

Intro / overview

CPSC 532S: Modern Statistical Learning Theory

10 January 2022

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

Admin: recordings

- [reminder to start recording]
- As you know, we're online for (at least) two weeks
- I'm going to try recording only the slides, no cameras
 - ...so you can have your camera on and not be on the recording
 - Your voice will be on the recording if you speak
 - Your public chats will also be saved; feel free to use an alias or DM me
- Recordings won't be public
 - Will be shared with a few people not officially in the course
 - Posted on Canvas Zoom tab and linked from Piazza
- Will try to watch chat and Zoom "raise hand"s; interrupt me if I miss it

Admin: teaching modality

- Once we're allowed to, I will be in person in DMP 101
- FYI: 40-seat room, not very spaced
- As of Sunday afternoon, there are ~70 people registered / waitlisted / unofficially on waitlist / wanting to unofficially sit in
- Plan to livestream and record lectures (Panopto or Zoom tbd)
- There most likely will be no required in-person activities for this course. But not 100% decided on that: message me if you're making major decisions (e.g. whether to come to Vancouver) affected by that decision.

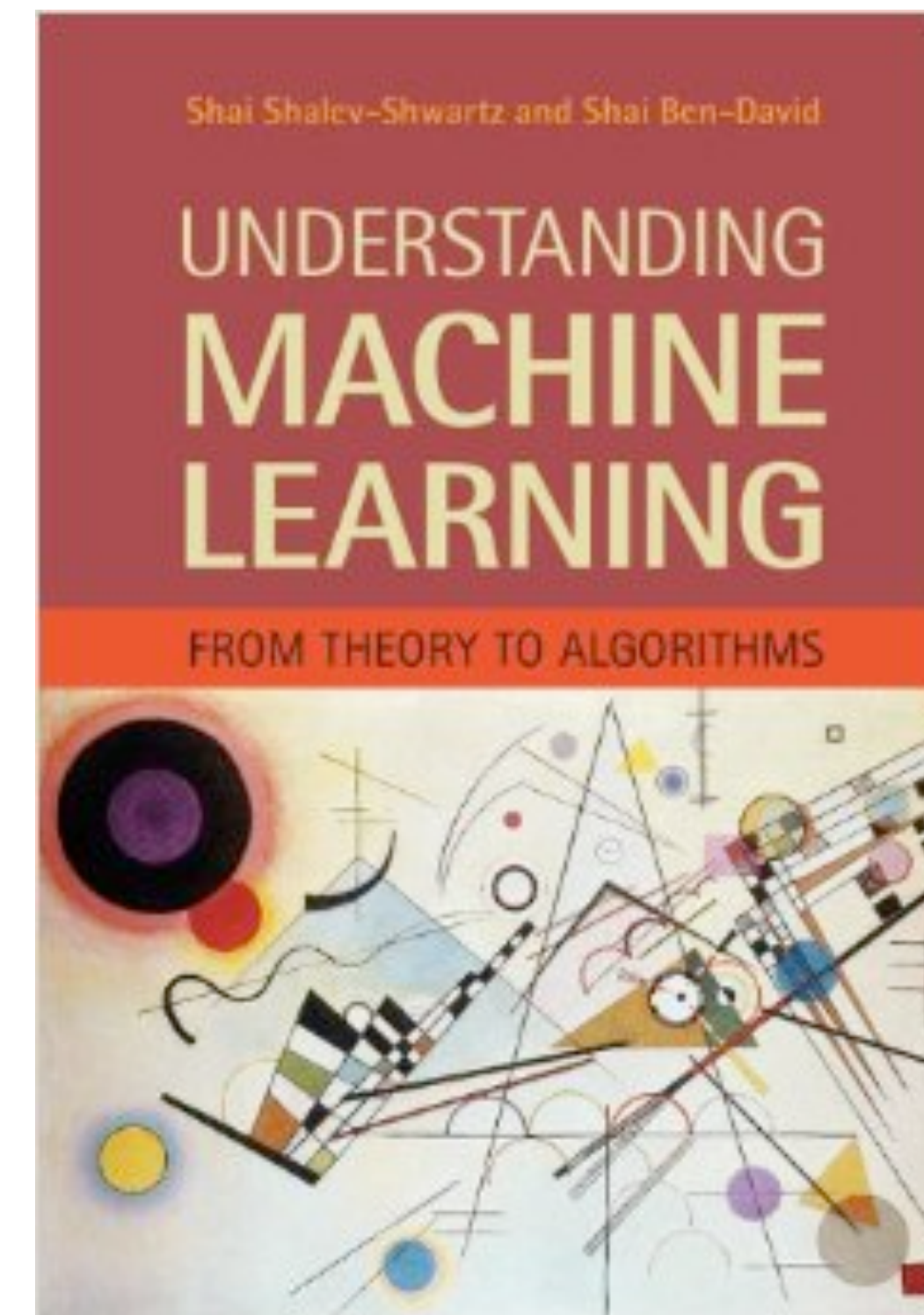


Admin: course cap

- 40 person cap
- As of Sunday afternoon, there are ~70 people registered / waitlisted / unofficially on waitlist / wanting to unofficially sit in
- Hoping we'll end up at ≤ 40 officially taking the course
 - Others totally welcome to sit in on Zoom / be on Piazza / etc
 - When we go in-person, seat priority to people registered
- If more than that, we'll see what happens...
- I may be able to expand the course cap above 40 for auditors only
- **I will probably teach this again next year** (not sure if T1 or T2 yet)

