PAC learning
+ uniform convergence

CPSC 532S: Modern Statistical Learning Theory

11 January 2022

cs.ubc.ca/~dsuth/532S/22/

Admin

- Now online until (at least) **February 7**

- A1 is up; get at it!
  - Due 11:59pm Thursday the 20th; do alone
  - Should be able to do all of it after today
  - Might require brushing up on linear algebra a bit
  - Submission instructions coming by this weekend

- We're making progress towards fitting in the cap :)
  - If you're pretty sure you'll drop, please don't wait until the last day, so people on the waitlist can plan appropriately
  - (But also, please don't drop if you want to stay!)

Last time: definitions

- $x \sim \mathscr{D}_x$, a distribution over $\mathscr{X}$; $y = f(x) \in \mathscr{Y}$; $S = \big((x_1, y_1), \ldots, (x_n, y_n)\big)$
- Want $h : \mathscr{X} \to \mathscr{Y}$ minimizing $L_{\mathscr{D}_x, f}(h) = \Pr_{x \sim \mathscr{D}_x}\big(h(x) \neq f(x)\big)$
- Training loss $L_S(h) = \frac{1}{n} \sum_{i=1}^{n} \begin{cases} 1 & \text{if } h(x_i) \neq y_i \\ 0 & \text{if } h(x_i) = y_i \end{cases}$

- Empirical risk minimization (ERM): choose h minimizing $L_s(h)$ from a ***hypothesis class*** $\mathscr{H}$ of functions $h : \mathscr{X} \to \mathscr{Y}$

- To start with something simple, assume **realizability**:

  there is an $h^* \in \mathscr{H}$ with $L_{\mathscr{D}_x, f}(h^*) = 0$

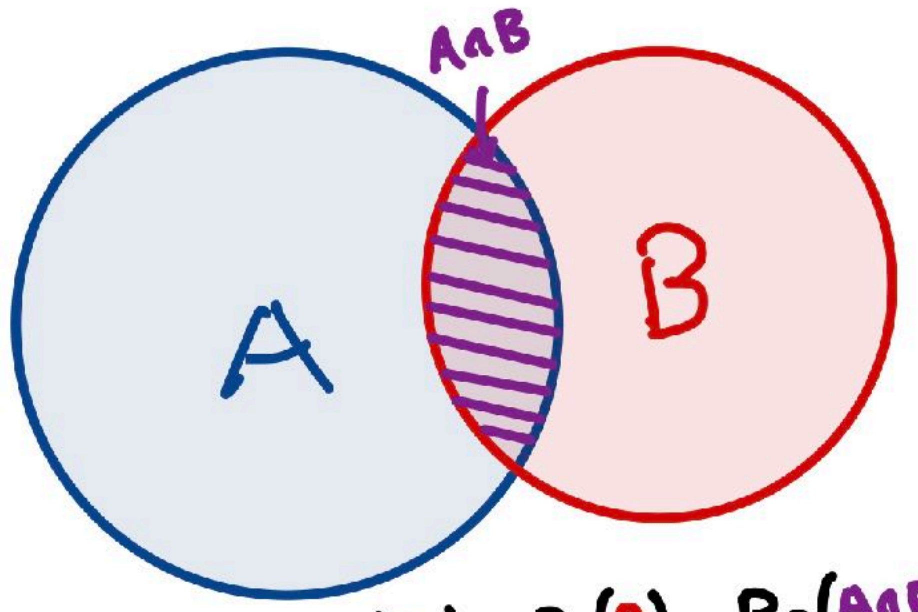
  - Implies (a.s.) that $L_S(h^*) = 0$

Realizable, finite $\mathscr{H}$

- $h_S \in \arg\min_{h \in \mathscr{H}} L_S(h)$: realizable means $L_S(h_S) = 0$, but maybe $L_{\mathscr{D}_x, f}(h_S) > 0$
- Would like to show $\Pr_S \left(L_{\mathscr{D}_x, f}(h_S) \le \varepsilon \right) \ge 1 - \delta$, i.e. $\Pr(L(h_S) > \varepsilon) < \delta$
- Call $\mathscr{H}_B$ the set of “bad” hypotheses, $\left\{ h \in \mathscr{H} : L_{\mathscr{D}_x, f}(h) > \varepsilon \right\}$
- $M = \left\{ S : \exists h \in \mathscr{H}_B .\ L_S(h) = 0 \right\}$ is set of “bad” samples
  - If $L_{\mathscr{D}_x, f}(h_S) > \varepsilon$, then $S \in M$
- For a “worst-case ERM”, we have

