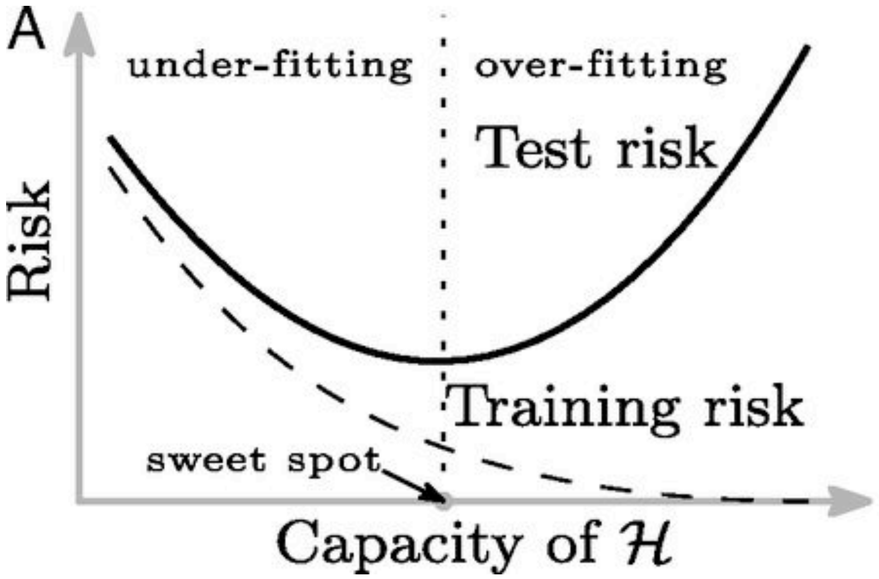
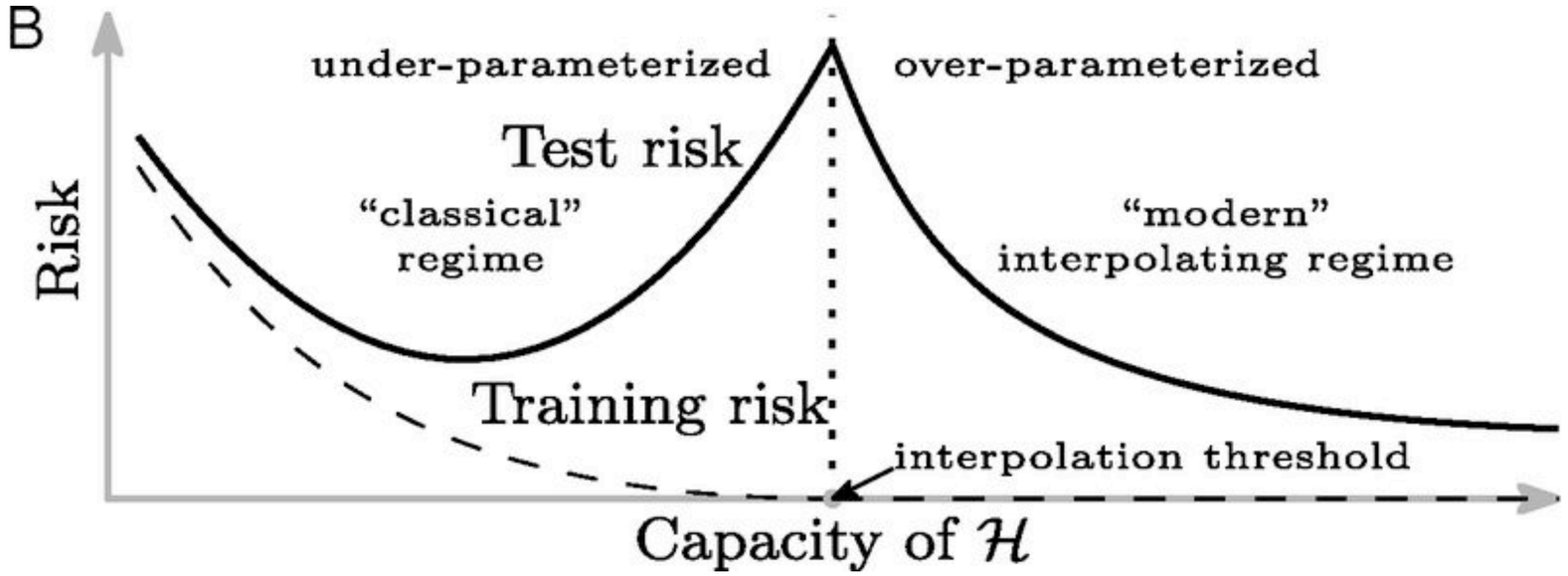
Grab bag:
Failures of uniform convergence
PAC-Bayes
Online learning