Admin: resources

- Course website: cs.ubc.ca/~dsuth/532S/
 - Slides, schedule, homeworks, etc
- Canvas (registered people only): canvas.ubc.ca/courses/83445
 - Zoom recordings, grades
- Piazza: piazza.com/ubc.ca/winterterm22022/cpsc532s
 - Discussion, also links to Zoom recordings
 - Prefer you post anything course-related here, but email is okay if it's easier for whatever reason
- First chunk of the course will roughly follow Shai+Shai
 - [Free PDF version](#) for personal use
- Won't cover all of it; will cover a bunch of stuff not in it
- Course site will link relevant (optional) readings throughout
- Other generally relevant books + lecture notes on course site



Admin: course format

- The first part of this course will be lecture-based.
- Depending on COVID situation / etc, *may* have some more discussion-oriented chunks later in term. Will let you know what the plan is.
- Grading: split TBD between
 - Assignments: several through the term
 - Including one small project / “big assignment”
 - Do some experiments exploring a paper, lit review, extend / unify papers, etc. Proposal beforehand; details to come.
 - Presentations in discussion time, if these happen
 - Might count for part of an assignment
 - Final exam – probably take-home

Admin: assignments

- First assignment will be up tonight on the course site
- Handin procedure TBA (soon)
- **Due Thursday the 20th (next week), 11:59pm**
- Do it in LaTeX; template available to fill in if that helps, or go from scratch
- Large parts you should be able to do already
- Some bits we're covering in class this week
- Assignment 1: do all of it, by yourself; cite any sources you use
- Later assignments will allow (+ encourage) group work
 - Might ask for only a subset of problems; maybe randomized/peer grading
- If you're not yet registered but want in, do the assignment
- If you're auditing/sitting in: encourage you to do it but *don't submit*, please

Admin: me (hi!)

- Danica Sutherland - <https://djsutherland.ml/> - ICICS X563 - she/her
 - “Danica” (North Am. English pronunciation, not authentic Slavic one)
/ “Professor Sutherland” / “Dr. Sutherland” are all fine
- New-ish at UBC (here 1/1.5 years depending on what you count...covid times)
 - 6 grad students (5 of you here, hi)
 - 2019-20: TTI-Chicago
 - 2016-19: University College London
 - 2011-16: Carnegie Mellon

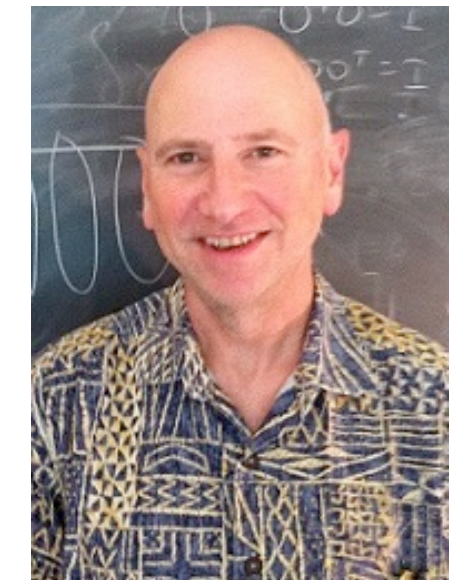
Admin: me (hi!)

- My research so far:
 - learning and testing on probability distributions (~80% of work)
 - kernels: especially “deep kernels,” especially on distributions (~80% of work)
 - various other stuff about representation learning (~20% of work)
 - statistical learning theory:
 - theorems about kernel / probability distribution stuff (~40% of work)
 - limits of uniform convergence: 3 papers (2 of which I understand)
 - limits of invariant risk minimization: 1 paper
- Teaching this partly because I want to learn the foundations in more depth
- **I will probably teach this again next year** (not sure if T1 or T2 yet)

(pause)

“If you’re analyzing data and proving theorems about it
in [ESB], that’s statistics.
If you do it in [ICICS], that’s machine learning.”