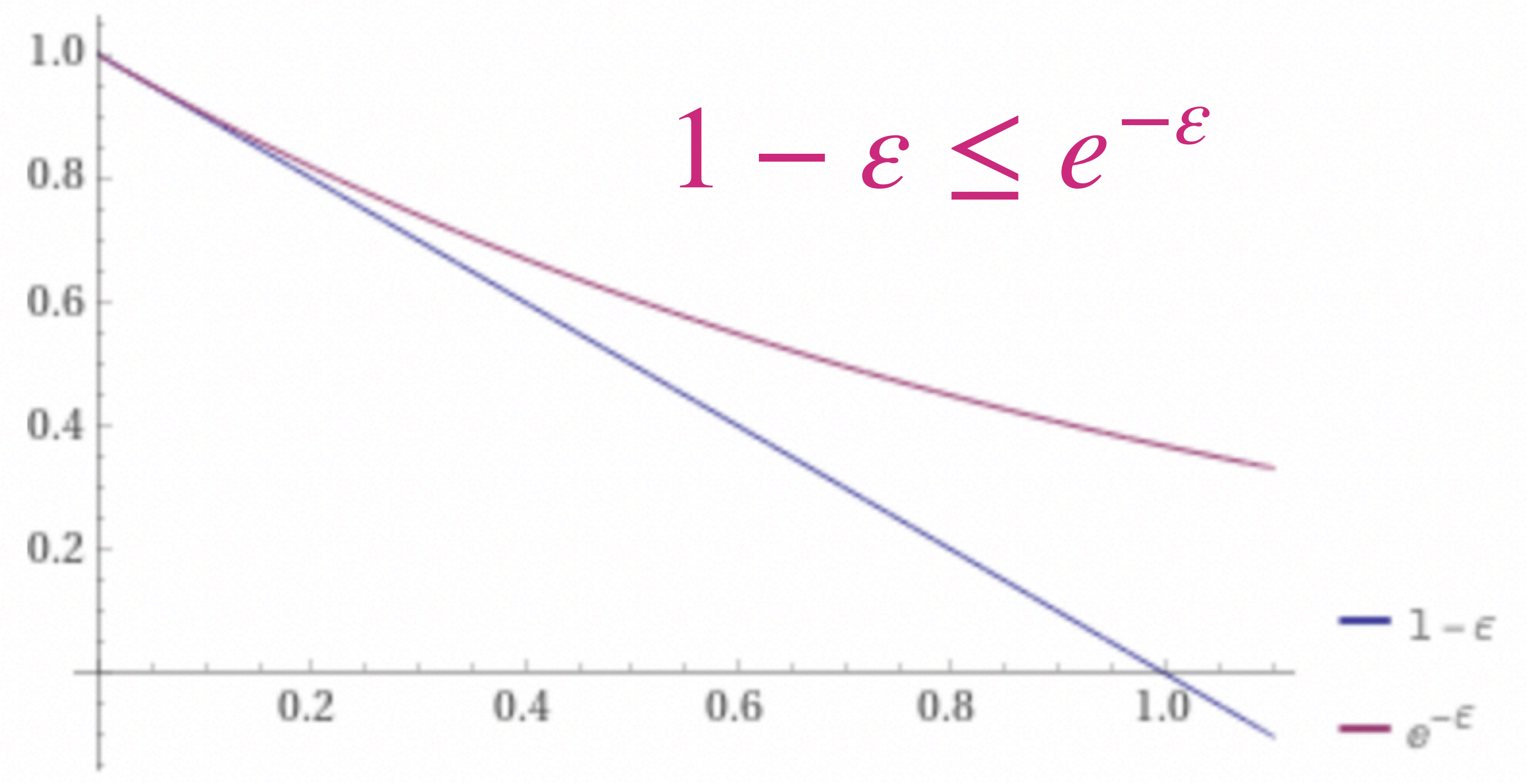
$$\Pr(L(h_S) > \varepsilon) = \mathscr{D}_x^n(M) = \mathscr{D}_x^n \left(\bigcup_{h \in \mathscr{H}_B} \{ S : L_S(h) = 0 \} \right) \le \sum_{h \in \mathscr{H}_B} \mathscr{D}_x^n \left(\{ S : L_S(h) = 0 \} \right)$$

