto h(x) \oplus h'(x)\}$, relate $\mathrm{VCD}(\Delta_c(\mathcal{C}))$ to $\mathrm{VCD}(\mathcal{C})$.

Uniform Convergence and the VC Excess-Risk Bound

Uniform deviation. If $\mathrm{VCD}(\mathcal{H}) = d$, then w.p. $\geq 1 - \beta$,

$$\sup_{h \in \mathcal{H}} \left| \mathcal{R}(h; \mathcal{D}) - \mathcal{R}(h; S) \right| \;\leq\; c \sqrt{\frac{d \log\left(\frac{em}{d}\right) + \log\left(\frac{2}{\beta}\right)}{m}}.$$

ERM excess risk (agnostic). For $\hat{h} \in \arg\min_{h \in \mathcal{H}} \mathcal{R}(h; S)$,

$$\mathcal{R}\left(\hat{h}; \mathcal{D}\right) - \inf_{h \in \mathcal{H}} \mathcal{R}(h; \mathcal{D}) \;\leq\; 2c \sqrt{\frac{d \log\left(\frac{em}{d}\right) + \log\left(\frac{2}{\beta}\right)}{m}}.$$

Hence

$$m \;=\; \widetilde{\mathcal{O}}\left(\frac{d + \log(1/\beta)}{\alpha^2}\right).$$

Definition: (ϵ, δ)-DP–PAC Learning

Privacy. $\mathcal{A}$ is (ϵ, δ)-DP if for neighboring S, S' and any measurable E,

$$\Pr[\mathcal{A}(S) \in E] \;\leq\; e^{\epsilon} \Pr[\mathcal{A}(S') \in E] + \delta.$$

Utility (PAC). $\mathcal{A}$ is an $(\alpha, \beta, \epsilon, \delta)$-DP–PAC learner for $\mathcal{H}$ if, for all $\mathcal{D}$,

$$\Pr_{S \sim \mathcal{D}^m}\Big[\, \mathbb{E}_{h \sim \mathcal{A}(S)}[\mathcal{R}(h; \mathcal{D})] \;\leq\; \alpha \,\Big] \;\geq\; 1 - \beta.$$

Packing Lower Bounds: General Recipe (Pure DP)

Goal. Lower bound m for any $(\alpha, \beta, \epsilon, 0)$-DP–PAC learner.
Step 1 (separation). Build N instances $\{(\mathcal{D}_\theta, h_\theta)\}$ with

$$\mathcal{R}(h_\theta; \mathcal{D}_\theta) = 0, \qquad \mathcal{R}(h_{\theta'}; \mathcal{D}_\theta) \geq \alpha \quad (\theta \neq \theta').$$

Step 2 (utility $\Rightarrow$ identification). Under $\mathcal{D}_\theta$, with prob $\geq 1 - \beta$ the learner outputs from a good set $\mathcal{G}_\theta$ (often $\{h_\theta\}$).
Step 3 (DP stability). Couple $S \sim \mathcal{D}_\theta^m$, $S' \sim \mathcal{D}_{\theta'}^m$ with Hamming distance $\leq \Delta$ (w.h.p.); group privacy gives

$$\Pr[\mathcal{A}(S') \in E] \;\geq\; e^{-\epsilon\Delta} \Pr[\mathcal{A}(S) \in E].$$

Step 4 (counting). For fixed θ',

$$\beta \;\geq\; \sum_{\theta \neq \theta'} \Pr[\mathcal{A}(S_{\theta'}) \in \mathcal{G}_\theta] \;\geq\; (N-1)e^{-\epsilon\Delta}(1-\beta).$$

Solve for m via Δ.