CPSC 532D: Modern Statistical Learning Theory

7 Dec 2023

cs.ubc.ca/~dsuth/532D/23w1/



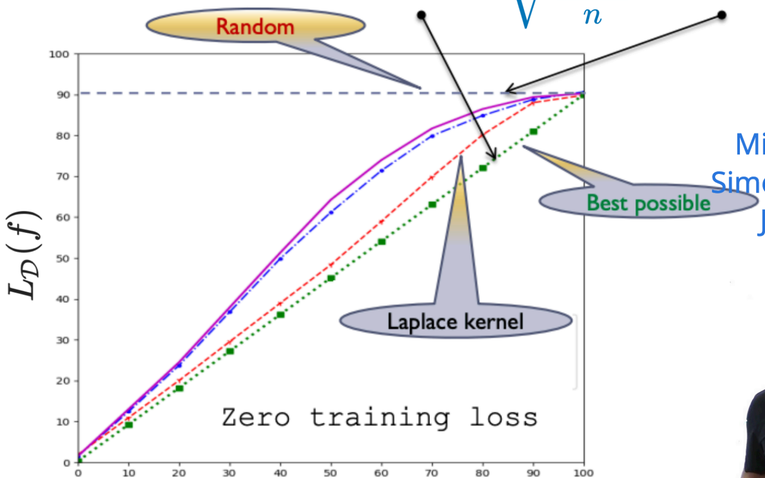


Bounds?

$$L_{\mathcal{D}}(\hat{f}) \leq \underbrace{L_{\mathbf{S}}(\hat{f})}_{0} + \sqrt{\frac{C_{\mathcal{F},\delta}}{n}}$$

What kind of generalization bound could work here?

$0.7 < \sqrt{\frac{C_{\mathcal{F},\delta}}{n}} < 0.9$



Misha Belkin
Simons Institute
July 2019

Belkin/Ma/Mandal, ICML 2018

Uniform convergence may be unable to explain generalization in deep learning

Vaishnavh Nagarajan
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA
vaishnavh@cs.cmu.edu

J. Zico Kolter
Department of Computer Science
Carnegie Mellon University &
Bosch Center for Artificial Intelligence
Pittsburgh, PA
zkolter@cs.cmu.edu

- [see the board pictures posted on Canvas for this stuff]

(pause)

A road to PAC-Bayes

- Bayesians say:
  - Start with a prior distribution $\pi(h)$ on choice of hypothesis
  - Observe data S with likelihood $\mathscr{L}(S \mid h)$
  - End up with posterior distribution $\rho(h \mid S) \propto \mathscr{L}(S \mid h)\,\pi(h)$
  - Make predictions/decision based on posterior mean/median, MAP, single draw, …
- This is optimal if you believe in your prior + likelihood! 😊
  - Frequentists say: “but how good is it actually???”
  - What if your model class / prior / … are wrong?
- Tempered likelihood (Zhang 06) / SafeBayes (Grünwald 12):
  - If your model is misspecified, can be provably better to use $\mathscr{L}^\lambda$ for $\lambda < 1$
  - No longer quite Bayesian inference, but turns a prior into a posterior
- PAC-Bayes: analyzes *any* prior-posterior pair (potentially even totally unrelated)

PAC-Bayes: McAllester bound

- We start with some prior π (independent of the data S) on hypotheses
- Our learning algorithm sees S and gives us a posterior ρ
- We'll analyze $L_{\mathcal{D}}(\rho) = \mathbb{E}_{h\sim\rho}[L_{\mathcal{D}}(h)]$ based on $L_S(\rho) = \mathbb{E}_{h\sim\rho}[L_S(h)]$
- McAllester-style bound (SSBD theorem 31.1):
  - If $\ell(h, z) \in [0,1]$, with probability at least $1-\delta$ over $S \sim \mathcal{D}^n$,

$$L_{\mathcal{D}}(\rho) - L_S(\rho) \leq \sqrt{\frac{\mathrm{KL}(\rho\|\pi) + \log\frac{n}{\delta}}{2(n-1)}}$$

    where $\mathrm{KL}(\rho\|\pi) = \mathbb{E}_{h\sim\rho} \log\frac{\rho(h)}{\pi(h)}$ (the usual KL divergence)

  - Proved in SSBD chapter 31 (not bad at all)

What learning algorithm?

$$L_{\mathscr{D}}(\rho) - L_S(\rho) \leq \sqrt{\frac{\mathrm{KL}(\rho \| \pi) + \log \frac{n}{\delta}}{2(n-1)}}$$

- What's the best learning algorithm, according to this bound?
  - Turns out to be the **Gibbs posterior**: $\rho(h) \propto \exp(-\lambda L_S(h))\ \pi(h)$
  - Same as tempered likelihood / SafeBayes if $\mathscr{L}(S \mid h) = -\log L_S(h) + \text{const}$
  - Typical choice (see 340): e.g. squared loss $\leftrightarrow$ Gaussian likelihood