Union bound



$\Pr(A \cup B) = \Pr(A) + \Pr(B) - \Pr(A \cap B)$
$\le \Pr(A) + \Pr(B)$

Realizable, finite $\mathscr{H}$



$1 - \varepsilon \leq e^{-\varepsilon}$

- $\Pr(L(h_S) > \varepsilon) \leq \sum_{h \in \mathscr{H}_B} \mathscr{D}_x^n \left(\{S : L_S(h) = 0\} \right)$
- $\mathscr{D}_x^n(\{S : L_S(h) = 0\}) = \mathscr{D}_x^n(\{S : \forall i,\, h(x_i) = f(x_i)\})$
- Because it's iid, this is just $\prod_{i=1}^{n} \mathscr{D}_x(\{x_i : h(x_i) = f(x_i)\})$

If a hypothesis is bad,
we're likely to sample
at least one x_i where it's wrong

- But $\mathscr{D}_x(\{x_i : h(x_i) = y_i\}) = 1 - L_{\mathscr{D}_x, f}(h) < 1 - \varepsilon$ since $h \in \mathscr{H}_B$
- $\Pr(L(h_S) > \varepsilon) < \sum_{h \in \mathscr{H}_B} (1 - \varepsilon)^n$

Not too likely to get unlucky
with *any* bad hypothesis

$$\leq |\mathscr{H}_B|(1-\varepsilon)^n \leq |\mathscr{H}|(1-\varepsilon)^n \leq |\mathscr{H}|e^{-\varepsilon n}$$

Finite $\mathcal{H}$ are (realizable) PAC-learnable

- We showed that $\Pr\left(L_{\mathcal{D}_x,f}(h_S) < \varepsilon\right) \geq 1 - |\mathcal{H}|e^{-\varepsilon n}$
- Or: if we have $n \geq \frac{1}{\varepsilon}\left(\log|\mathcal{H}| + \log\frac{1}{\delta}\right)$, $L_{\mathcal{D}_x,f}(h) \leq \varepsilon$ with prob. at least $1 - \delta$.
- Or: error is at most $\frac{1}{n}\left(\log|\mathcal{H}| + \log\frac{1}{\delta}\right)$ with high probability

- $\mathcal{H}$ is **PAC learnable** if there is a function $n_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ and a learning alg. s.t.:
  - For every $\varepsilon, \delta \in (0,1)$, for every $\mathcal{D}_x$ over $\mathcal{X}$, and every labeler $f : \mathcal{X} \to \{0,1\}$:
  - If $\mathcal{H}$ is realizable for $\mathcal{D}_x$ and f,
  - then running the algorithm on $n \geq n_{\mathcal{H}}(\varepsilon, \delta)$ i.i.d. examples from $\mathcal{D}_x$ labeled by f,
  - will return a hypothesis h with $L_{\mathcal{D}_x,f}(h) \leq \varepsilon$
  - with probability at least $1 - \delta$ over the choice of examples

Example: Boolean conjunctions

a	b	c	d	e	f	y
0	1	1	0	1	1	+
0	0	1	0	0	1	+
0	1	1	1	1	1	-
1	1	1	0	1	1	+
0	1	0	0	1	0	-
1	0	1	0	0	0	-
1	1	1	1	0	1	?