– *Larry Wasserman*
(who said it with Baker and Gates, CMU’s equivalents)



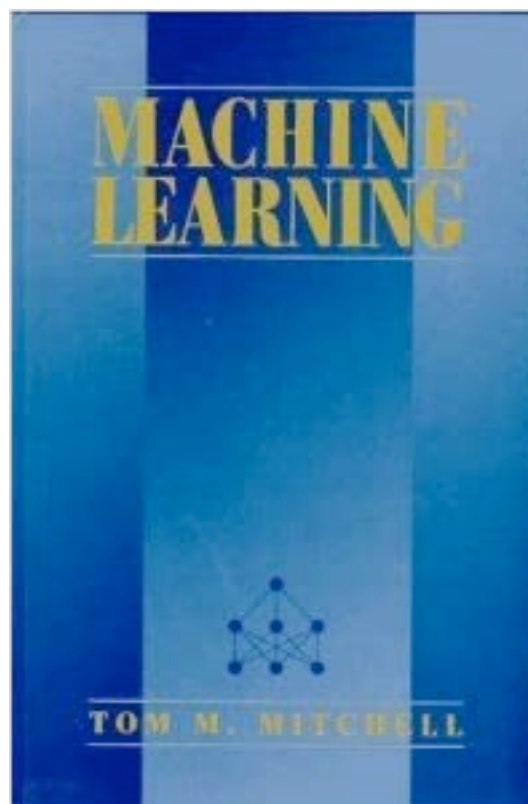
Statistical Science
2001, Vol. 16, No. 3, 199–231

Statistical Modeling: The Two Cultures

Leo Breiman

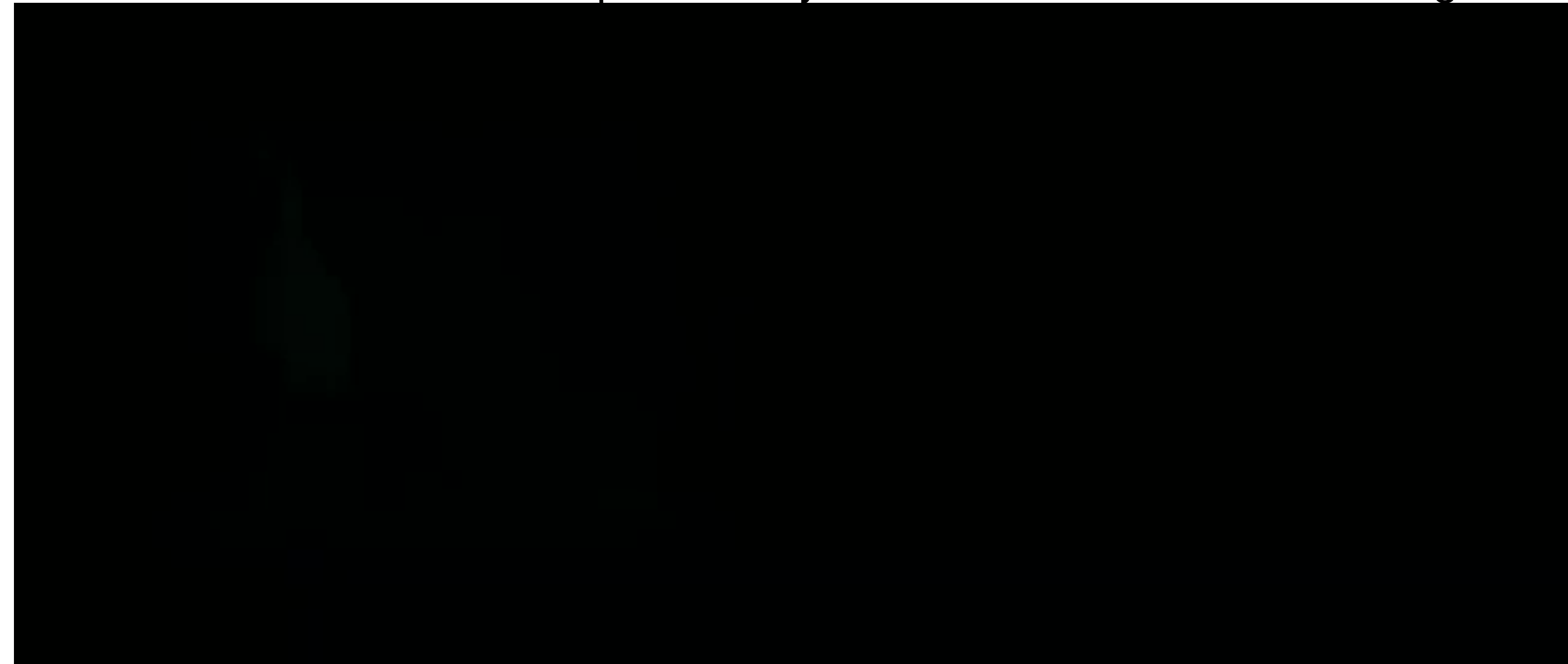
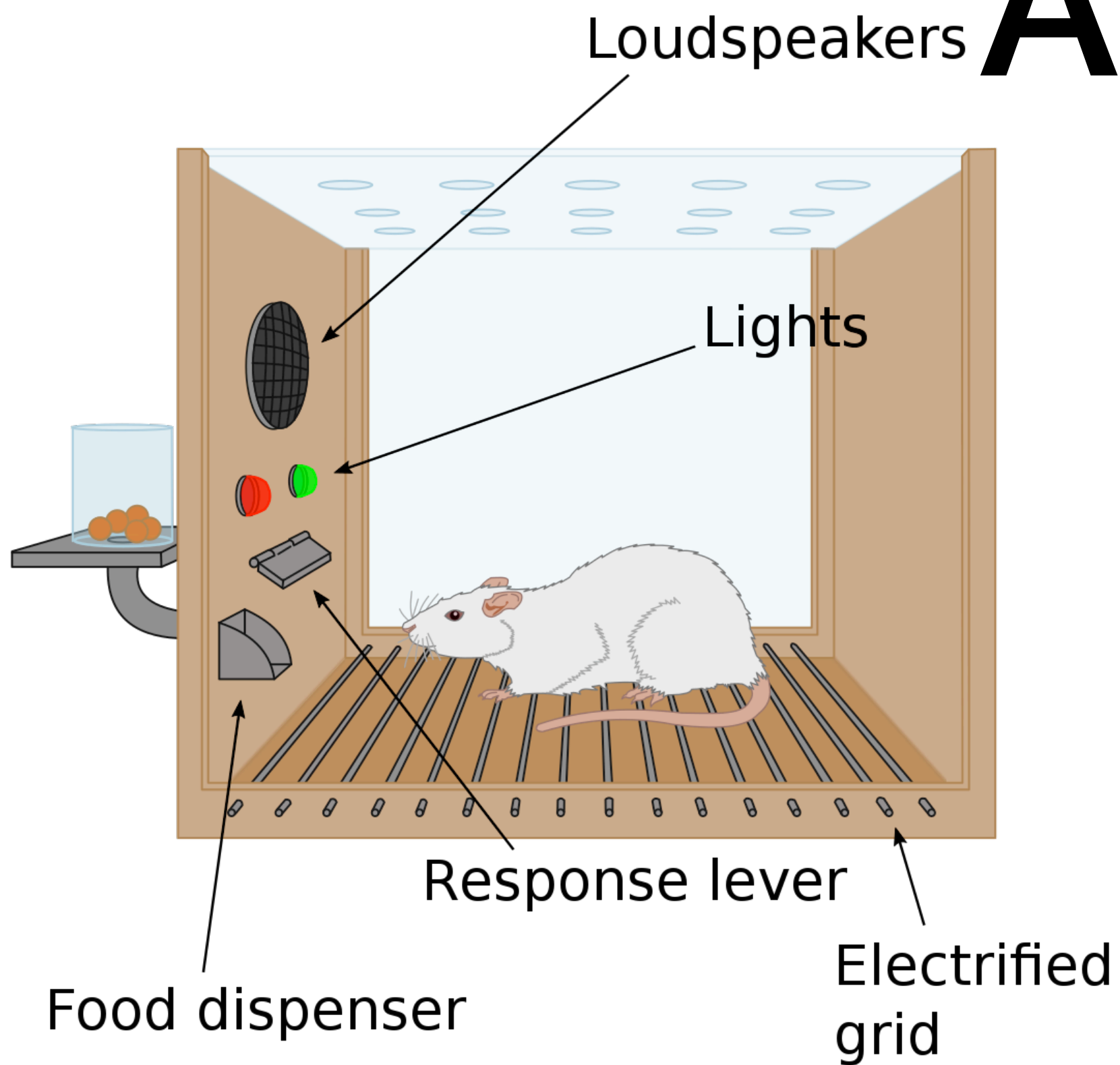
Machine learning