- But the bound applies to **any** prior-posterior pair (with π independent of S)
  - For instance: could learn a $\hat{h}$ with (S)GD and then add noise to it
  - If $\hat{h}$ is in a **flat minimum**, then $\hat{h}$ + noise will still be good
  - But note that if $\rho \to$ point mass and π continuous, $\mathrm{KL}(\rho \| \pi) \to \infty$

What prior?

$$L_{\mathscr{D}}(\rho) - L_S(\rho) \leq \sqrt{\frac{\mathrm{KL}(\rho \| \pi) + \log \frac{n}{\delta}}{2(n-1)}}$$

- What's the best prior?
  - Bound on generalization gap is better if ρ is "closer" to π
    - S didn't make us "change our mind" too much – similar to MDL
  - But we also want a good ρ, i.e. average training loss $L_S(\rho)$ should be small
- Notice π only shows up in the bound – nothing to do with the learning algorithm
  - We could potentially pick a prior that actually **depends on** $\mathscr{D}$
  - …as long as we can still bound $\mathrm{KL}(\rho \| \pi)$

Other forms of PAC-Bayes bounds

- Linear bound: $L_{\mathscr{D}}(\rho) \leq \frac{1}{\beta} L_S(\rho) + \frac{\mathrm{KL}(\rho\|\pi) + \log\frac{1}{\delta}}{2\beta(1-\beta)n}$ for any $\beta \in (0,1)$

- Catoni bound: for $\alpha > 1$, $\Phi_\gamma^{-1}(x) = (1-\exp(-\gamma x))/(1-\exp(-\gamma))$,

  $$L_{\mathscr{D}}(\rho) \leq \inf_{\lambda>1} \Phi_{\lambda/n}^{-1}\left(L_S(\rho) + \frac{\alpha}{\lambda}\left[\mathrm{KL}(\rho\|\pi) - \log\varepsilon + 2\log\frac{\log(\alpha^2\lambda)}{\log\alpha}\right]\right)$$

  - Can be much tighter (unfortunately) if $\mathrm{KL}(\rho\|\pi)/n$ is big

- Also variants based on general f-divergences, Wasserstein, …

NON-VACUOUS GENERALIZATION BOUNDS AT THE IMAGENET SCALE: A PAC-BAYESIAN COMPRESSION APPROACH

Wenda Zhou
Columbia University
New York, NY
wz2335@columbia.edu

Victor Veitch
Columbia University
New York, NY
victorveitch@gmail.com

Morgane Austern
Columbia University
New York, NY
ma3293@columbia.edu

Ryan P. Adams
Princeton University
Princeton, NJ
rpa@princeton.edu

Peter Orbanz
Columbia University
New York, NY
porbanz@stat.columbia.edu

- Pre-pick a coding scheme to represent networks (e.g. compress the weights)
- Train a network with SGD, sparsify it/etc to $\hat{h}$, then add a little noise to weights

Table 1: Summary of bounds obtained from compression

Dataset	Orig. size	Comp. size	Robust. Adj.	Eff. Size	Error Bound Top 1	Error Bound Top 5
MNIST	168.4 KiB	8.1 KiB	1.88 KiB	6.23 KiB	$< 46\,\%$	NA
ImageNet	5.93 MiB	452 KiB	102 KiB	350 KiB	$< 96.5\,\%$	$< 89\,\%$

Derandomizing PAC-Bayes

- In practice, we don't actually use randomized predictors (usually)
- Possible to "derandomize" to a high-probability bound on $L_{\mathcal{D}}(h) - L_S(h)$:
  - Show convergence of $L_{\mathcal{D}}(h)$ to $\mathbb{E}_{h\sim\rho}L_{\mathcal{D}}(h)$, $L_S(h)$ to $\mathbb{E}_{h\sim\rho}L_S(h)$, under ρ
  - Or, use a margin-type loss to show 0-1 error doesn't change under ρ
- But…these then become "two-sided" bounds
  - Subject to the Nagarajan/Kolter failure mode (their Appendix J)

Uniform convergence may be unable to explain generalization in deep learning

---

Vaishnavh Nagarajan
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA
vaishnavh@cs.cmu.edu

J. Zico Kolter
Department of Computer Science
Carnegie Mellon University &
Bosch Center for Artificial Intelligence
Pittsburgh, PA
zkolter@cs.cmu.edu

(pause)

Online learning

- Class so far has been in the **(passive) batch setting**:
  - Observe training set $S \sim \mathscr{D}^n$, pick h, test on new examples from $\mathscr{D}$