$\mathscr{H}$: conjunctions of the form

$$a \wedge \bar{c} \wedge f$$

Algorithm:

- Start with $a \wedge \bar{a} \wedge \cdots \wedge f \wedge \bar{f}$
- Cross out bits inconsistent with the positives

Example: Boolean conjunctions

a	b	c	d	e	f	y
0	1	1	0	1	1	+
0	0	1	0	0	1	+
0	1	1	1	1	1	-
1	1	1	0	1	1	+
0	1	0	0	1	0	-
1	0	1	0	0	0	-
1	1	1	1	0	1	?

$\mathscr{H}$: conjunctions of the form

$$a \wedge \bar{c} \wedge f$$

Algorithm:

- Start with $a \wedge \bar{a} \wedge \cdots \wedge f \wedge \bar{f}$
- Cross out bits inconsistent with the positives

Example: Boolean conjunctions

a	b	c	d	e	f	y
0	1	1	0	1	1	+
0	0	1	0	0	1	+
0	1	1	1	1	1	-
1	1	1	0	1	1	+
0	1	0	0	1	0	-
1	0	1	0	0	0	-
1	1	1	1	0	1	?

$\mathscr{H}$: conjunctions of the form

$$a \wedge \bar{c} \wedge f$$

Algorithm:

- Start with $a \wedge \bar{a} \wedge \cdots \wedge f \wedge \bar{f}$
- Cross out bits inconsistent with the positives

Example: Boolean conjunctions

a	b	c	d	e	f	y
0	1	1	0	1	1	+
0	0	1	0	0	1	+
0	1	1	1	1	1	-
1	1	1	0	1	1	+
0	1	0	0	1	0	-
1	0	1	0	0	0	-
1	1	1	1	0	1	?

$\mathscr{H}$: conjunctions of the form

$$a \wedge \bar{c} \wedge f$$

Algorithm:

- Start with $a \wedge \bar{a} \wedge \cdots \wedge f \wedge \bar{f}$
- Cross out bits inconsistent with the positives

Example: Boolean conjunctions

$c \wedge \bar{d} \wedge f$

a	b	c	d	e	f	y
0	1	1	0	1	1	+
0	0	1	0	0	1	+
0	1	1	1	1	1	-
1	1	1	0	1	1	+
0	1	0	0	1	0	-
1	0	1	0	0	0	-
1	1	1	1	0	1	?

$|\mathscr{H}| = 3^d$: $\left\lceil \frac{1}{\varepsilon}\left(d \log(3) + \log \frac{1}{\delta}\right)\right\rceil$ samples enough

$\mathscr{H}$: conjunctions of the form

$a \wedge \bar{c} \wedge f$

Algorithm:

- Start with $a \wedge \bar{a} \wedge \cdots \wedge f \wedge \bar{f}$
- Cross out bits inconsistent with the positives

Assuming realizability, this gives an ERM

- Algorithm makes every + example a +
- True function f is only “less specific” than h: h(x) = - for anything truly -

So, are we done?

- Every practical $\mathcal{H}$ is finite if you put it on a computer
- Total size of weights in a big deep network is typically up to ~1GB
- Say 100MB, $8 * 100 * 2^{20}$ bits, so there are $2^{25 \cdot 2^{25}}$ possible networks
- $\log\left(2^{25 \cdot 2^{25}}\right) = 252^{25} \log(2) \approx$ 252 million
  - If we want, say, $\varepsilon = 0.1$ (90% accuracy): 2.5 billion training points

- (Plus, we don't actually do ERM with realizable, fixed hypothesis classes…)