- “A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E .”
- “A checkers learning problem:
 - Task T : playing checkers
 - Performance measure P : percent of games won against opponents
 - Training experience E : playing practice games against itself”
- “A handwriting recognition learning algorithm:
 - Task T : recognizing and classifying handwritten words within images
 - Performance measure P : percent of words correctly classified
 - Training experience E : a database of handwritten words with given classifications”
- “a database system that allows users to update data entries would fit our definition of a learning system: it improves its performance at answering database queries, based on the experience gained from database updates”



Animal learning

<https://www.youtube.com/watch?v=Qv4H81gEGDQ>



“‘Superstition’ in the pigeon” - Skinner

Rats learn to associate
food types ↔ toxin

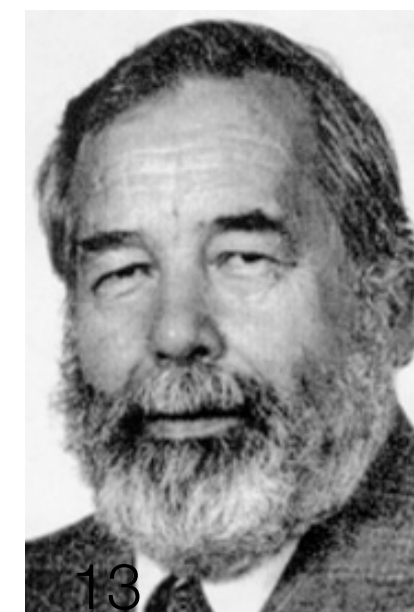
but *don't learn*
food ↔ shock

lights ↔ shock

lights ↔ toxin

**Relation of cue to consequence in
avoidance learning¹**

JOHN GARCIA AND ROBERT A. KOELLING
HARVARD MEDICAL SCHOOL AND MASSACHUSETTS GENERAL HOSPITAL



...why?