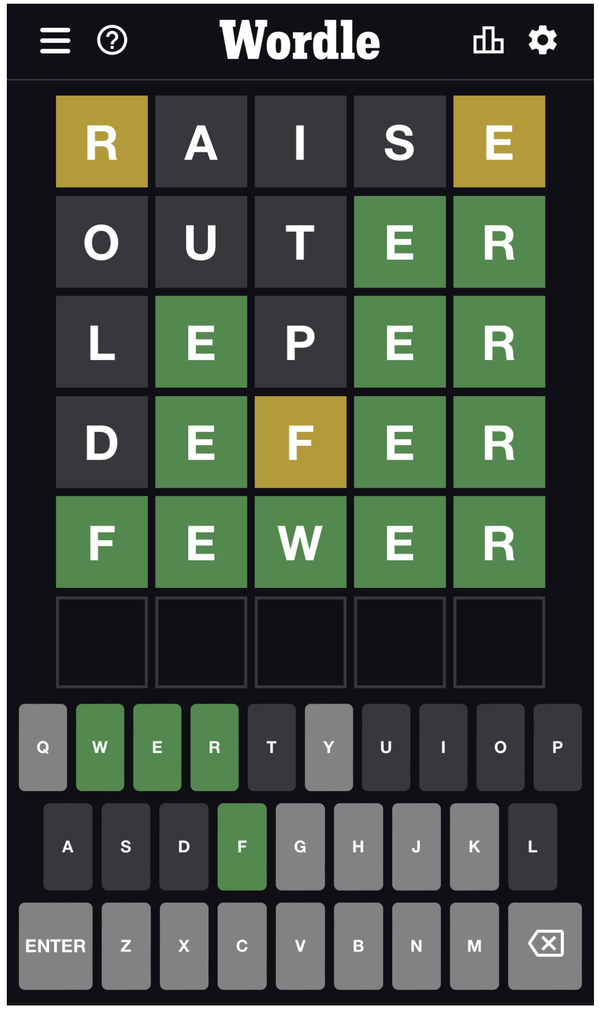
- Today: the **online** setting
  - See an x_t, make a prediction $\hat{y}_t$, see true label y_t, repeat
  - We learn how to predict as we go
  - Focusing on binary classification to start
  - Usual analysis does *not* assume a fixed distribution $\mathscr{D}$
    - Labels can even be chosen **adversarially**

Hello Danica,

I am incredibly sorry about this! It looks like the earlier CMT emails went to my spam folder. I can do this review within the next 12 hours (i.e. by midnight

Realizable online setting

- **Realizable** setting: labels y_t have to be consistent with some $h^* \in \mathscr{H}$



Mistake bounds

- Take a sequence $S = \big((x_1, h^*(x_1)), \ldots, (x_T, h^*(x_T))\big)$
- $M_A(S)$ is the number of **mistakes** the algorithm A makes on S
- $M_A(\mathscr{H})$ is the **worst-case** number of mistakes for **any** S with labels in $\mathscr{H}$
- $\mathscr{H}$ is **online learnable** if there's an A with $M_A(\mathscr{H}) < \infty$

- If $\mathscr{H}$ is finite, consider the algorithm `Consistent` (basically ERM):
  - Start with the **version space** $V_1 = \mathscr{H}$
  - Given x_t, predict $\hat{y}_t = h(x_t)$ for any arbitrary $h \in V_t$
  - Seeing y_t, update $V_{t+1} = \{h \in V_t : h(x_t) = y_t\}$
- Have mistake bound $M_{\texttt{Consistent}}(\mathscr{H}) \leq |\mathscr{H}| - 1$

A smarter algorithm for finite, realizable $\mathscr{H}$

- If `Consistent` made a mistake, we might only remove **one** h from V_t
- Better algorithm can always either (a) be right or (b) make lots of progress
- `Halving`:
  - Start with the version space $V_1 = \mathscr{H}$
  - Given x_t, predict $\hat{y}_t \in \operatorname{argmax}_{r \in \{0,1\}} \left| \{h \in V_t : h(x_t) = r\} \right|$
  - Seeing y_t, update $V_{t+1} = \{h \in V_t : h(x_t) = y_t\}$
- If we were wrong, we removed **at least half** of V_t
- $M_{\texttt{Halving}}(\mathscr{H}) \leq \log_2 |\mathscr{H}|$ – *way* better bound

Online learnability

- Think about the **game tree** for the learner and the **adversary**
  - Put points $x_t \in \mathcal{X}$ into a full binary tree
  - Start at the root, move left if learner predicts 0, right if it predicts 1
- $\mathcal{H}$ **shatters a tree** if everywhere in the tree is reached by some $h \in \mathcal{H}$
- The **Littlestone dimension** $\mathrm{Ldim}(\mathcal{H})$ is the max depth of any tree $\mathcal{H}$ shatters
- Any algorithm A must have $M_A(\mathcal{H}) \geq \mathrm{Ldim}(\mathcal{H})$
- If $\mathcal{H}$ can shatter a set, it can shatter any tree from that set
  - $\mathrm{VCdim}(\mathcal{H}) \leq \mathrm{Ldim}(\mathcal{H})$

Littlestone dimension examples

- If $\mathscr{H}$ is finite, can't shatter a full tree deeper than $\log_2|\mathscr{H}|$
- If $\mathscr{X} = [d]$, $\mathscr{H} = \{x \mapsto \mathbb{I}(x = i) : i \in [d]\}$, have $\mathrm{Ldim}(\mathscr{H}) = 1$
- If $\mathscr{X} = [0,1]$ and $\mathscr{H} = \{x \mapsto \mathbb{I}(x \leq a) : a \in [0,1]\}$, have $\mathrm{Ldim}(\mathscr{H}) = \infty$ (!)