Packing Lower Bound for $\textsc{Points}_d$ (Pure DP)

Class. $\mathcal{X} = \{0, \ldots, 2^d - 1\}$, $h_a(x) = \mathbf{1}[x = a]$, $N = 2^d$.
Choice A (full spike; $\Delta = m$). $\mathcal{D}_a$ puts all mass on $(x = a, y = 1)$. Then S_a and S_b differ in all m entries; group privacy gives

$$\beta \;\geq\; (2^d - 1)e^{-\epsilon m}(1 - \beta) \;\Rightarrow\; m \;\geq\; \frac{d \ln 2 + \ln\left(\frac{1-\beta}{\beta}\right)}{\epsilon}.$$

Choice B (spike+slab; $\Delta \approx \alpha m$). $\mathcal{D}_a$ has mass α on $(a, 1)$ and $(1 - \alpha)$ on negatives elsewhere. Typical S_a, S_b differ in $\approx \alpha m$ entries; thus

$$m \;\geq\; \Omega\left(\frac{d + \log(1/\beta)}{\alpha\,\epsilon}\right).$$

EM Upper Bound for $\textsc{Points}_d$: Step 1 — Make a gap

Counts (scores). $C_a := \#\{i : \ x_i = a, \ y_i = 1\}$. Realizable: $C_{a^\star} \sim \mathrm{Bin}(m, \alpha)$ and $C_b = 0$ for $b \neq a^\star$.
Chernoff (lower tail). With prob $\geq 1 - \beta/2$,

$$C_{a^\star} \ \geq \ \frac{\alpha m}{2} \quad \text{if} \quad m \ \geq \ \frac{8}{\alpha} \ln \frac{2}{\beta}.$$

EM Upper Bound for $\textsc{Points}_d$: Step 2 — Softmax picks the top

EM rule. $\Pr[\hat{a} = a] \;=\; \dfrac{\exp\!\left(\frac{\epsilon}{2} C_a\right)}{\sum_u \exp\!\left(\frac{\epsilon}{2} C_u\right)}.$

Compare $b \neq a^\star$ to $a^\star$:

$$\frac{\Pr[\hat{a} = b]}{\Pr[\hat{a} = a^\star]} = \exp\!\Big(-\tfrac{\epsilon}{2}\,(C_{a^\star} - C_b)\Big) \;\Rightarrow\; \Pr[\hat{a} = b] \;\le\; \exp\!\Big(-\tfrac{\epsilon}{2}\,(C_{a^\star} - C_b)\Big).$$

Union bound.

$$\Pr[\hat{a} \neq a^\star] \;\le\; \sum_{b \neq a^\star} \exp\!\Big(-\tfrac{\epsilon}{2}\,(C_{a^\star} - C_b)\Big).$$

On the good event: $C_{a^\star} \geq \alpha m/2$ and $C_b = 0$,

$$\Pr[\hat{a} \neq a^\star] \;\le\; (2^d - 1)\,\exp\!\Big(-\tfrac{\epsilon}{2} \cdot \tfrac{\alpha m}{2}\Big) \;\le\; \tfrac{\beta}{2}$$

whenever

$$m \;\geq\; \frac{4}{\alpha\epsilon}\Big(d \ln 2 + \ln \tfrac{2}{\beta}\Big).$$

Combine Step 1+2:

$$m \;\geq\; \max\Big\{ \tfrac{8}{\alpha} \ln \tfrac{2}{\beta},\; \tfrac{4}{\alpha\epsilon}(d \ln 2 + \ln \tfrac{2}{\beta}) \Big\} \;=\; O\Big(\tfrac{d + \log(1/\beta)}{\alpha\epsilon}\Big).$$

Representation Dimension: Tiny Menus That Still Work

Why. EM utility pays $\log |\mathcal{O}|$; huge $\mathcal{H}$ is costly. *Idea:* sample a tiny menu $\mathcal{H}'$ that still contains an α-good hypothesis for any $\mathcal{D}$ w.h.p.

Probabilistic (α, β)-representation. A distribution $\mathcal{Q}$ over sub-classes $\mathcal{H}' \subseteq \mathcal{H}$ is an (α, β)-*representation* if for every $\mathcal{D}$,

$$\Pr_{\mathcal{H}' \sim \mathcal{Q}}\left[\exists h \in \mathcal{H}' : \ \mathcal{R}(h; \mathcal{D}) \ \leq \ \inf_{g \in \mathcal{H}} \mathcal{R}(g; \mathcal{D}) \ + \ \alpha\right] \ \geq \ 1 - \beta.$$

Representation Dimension.

$$\mathrm{RepDim}_{\alpha}(\mathcal{H}) \ := \ \min \ \left\{k : \ \exists\,(\alpha, \tfrac{1}{3})\text{-representation } \mathcal{Q} \text{ with } |\mathcal{H}'| \leq 2^{k} \text{ a.s.}\right\}.$$

Using RepDim with EM (Improper, Pure DP Upper Bound)

Learner.