- Apparently, different *hypothesis classes*
- Rats maybe have built-in that food \leftrightarrow gastric, light \leftrightarrow shock, not others
 - Helps when it's right
 - Makes it impossible to learn that a light is a “poison detector”
- Pigeons, maybe, don't have these built-ins
 - Presumably could learn that flapping wings \rightarrow food
 - But can cause *overfitting* in other situations

Statistical learning theory

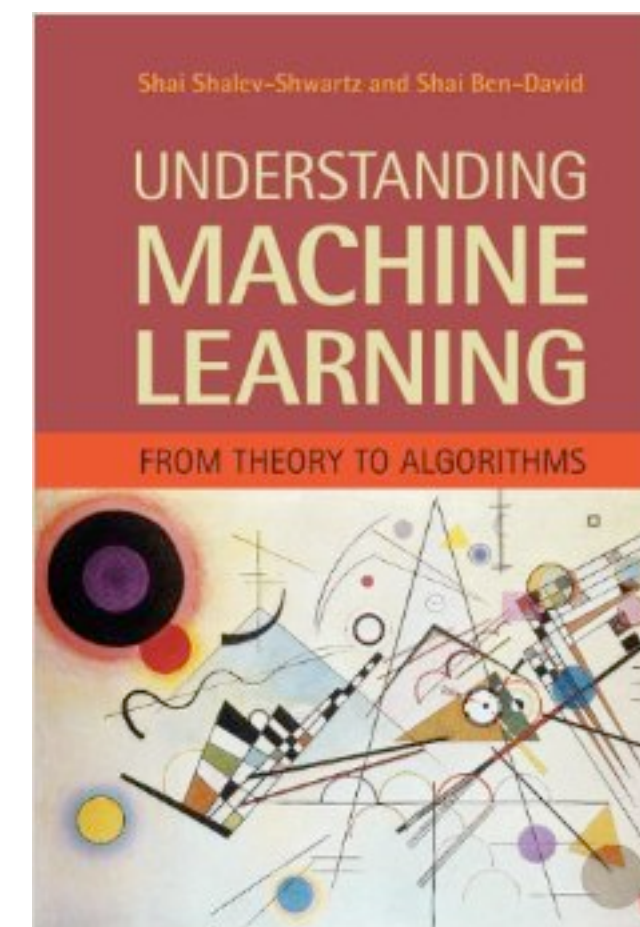
- One main goal of statistical learning theory:
be able to understand these kinds of questions
 - What determines when we can learn?
 - What resources (data in different forms, computation) do we need to do it?
- We'll strive to do it formally and quantitatively:
 - What kinds of assumptions do we need on the data, the learner, ...?
 - Aim for finite-sample, high-probability guarantees
 - How are different analysis techniques related? What limitations are there?

Well-studied foundations...

(kernels!)

THEOREM 6.7 (The Fundamental Theorem of Statistical Learning) *Let \mathcal{H} be a hypothesis class of functions from a domain \mathcal{X} to $\{0, 1\}$ and let the loss function be the 0 – 1 loss. Then, the following are equivalent:*

1. \mathcal{H} has the uniform convergence property.
2. Any ERM rule is a successful agnostic PAC learner for \mathcal{H} .
3. \mathcal{H} is agnostic PAC learnable.
4. \mathcal{H} is PAC learnable.
5. Any ERM rule is a successful PAC learner for \mathcal{H} .
6. \mathcal{H} has a finite VC-dimension.