PAC learnability and computational efficiency


RESEARCH CONTRIBUTIONS

Artificial Intelligence and Language Processing

David Waltz
Editor

A Theory of the Learnable

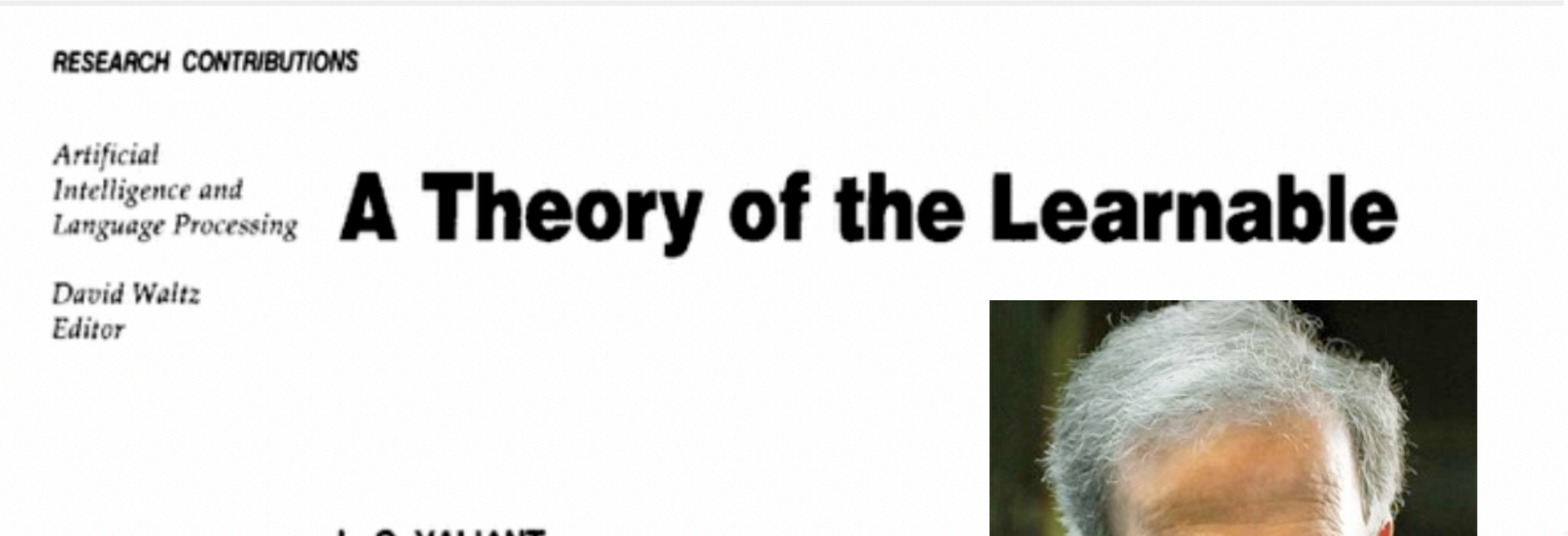
L. G. VALIANT

Communications of the ACM, 1984

- Valiant (1984)'s formulation
  required the algorithm
  to run in polynomial time
- We're going to think about runtime separately,
  but be aware many authors keep that in the definition

- Independent(?), closely related development by Vapnik and Chervonenkis
  in the USSR; much more on their work soon

PAC learnability and computational efficiency


RESEARCH CONTRIBUTIONS

Artificial Intelligence and Language Processing

David Waltz Editor

A Theory of the Learnable

L. G. VALIANT

Communications of the ACM, 1984

- Valiant (1984)'s formulation
  required the algorithm
  to run in polynomial time
- We're going to think about runtime separately,
  but be aware many authors keep that in the definition

- Independent(?), closely related development by Vapnik and Chervonenkis
  in the USSR; much more on their work soon

PAC learnability and computational efficiency

- A class that can be PAC-learned but **not in polynomial time** (assuming P = BPP and P ≠ NP):
- 3-DNF: 3-term clauses in *disjunctive normal form*