1. Sample $\mathcal{H}' \sim \mathcal{Q}$ from an $(\alpha, \frac{1}{3})$-representation (data-independent), with $|\mathcal{H}'| \leq 2^{\mathrm{RepDim}_\alpha(\mathcal{H})}$.
2. Run EM on $\mathcal{H}'$ with score $q(S, h) = -m\,\mathcal{R}\,(h; S)$ (sensitivity 1).
3. Output $\hat{h} \in \mathcal{H}'$.

Privacy. EM is $(\epsilon, 0)$-DP; sampling $\mathcal{H}'$ is free (independent of S).
Utility. By the representation guarantee and EM utility (pays $\log|\mathcal{H}'|$), plus DP⇒generalization, we get

$$\mathcal{R}\left(\hat{h}; \mathcal{D}\right) \leq \inf_{h\in\mathcal{H}} \mathcal{R}\,(h; \mathcal{D}) + \alpha$$

provided

$$m = O\left(\frac{\mathrm{RepDim}_\alpha(\mathcal{H}) + \ln(1/\beta)}{\alpha\,\epsilon}\right).$$

Approximate DP: Relation to Online Learnability (Pointers)

Relax $(\epsilon, 0)$ to (ϵ, δ) and allow *improper* learning:

- Any $(\alpha, \beta, \epsilon, \delta)$-DP–PAC learner requires $\Omega(\log^*(\mathrm{LDim}\,(\mathcal{H})))$ samples [ALMM19].
- There exists an $(\alpha, \beta, \epsilon, \delta)$-DP–PAC learner using $\mathrm{poly}\,(()\,\mathrm{LDim}\,(\mathcal{H}))$ samples [BLM20].

Moral. Private learnability (approx. DP, improper) $\iff$ online learnability (finite Littlestone dimension), up to poly factors.

In-Class Exercises (Optional)

1. Prove $\mathrm{VCD}\left(\right)$ (intervals on $\mathbb{R}$) $= 2$.
2. Realizable PAC for rectangles in $\mathbb{R}^d$ via "corner" argument.
3. ($\textsc{Points}_d$) Formalize packing to get $m = \Omega((\log d + \log(1/\beta))/(\alpha\epsilon))$.
4. (RepDim) Given an $(\alpha, 1/3)$-representation with $|\mathcal{H}'| \leq 2^k$, analyze EM and derive the pure-DP upper bound.

Takeaways

- **Non-private:** Finite VC $\Rightarrow$ uniform convergence $\Rightarrow$ ERM learns with $m = \widetilde{\mathcal{O}}\left(\frac{d+\log(1/\beta)}{\alpha^2}\right)$.
- **Pure DP:** Privacy can be strictly costlier (e.g., $\textsc{Points}_d$). Packing yields strong lower bounds.
- **Pure DP upper:** RepDim replaces $\log|\mathcal{H}|$; get $m = O((\mathrm{RepDim}_\alpha(\mathcal{H}) + \log(1/\beta))/(\alpha\epsilon))$.
- **Approx DP:** Tied to online learnability (Littlestone).

[ALMM19] Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private pac learning implies finite littlestone dimension. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 852–860, 2019.

[BLM20] Mark Bun, Roi Livni, and Shay Moran. An equivalence between private classification and online prediction. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, 2020.

[Vap13] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

Lecture 5.1: Online Learning, Mistake bound model, Littlestone dimension

Advanced Topics in Machine Learning

Amartya Sanyal

University of Copenhagen

Oct 9, 2024

Introduction

Core assumption of Statistical Learning

- Studied i.i.d. sample from some unknown distribution $\mathcal{D}$.
- Performance evaluated on $\mathcal{D}$.
- This is necessary for generalisation.

Talk about Agnostic learning and mention Sauer Shelah Lemma

Today: the online learning model

Assumption: data is labelled by some classifier in $\mathcal{H}$.