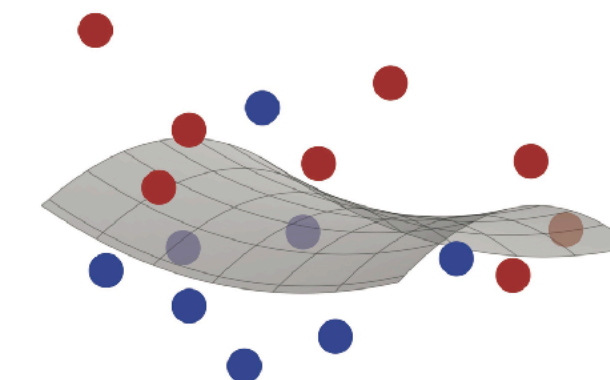


Learning Theory Estimates via Integral Operators and Their Approximations

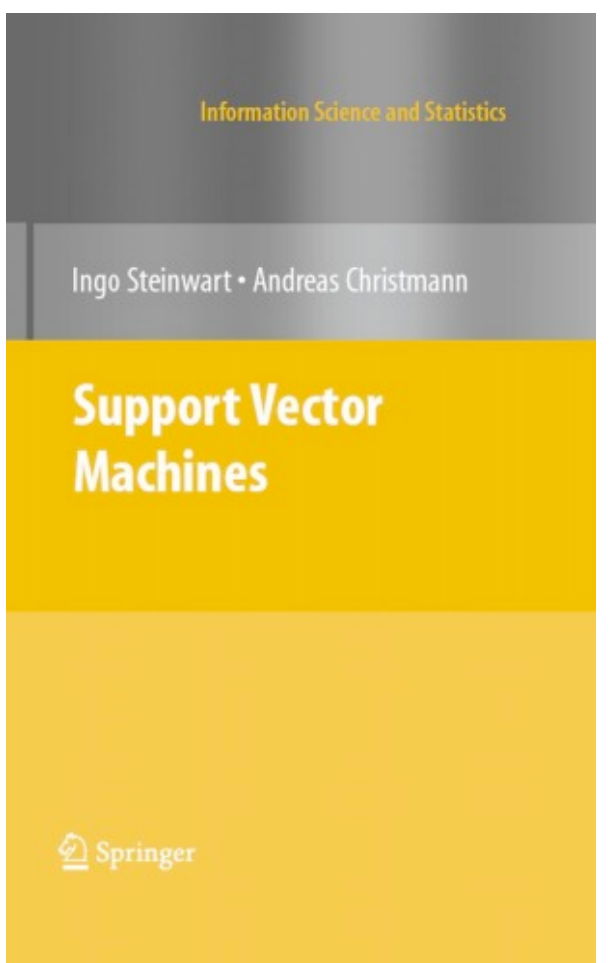
[Steve Smale](#) ✉ & [Ding-Xuan Zhou](#) ✉

[Constructive Approximation](#) **26**, 153–172 (2007) | [Cite this article](#)

Foundations of Machine Learning second edition



Mehryar Mohri,
Afshin Rostamizadeh,
and Ameet Talwalkar



which we're going to learn first!

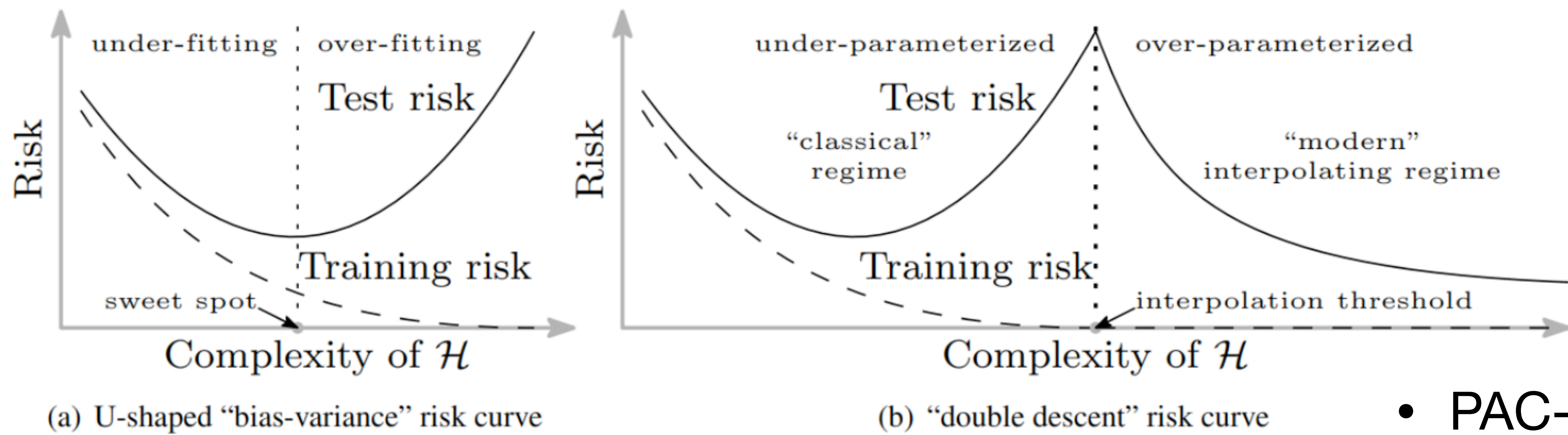
...but they don't explain modern ML

Training error consistently decreases with model complexity, typically dropping to zero if we increase the model complexity enough. However, a model with zero training error is overfit to the training data and will typically generalize poorly.

Table 1: The training and test accuracy (in percentage) of various models on the CIFAR10 dataset.

model	# params	random crop	weight decay	train accuracy	test accuracy
Inception	1,649,402	yes	yes	100.0	89.05
		yes	no	100.0	89.31
		no	yes	100.0	86.03
		no	no	100.0	85.75

To put this in concrete terms, on MNIST, having even 72 hidden units in a fully connected first layer yields vacuous PAC bounds.