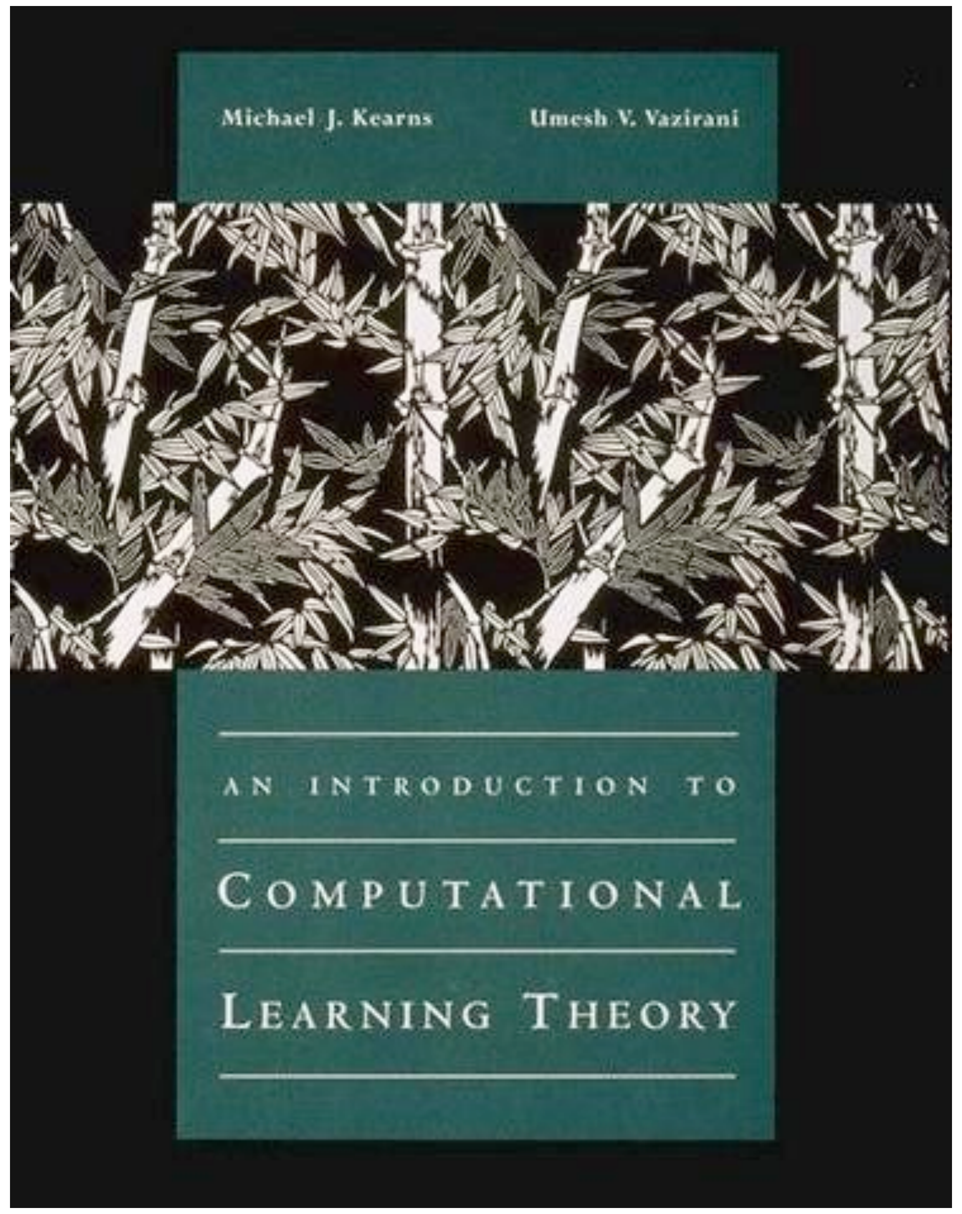
  $T_1 \vee T_2 \vee T_3$

  terms are conjunctions: $T_1 = a \wedge \bar{c} \wedge \cdots$

  - Graph 3-coloring reduces to learning 3-DNFs

- But: 3-DNF ⊂ 3-CNF, $\bigwedge (a \vee b \vee c)$,
- $T_1 \vee T_2 \vee T_3 = \bigwedge_{u \in T_1, v \in T_2, w \in T_3} (u \vee v \vee w)$
- and 3-CNF **can** be efficiently PAC-learned