Standard Optimal Algorithm

- Like `Halving`, but tries to reduce Littlestone dimension instead of cardinality:
  - Start with the version space $V_1 = \mathscr{H}$
  - Given x_t, predict $\hat{y}_t \in \operatorname{argmax}_{r \in \{0,1\}} \operatorname{Ldim}\left(\left\{h \in V_t : h(x_t) = r\right\}\right)$
  - Seeing y_t, update $V_{t+1} = \{h \in V_t : h(x_t) = y_t\}$
- Whenever we make a mistake, $\operatorname{Ldim}(V_{t+1}) \leq \operatorname{Ldim}(V_t) - 1$:
  - If not, $\operatorname{Ldim}\left(\left\{h \in V_t : h(x_t) = 0\right\}\right) = \operatorname{Ldim}(V_t) = \operatorname{Ldim}\left(\left\{h \in V_t : h(x_t) = 1\right\}\right)$
  - Then combine shattered trees into one shattered tree of depth $\operatorname{Ldim}(V_t) + 1$
  - But then $\operatorname{Ldim}(V_t) = \operatorname{Ldim}(V_t) + 1$…contradiction
- Thus $M_{\texttt{SOA}}(\mathscr{H}) = \operatorname{Ldim}(\mathscr{H})$, the best possible mistake bound
- But SOA is not necessarily easy to compute!

(pause)

Unrealizable online learning

- In the batch setting:
  - Realizable PAC assumes labels come from $h^* \in \mathscr{H}$
  - Agnostic PAC just has us **compete with** best $h^* \in \mathscr{H}$
- In the online setting:
  - Realizable assumes labels come from $h^* \in \mathscr{H}$
  - Unrealizable has us compete with best $h^* \in \mathscr{H}$

$$\text{Regret}_A(h, T) = \sup_{(x_1,y_1),\ldots,(x_T,y_T)} \left[\sum_{t=1}^{T} |\hat{y}_t - y_t| - \sum_{t=1}^{T} |h(x_t) - y_t| \right]$$

$$\text{Regret}_A(\mathscr{H}, T) = \sup_{h \in \mathscr{H}} \text{Regret}_A(h, T)$$

  - Ideally, we want **sublinear regret**: $\frac{1}{T}\text{Regret}_A(\mathscr{H}, T) \xrightarrow{T\to\infty} 0$

Regret: impossible to avoid

- Regret: “how much better it would have been to just play $h(x_t)$ every time”
- Consider $\mathscr{H} = \{x \mapsto 0, x \mapsto 1\}$
  - Adversary could always just say “no, you’re wrong” and get T mistakes
  - For any sequence of true y_t, either $x \mapsto 0$ or $x \mapsto 1$ has $\leq \frac{T}{2}$ mistakes
  - So regret would be at least $T - \frac{T}{2} = \frac{T}{2}$
- To avoid this:
  - Learner has **random** prediction, $\Pr(\hat{y}_t = 1) = p_t$
  - Adversary commits to y_t without knowing the roll
  - Measure **expected** loss $\Pr(\hat{y}_t \neq y_t) = |p_t - y_t|$

Low regret for online classification

- For every $\mathscr{H}$, there's an algorithm with

$$\text{Regret}_A(\mathscr{H}, T) \leq \sqrt{2 \min\left(\log|\mathscr{H}|, (1 + \log T)\ \text{Ldim}(\mathscr{H})\right) T}$$

- Also a lower bound of $\Omega\left(\sqrt{\text{Ldim}(\mathscr{H})\ T}\right)$

- Based on Weighted-Majority algorithm for **learning with expert advice**

Learning from expert advice

- There are d available experts who make predictions
- At time t, learner chooses to follow expert i with probability $(w_t)_i$
- Sees potential costs $v_t \in \mathbb{R}^d$; pays expectation $\langle w_t, v_t \rangle$
- `Weighted-Majority` algorithm:
  - Start with $\tilde{w}_1 = (1,\ldots,1);\ \eta = \sqrt{2\log(d)/T}$
  - For $t = 1,2,\ldots$
    - Follow with probabilities $w_t = \tilde{w}_t / \|w_t\|_1$
    - Update based on costs v_t as $\tilde{w}_{t+1} = \tilde{w}_t \exp(-\eta v_t)$ (exp is elementwise)
- **Theorem** (SSBD 21.11): $\sum_{t=1}^{T} \langle w_t, v_t \rangle - \min_{i\in[d]} \sum_{t=1}^{T} (v_t)_i \le \sqrt{2\log(d)\,T}$ if $T > 2\log d$
- Can avoid knowing T by *doubling trick*: run for $T = 1$, $T = 2$, $T = 4$, … sequentially
  - Only blows up regret by < 3.5x (SSBD exercise 21.4)