- PAC-Bayes
- Oracle bounds

[Submitted on 13 Feb 2019 (v1), last revised 19 Dec 2019 (this version, v3)]

Uniform convergence may be unable to explain generalization in deep learning

Vaishnavh Nagarajan, J. Zico Kolter

[Submitted on 26 Jun 2019 (v1), last revised 29 Jan 2020 (this version, v3)]

Benign Overfitting in Linear Regression

Peter L. Bartlett, Philip M. Long, Gábor Lugosi, Alexander Tsigler

[Submitted on 9 Aug 2020]

What Neural Networks Memorize and Why: Discovering the Long Tail via Influence Estimation

Vitaly Feldman, Chiyuan Zhang

[Submitted on 1 Dec 2020 (v1), last revised 7 Oct 2021 (this version, v3)]

On the robustness of minimum norm interpolators and regularized empirical risk minimizers

Geoffrey Chinot, Matthias Löffler, Sara van de Geer

[Submitted on 10 Nov 2021]

Tight bounds for minimum l1-norm interpolation of noisy data

Guillaume Wang, Konstantin Donhauser, Fanny Yang

[Submitted on 9 Dec 2019 (v1), last revised 27 Feb 2020 (this version, v2)]

In Defense of Uniform Convergence: Generalization via derandomization with an application to interpolating predictors

Jeffrey Negrea, Gintare Karolina Dziugaite, Daniel M. Roy

[Submitted on 16 Oct 2020 (v1), last revised 20 Jan 2021 (this version, v3)]

Failures of model-dependent generalization bounds for least-norm interpolation

Peter L. Bartlett, Philip M. Long

[Submitted on 8 Dec 2021]

Optimistic Rates: A Unifying Theory for Interpolation Learning and Regularization in Linear Regression

Lijia Zhou, Frederic Koehler, Danica J. Sutherland, Nathan Srebro

[Submitted on 6 Oct 2021 (v1), last revised 10 Nov 2021 (this version, v4)]

Foolish Crowds Support Benign Overfitting


Niladri S. Chatterji, Philip M. Long

Other important questions we might get to

- Do we get “implicit regularization” from optimization algorithms?
- When does (S)GD find a good minimum for neural networks?
 - Analysis via neural tangent kernels
- What can deep networks learn that kernels can't?
- When do GPs learn the right posterior distribution?
- When can we learn online? When can we learn privately?
 - ...and is it foreshadowing that these are on the same bullet?
- Does actively selecting points to be labeled help?
- When does self-supervised learning work?
- Does everything break if training and test aren't *exactly* the same distribution?
- Have vision architectures/algorithms overfit to the CIFAR / ImageNet test set?

(pause)

Back to basics: data

- Data x comes from some set \mathcal{X} (domain): often \mathbb{R}^d 
- Labels y from \mathcal{Y} , say $\{0, 1, \dots, 9\}$
- Training data: $S = ((x_1, y_1), \dots, (x_n, y_n))$
 - Referred to as a set, but usually either actually a multiset or a sequence
- Data generation process:
 - Sample x i.i.d. from some distribution \mathcal{D}_x
 - Label $y = f(x)$ according to some function f
 - (We'll allow noise in the process soon)

Back to basics: choosing a model

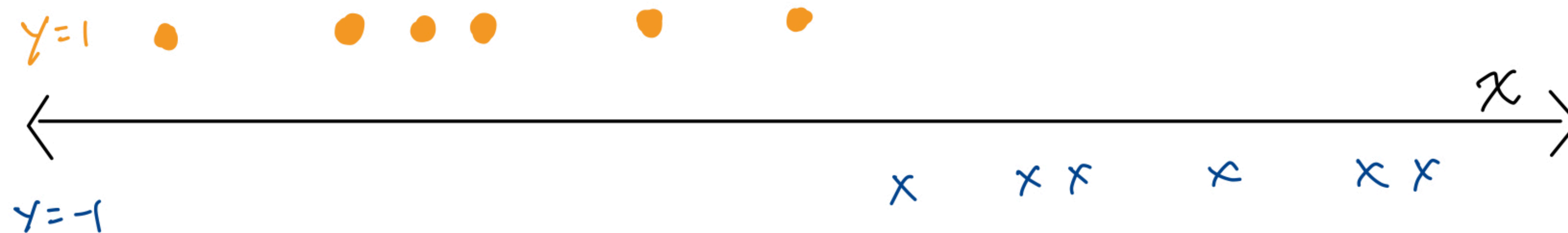
- $x \sim \mathcal{D}_x$, a distribution over \mathcal{X} ; $y = f(x) \in \mathcal{Y}$; $S = ((x_1, y_1), \dots, (x_n, y_n))$
- Want to find a prediction rule $h : \mathcal{X} \rightarrow \mathcal{Y}$
- Would like *most accurate* predictor:
 - $L_{\mathcal{D}_x, f}(h) = \Pr_{x \sim \mathcal{D}_x} (h(x) \neq f(x)) = \mathcal{D}_x(\{x : h(x) \neq f(x)\})$
 - “Generalization error”, “risk”, “true error”, “population loss”
- But we don’t know \mathcal{D}_x or f !