There is no distribution over data.

Online Model

Realistically,

- Decisions made without future knowledge.
- Example: Deciding when to leave for a flight.
- Goal: Make right decisions in the long run.

Game between adversary and learner

- Learner $\mathcal{A}$ tries to make predictions.
- Adversary provides data points and true labels.
- Both know the hypothesis class $\mathcal{H}$.
- Game progresses in a series of steps.

Mistake Bound Model

1. Adversary presents data point x_t.
2. Learner predicts a label $\hat{y}_t$.
3. Adversary reveals the true label y_t.
4. Mistake occurs if $\hat{y}_t \neq y_t$.

Definition

An algorithm $\mathcal{A}$ is said to online learn a hypothesis class $\mathcal{H}$ with mistake bound M if for any sequence $((x_0, y_0), \ldots, (x_T, y_T))$ of length T, for any $T \in \mathbb{Z}_+$, that is realisable by some $h^* \in \mathcal{H}$ i.e. $h^*(x_t) = y_t$ for all $t \in \{1, \ldots, T\}$, the total number of mistakes incurred by $\mathcal{A}$ is bounded by M i.e. $\mathcal{M}(\mathcal{A}) \leq M$.

In class exercise

Exercise

Let $\mathcal{X}_d = \{0,1\}^d$ be the boolean hypercube and let $\mathcal{Y} = \{0,1\}$. Let $\mathcal{H}$ be the class of monotone conjunctions in d dimensions (i.e. all functions of the form $x_1 \wedge x_2 \wedge x_{10}$ etc. where no literal is negated). Design an algorithm that online learns $\mathcal{H}$ with finite mistake bound M. Make M as small as possible.

Algorithm: Trivial Online Learner for Finite Classes I

Algorithm Trivial Algorithm

Require: Hypothesis class $\mathcal{H}$

Let $\mathcal{C} = \mathcal{H}$
for t $= 1 \ldots T$ **do**
  Receive x_t
  Choose $\hat{h} \in \mathcal{C}$ randomly and **output** $\hat{y}_t = \hat{h}(x_t)$
  Receive y_t
  if $\hat{y}_t \neq y_t$ **then**
    $\mathcal{C} = \mathcal{C} \setminus \left\{\hat{h}\right\}$
  end if
end for

Theorem

For any finite hypothesis class $\mathcal{H}$, this algorithm learns $\mathcal{H}$ with mistake bound $|\mathcal{H}|$.

Algorithm: Halving Algorithm for Finite Classes I

We can do much better than $|\mathcal{H}|$.

Algorithm Halving Algorithm

Require: Hypothesis class $\mathcal{H}$

Let $\mathcal{C} = \mathcal{H}$

for t $= 1 \ldots T$ **do**

Receive x_t

Output $\hat{y}_t = \mathrm{argmax}_{l \in \{0,1\}} \left| h \in \mathcal{H} \text{ such that } \hat{h}(x_t) = l \right|$

Receive y_t

if $\hat{y}_t \neq y_t$ **then**

$\mathcal{C} = \mathcal{C} \setminus \{h \in \mathcal{H} \text{ such that } h(x_t) \neq y_t\}$

end if

end for

Theorem

For any finite hypothesis class $\mathcal{H}$, the Halving algorithm learns $\mathcal{H}$ with mistake bound $\log(|\mathcal{H}|)$.

Takeaways

- Similar to PAC learning. For example if $\mathcal{H}$ has 2^d functions, then this algorithm makes at most d mistakes.
- Can we expect a similar algorithm to work for infinite hypothesis classes?

Theorem

Let $\mathcal{X} = (0,1), \mathcal{Y} = \{0,1\}$, and $\mathcal{H} = \{h_a : a \in (0,1)\}$ where $h_a(x) = \mathbb{I}\{x \geq a\}$ be the class of one dimensional linear thresholds on $\mathcal{X} \times \mathcal{Y}$. The, for any $T > 0$ and any online learner $\mathcal{A}$, there exists a sequence $\{(x_1, y_1), \ldots, (x_T, y_t)\} \in (\mathcal{X} \times \mathcal{Y})^T$ such that $\mathcal{A}$ make T mistakes on the sequence.