(Sec 1.4-1.5
PDF through UBC: log in here)



Computational Limitations on Learning from Examples

(1988)

LEONARD PITT

University of Illinois, Urbana-Champaign, Urbana, Illinois

AND

LESLIE G. VALIANT

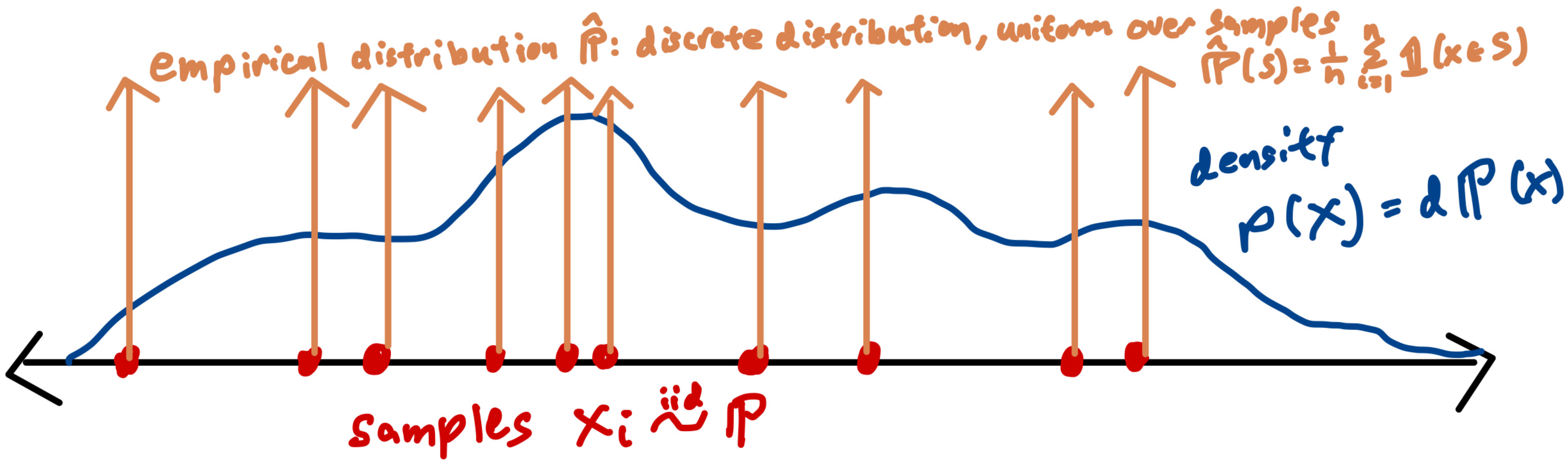
Harvard University, Cambridge, Massachusetts

(pause)

Non-realizable (agnostic) learning

- What if there's some noise in the data?
  - e.g. two identical xs might have different ys
- Instead of saying $x \sim \mathcal{D}_x$ and $y = f(x)$, have **joint distribution** $(x, y) \sim \mathcal{D}$
- $\mathcal{D}$ is a distribution over domain $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$
- Population loss is now $L_{\mathcal{D}}(h) = \Pr(h(x) \neq y) = \mathcal{D}\left(\left\{(x, y) : h(x) \neq y\right\}\right)$
- Empirical loss still $L_S(h) = \frac{1}{n} \sum_{i=1}^{n} \begin{cases} 1 & \text{if } h(x_i) \neq y_i \\ 0 & \text{if } h(x_i) = y_i \end{cases}$
  - Notice this is the population loss over the *empirical distribution* on S

Non-realizable (agnostic) learning



- Population loss is now $L_{\mathscr{D}}(h) = \Pr(h(x) \neq y) = \mathscr{D}\left(\{(x, y) : h(x) \neq y\}\right)$
- Empirical loss still $L_S(h) = \frac{1}{n} \sum_{i=1}^{n} \begin{cases} 1 & \text{if } h(x_i) \neq y_i \\ 0 & \text{if } h(x_i) = y_i \end{cases}$
    - Notice this is the population loss over the *empirical distribution* on S