Low regret for online classification

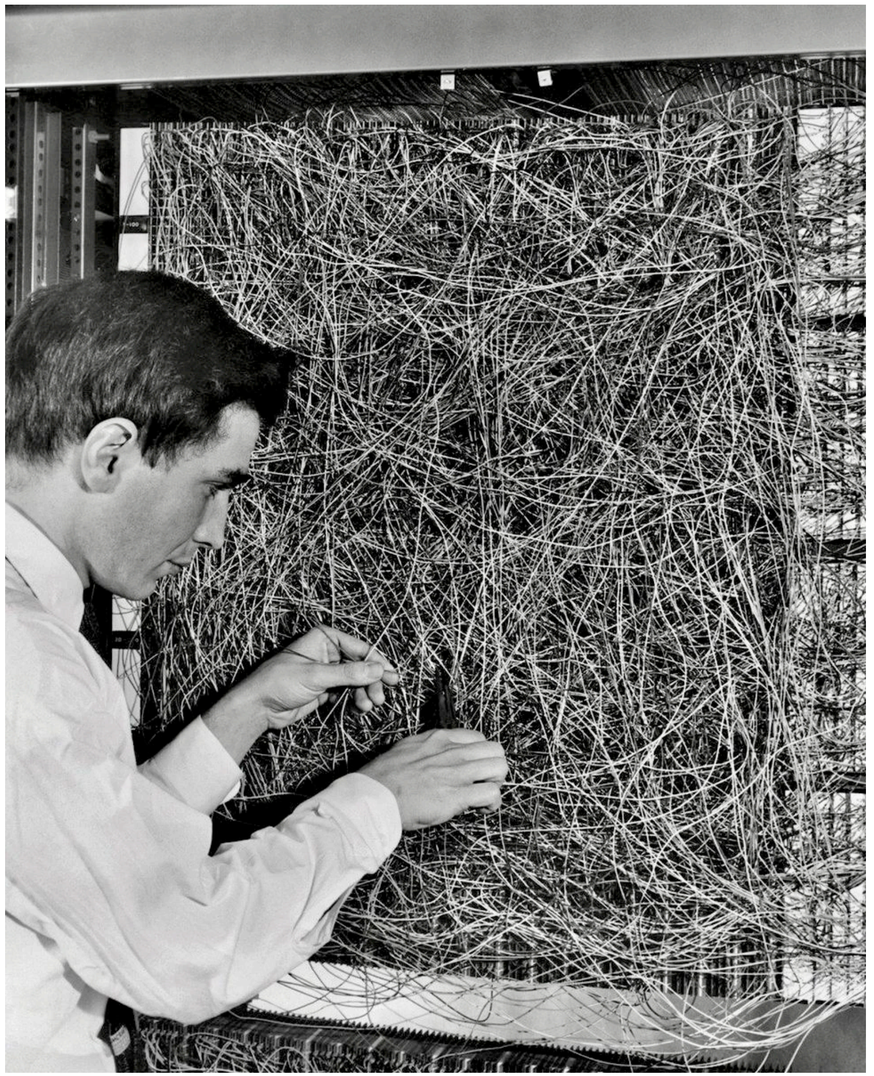
- For finite $\mathscr{H}$, we can just run `Weighted-Majority` with each $h \in \mathscr{H}$
  - Plugging in previous theorem, $\text{Regret}_{\texttt{WM}}(\mathscr{H}, T) \leq \sqrt{2 \log |\mathscr{H}| \; T}$
- For infinite $\mathscr{H}$, we need a not-too-big set of experts where one is still good
  - Expert($i_1, i_2, \ldots, i_L$) runs SOA on $x_1, \ldots, x_T$,
    but takes choice with smaller Ldim on indices $i_1, i_2, \ldots, i_L$
  - Can show (21.13-14) that one expert is as good as the best $h \in \mathscr{H}$,
    and there aren't too many of them,
    giving $\text{Regret}_A(\mathscr{H}, T) \leq \sqrt{2(1 + \log T) \; \text{Ldim}(\mathscr{H}) \; T}$