For Large T: $T \to \infty$, the number of mistakes incurred by any algorithm for one dimensional thresholds also goes to infinity.

Existence of hypothesis classes that are PAC learnable but not online learnable.

Infinite Hypothesis Classes

- Does not mean that all infinite hypothesis classes are not online learnable.
- Possible to characterise online learnability of hypothesis classes with a combinatorial complexity of hypothesis similar to what we did for PAC learning.

Complexity Measure

- Clearly, VC dimension is not the right measure (see Theorem 3).
- Need a complexity measure that is larger than VC dimension but smaller than $|\mathcal{H}|$.

Shattering a Tree

To understand the new measure, we will first need to understand the concept of *shattering a tree*.

Definition

Consider a full binary tree of depth d such that each node is indexed by some $x \in \mathcal{X}$. This tree is said to be shattered by $\mathcal{H}$ if for every set of labels $\{y_i\}_{i=1}^{d} \in \{0,1\}^d$, the root to leaf path $x_1, \ldots, x_d$ dictated by taking the left child at node x_i if $y_i = 0$ and right child otherwise is such that $\exists h \in \mathcal{H}$ where $h(x_i) = y_i$ for all $1 \leq i \leq d$.

Intuition Intuitively, a tree is shattered if for every path from root to leaf of the tree, there is a hypothesis in $\mathcal{H}$ which correctly predicts the label of a node.

Littlestone Dimension

Definition

The littlestone dimension [Lit88] of a hypothesis class $\mathcal{H}$ is the maximal depth such that a tree of that depth can be shattered by $\mathcal{H}$.

Properties of Littlestone Dimension

$$\mathrm{VCD}(\mathcal{H}) \leq \mathrm{LDim}(\mathcal{H}) \leq \log(|\mathcal{H}|)$$

Exercise

Show that the Littlestone dimension of the class of one-dimensional thresholds is ∞.

Theorem: Online Learnability and Littlestone Dimension

Theorem

For any hypothesis class $\mathcal{H}$,

1. *There is an online learning algorithm $\mathcal{A}$ that makes at most $\mathrm{LDim}(\mathcal{H})$ mistakes on any sequence of points labelled by some $h^* \in \mathcal{H}$.*
2. *For any algorithm $\mathcal{A}$, there exists a sequence of points labelled by some $h^* \in \mathcal{H}$ such that $\mathcal{A}$ makes at least $\mathrm{LDim}(\mathcal{H})$ mistakes on the sequence.*

- Theorem 4 characterises online learnability of any hypothesis class exactly with Littlestone dimension just like VC does for PAC learning.
- We'll discuss the proof sketch for showing that any hypothesis class $\mathcal{H}$ can be online learned with a mistake bound of $\mathrm{LDim}(\mathcal{H})$.
- **Core idea**: Similar to halving algorithm.

Algorithm: Standard Optimal Algorithm

Algorithm Standard Optimal Algorithm

Require: Hypothesis class $\mathcal{H}$
  Let $\mathcal{C} = \mathcal{H}$
  for t $= 1 \ldots T$ **do**
    Receive x_t
    Output $\hat{y}_t = \operatorname{argmax}_{l \in \{0,1\}} \operatorname{LDim}\left(h \in \mathcal{H} \text{ such that } \hat{h}(x_t) = l\right)$
    Receive y_t
    if $\hat{y}_t \neq y_t$ **then**
      $\mathcal{C} = \mathcal{C} \setminus \{h \in \mathcal{H} \text{ such that } h(x_t) \neq y_t\}$
    end if
  end for

- The algorithm makes at most $\operatorname{LDim}(\mathcal{H})$ mistakes.
- For every mistake, Littlestone dimension decreases by 1.

To prove the other direction: Use an adversary strategy to show at least $\operatorname{LDim}(\mathcal{H})$ mistakes.

Conclusion

Thus, so far we have:

- VC dimension characterises PAC learnability.
- Littlestone dimension characterises Online learnability.

Next lecture: Can Differentially Private (DP)-PAC learnability be characterised by some complexity measure?

[Lit88] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 1988.