General loss functions

- So far we've only looked at the error rate
- More generally, allow a loss function $\ell : \mathscr{H} \times \mathscr{Z} \to \mathbb{R}$

$$L_{\mathscr{D}}(h) = \mathbb{E}_{z \sim \mathscr{D}}[\ell(h, z)] \qquad\qquad L_S(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h, z_i)$$

- 0-1 loss: $\mathscr{Z} = \mathscr{X} \times \mathscr{Y}$, $\ell_{0-1}(h, (x, y)) = \begin{cases} 0 & \text{if } h(x) = y \\ 1 & \text{if } h(x) \neq y \end{cases}$

  gives classification error rate

- Square loss ($\mathscr{Y} \subseteq \mathbb{R}$) is $\ell_{\text{sq}}(h, (x, y)) = (h(x) - y)^2$

- Tons of other options!

Agnostic PAC

- $\mathcal{H}$ is **agnostically PAC learnable** for a set $\mathcal{Z}$ and loss $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}$ if there is a function $n_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ and a learning algorithm such that:
  For every $\varepsilon, \delta \in (0,1)$ and every distribution $\mathcal{D}$ over $\mathcal{Z}$,
  then running the algorithm on $n \geq n_{\mathcal{H}}(\varepsilon, \delta)$ i.i.d. examples from $\mathcal{D}$
  will return a hypothesis $h \in \mathcal{H}$ with $L_{\mathcal{D}}(h) \leq \inf_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \varepsilon$
  with probability at least $1 - \delta$ over the choice of examples
- We don't (necessarily) get error arbitrarily close to 0 anymore!
  - Realizable means $\inf_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') = 0$: then, this is same as realizable PAC
  - Otherwise, $\inf_{h' \in \mathcal{H}} L_{\mathcal{D}}(h')$ is the best loss achievable in $\mathcal{H}$

Improper Agnostic PAC

- $\mathcal{H}$ is **improperly agnostically PAC learnable** in $\mathcal{H}'$ for $\mathcal{Z}$, loss $\ell : \mathcal{H}' \times \mathcal{Z} \to \mathbb{R}$ if there is a function $n_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ and a learning algorithm such that:
  For every $\varepsilon, \delta \in (0,1)$ and every distribution $\mathcal{D}$ over $\mathcal{Z}$,
  then running the algorithm on $n \geq n_{\mathcal{H}}(\varepsilon, \delta)$ i.i.d. examples from $\mathcal{D}$
  will return a hypothesis $h \in \mathcal{H}' \supset \mathcal{H}$ with $L_{\mathcal{D}}(h) \leq \inf_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \varepsilon$
  with probability at least $1 - \delta$ over the choice of examples

- e.g.: learn a polynomial classifier almost as good as the best linear classifier,
  or learn a 3-DNF function with a 3-CNF

- Shai+Shai: “there is nothing improper about representation-independent learning”

Bayes error rate

- What can we say about $\inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$?
- It's at least as big as the **Bayes error**: error of the Bayes-optimal classifier

$$f_{\mathcal{D}}(x) = \begin{cases} 1 & \text{if } \Pr(y = 1 \mid x) \geq \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}$$

- This is the best conceivable classifier. (See homework!)
  The best classifier in $\mathcal{H}$ might be this good, or it might be worse
- Other losses have corresponding Bayes-optimal predictors;
  for reasonable classification losses, it's this same $f_{\mathcal{D}}$.

Summary

- PAC learnability: realizable, agnostic, improper
    - Finite classes: realizable PAC by ERM with $n \geq \frac{1}{\varepsilon}\left(\log|\mathscr{H}| + \log\frac{1}{\delta}\right)$
- Extended definition to general loss functions on $\mathscr{Z}$, e.g. $\mathscr{X} \times \mathscr{Y}$
- Bayes classifier / Bayes error rate

- Next time: finite classes in the agnostic case + uniform convergence