Online convex optimization

- **Online convex optimization** is
  - Convex hypothesis class $\mathscr{H}$
  - At each step: learner picks $w_t \in \mathscr{H}$, environment picks convex loss $\ell_t(w_t)$
- $$\text{Regret}(w^*, T) = \sum_{t=1}^{T} \ell_t(w_t) - \sum_{t=1}^{T} \ell_t(w^*), \quad \text{Regret}(\mathscr{H}, T) = \sup_{w^* \in \mathscr{H}} \text{Regret}(w^*, T)$$
- **Online gradient descent** (exactly like SGD) has:
  - $\text{Regret}(w^*, T) \leq \frac{\|w^*\|^2}{2\eta} + \frac{\eta}{2} \sum_{t=1}^{T} \|v_t\|^2$ where $v_t \in \partial \ell_t(w_t)$ are step directions
  - $\text{Regret}(w^*, T) \leq \frac{1}{2} \left(\|w^*\|^2 + \rho^2 \right) \sqrt{T}$ if ℓ_t are ρ-Lipschitz, $\eta = 1/\sqrt{T}$
  - $\text{Regret}(w^*, T) \leq B\rho\sqrt{T}$ if ℓ_t are ρ-Lipschitz, $\mathscr{H}$ is B-bounded, $\eta = B/\left(\rho\sqrt{T}\right)$

Online Perceptron

- Perceptron algorithm: constant-learning-rate online gradient descent on hinge loss of linear classifier
- Get $(R/\gamma)^2$ margin-based mistake bound
  - Ldim = ∞ without the margin condition



NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI) —The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's $2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of $100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human beings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.

Without Human Controls

The Navy said the perceptron would be the first non-living mechanism "capable of receiving, recognizing and identifying its surroundings without any human training or control."

The "brain" is designed to remember images and information it has perceived itself. Ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech or writing in another language, it was predicted.

Mr. Rosenblatt said in principle it would be possible to build brains that could reproduce themselves on an assembly line and which would be conscious of their existence.

In today's demonstration, the "704" was fed two cards, one with squares marked on the left side and the other with squares on the right side.

Learns by Doing

In the first fifty trials, the machine made no distinction between them. It then started registering a "Q" for the left squares and "O" for the right squares.

Dr. Rosenblatt said he could explain why the machine learned only in highly technical terms. But he said the computer had undergone a "self-induced change in the wiring diagram."

The first Perceptron will have about 1,000 electronic "association cells" receiving electrical impulses from an eye-like scanning device with 400 photo-cells. The human brain has 10,000,000,000 responsive cells, including 100,000,000 connections with the eyes.

Online-to-batch conversion

- If we have a good online algorithm, we have a good batch algorithm: just run it on the batch
- MRT **Lemma** 8.14: If $S \sim \mathscr{D}^T$ gives $h_1, \ldots, h_T$ for $0 \leq \ell(h, (x, y)) \leq M$,

$$\frac{1}{T}\sum_{t=1}^{T} L_{\mathscr{D}}(h_t) \leq \frac{1}{T}\sum_{t=1}^{T} \ell(h_t(x_t), y_t) + M\sqrt{\frac{2}{T}\log\frac{1}{\delta}}$$

- MRT **Theorem** 8.15: if $\ell(\cdot, z)$ is also convex,

$$L_{\mathscr{D}}\left(\frac{1}{T}\sum_{t=1}^{T} h_t\right) \leq \inf_{h \in \mathscr{H}} L_{\mathscr{D}}(h) + \frac{1}{T}\mathrm{Regret}_A(\mathscr{H}, T) + 2M\sqrt{\frac{2}{T}\log\frac{2}{\delta}}$$

(pause)

Differential privacy

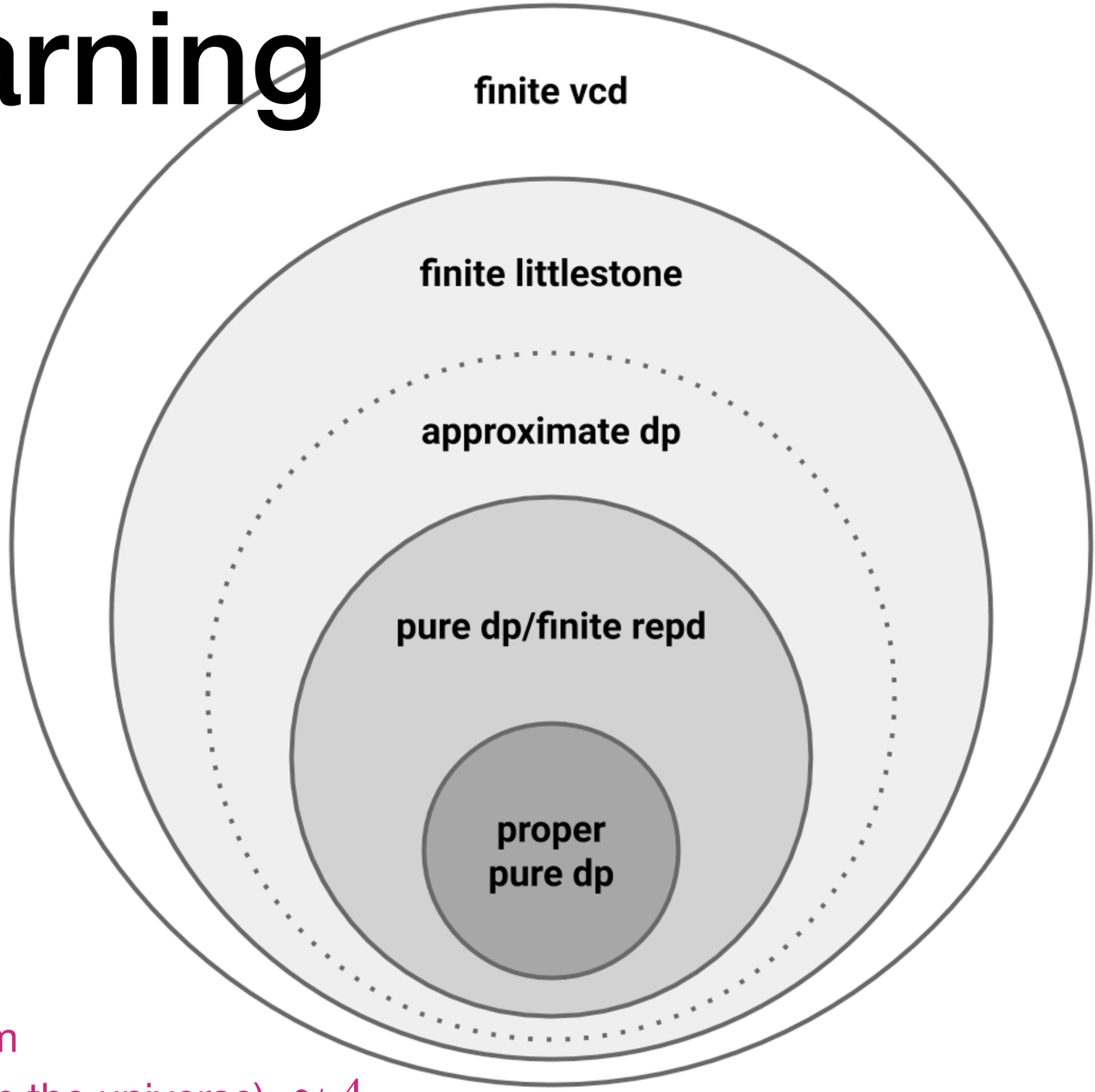
- Randomized learning algorithm $A(S)$ is called **(ε, δ) differentially private** if
  - for all S_1, S_2 that differ on a single element (i.e. one person's data),
  - for all subsets $H \subseteq \mathscr{H}$, $\Pr(A(S_1) \in H) \leq \exp(\varepsilon) \Pr(A(S_2) \in H) + \delta$
- Called **pure** DP if $\delta = 0$

- Used in practice (US Census, Apple, …), tons of work on algorithms
  - Mijung Park and Mathias Lecuyer both work on this, teach courses (532P next fall, 538L now [but not next year])

- Can be thought of as a particular **form of stability**

DP and online learning

- Feldman and Xiao 2014:
  Pure private PAC learning takes
  $\Omega(\mathrm{Ldim}(\mathscr{H}))$ samples
  - Related to communication complexity
- Alon, Livni, Malliaris, Moran 2019:
  Approximate private PAC learning takes
  $\Omega(\log^*(\mathrm{Ldim}(\mathscr{H})))$ samples
- Bun, Livni, Moran 2020:
  Approximate private PAC learning in
  $2^{\mathscr{O}(\mathrm{Ldim}(\mathscr{H}))}$ samples
  - analysis via “global stability”



$\log^*$ = iterated logarithm

$\log^*$(number of atoms in the universe) ≈ 4

https://differentialprivacy.org/private-pac/

DP and online learning

- Can learn differentially privately iff can learn online
  - Close connections via stability
  - But huge gap in sample and time complexity
  - Indications (Bun 2020) that converting one to the other isn't possible with polynomial time + sample complexity
  - Still a lot to understand here

Some of the stuff we didn't cover

- **Multiclass learning**: can use same techniques, need right loss
- **Ranking**: which search results are most relevant?
- **Boosting**: combine "weak learners" to a strong one
- **Transfer learning** / **out-of-domain generalization** / …: train on $\mathscr{D}$, test on $\mathscr{D}'$
- Do ImageNet Classifiers Generalize to ImageNet? / The Ladder mechanism
- **Robustness**: what if we have some adversarially-corrupted training data?
- **Unsupervised learning**: "How well can we 'understand' a data distribution?"
- **Semi-supervised learning**
- **Active learning**: if xs are available but labeling them is expensive, can we choose which to label?
- **Multi-armed bandits**: which action should I take?
- **Reinforcement learning**: interacting with an environment with hidden state
- …