Back to basics: choosing a model

- $x \sim \mathcal{D}_x$, a distribution over \mathcal{X} ; $y = f(x) \in \mathcal{Y}$; $S = ((x_1, y_1), \dots, (x_n, y_n))$
- Want $h : \mathcal{X} \rightarrow \mathcal{Y}$ minimizing $L_{\mathcal{D}_x, f}(h) = \Pr_{x \sim \mathcal{D}_x} (h(x) \neq f(x))$
- Can only estimate $L_{\mathcal{D}_x, f}$ based on sample S
 - $$L_S(h) = \frac{1}{n} \sum_{i=1}^n \begin{cases} 1 & \text{if } h(x_i) \neq y_i \\ 0 & \text{if } h(x_i) = y_i \end{cases}$$
 - Names: {"Training", "empirical"} \times {"error", "loss", "risk"}
 - Also written \hat{L}

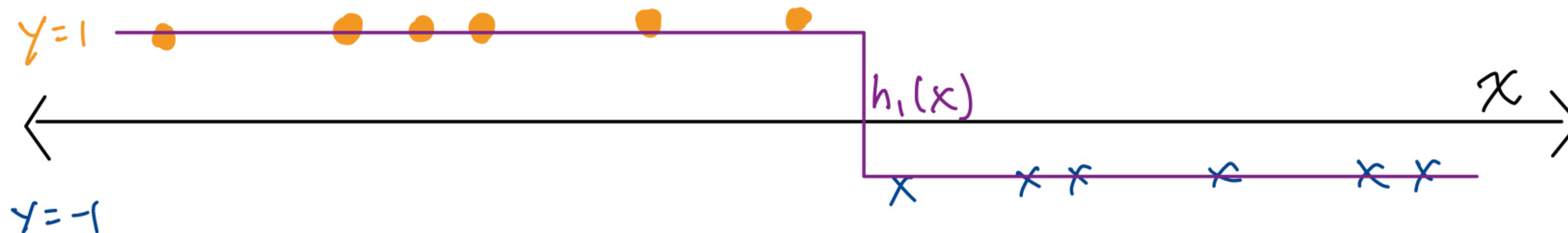
Back to basics: empirical risk minimization

- $x \sim \mathcal{D}_x$, a distribution over \mathcal{X} ; $y = f(x) \in \mathcal{Y}$; $S = ((x_1, y_1), \dots, (x_n, y_n))$
- Want $h : \mathcal{X} \rightarrow \mathcal{Y}$ minimizing $L_{\mathcal{D}_x, f}(h) = \Pr_{x \sim \mathcal{D}_x} (h(x) \neq f(x))$
- 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 $L_S(h)$



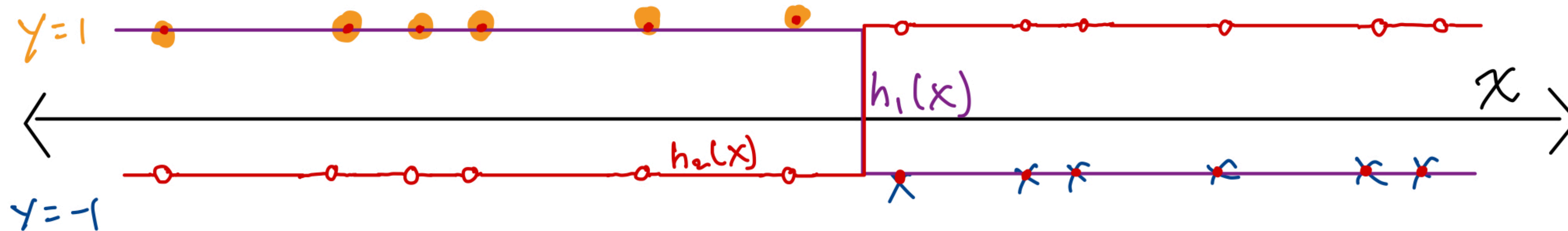
Back to basics: empirical risk minimization

- $x \sim \mathcal{D}_x$, a distribution over \mathcal{X} ; $y = f(x) \in \mathcal{Y}$; $S = ((x_1, y_1), \dots, (x_n, y_n))$
- Want $h : \mathcal{X} \rightarrow \mathcal{Y}$ minimizing $L_{\mathcal{D}_x, f}(h) = \Pr_{x \sim \mathcal{D}_x} (h(x) \neq f(x))$
- 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 $L_S(h)$



Back to basics: empirical risk minimization

- $x \sim \mathcal{D}_x$, a distribution over \mathcal{X} ; $y = f(x) \in \mathcal{Y}$; $S = ((x_1, y_1), \dots, (x_n, y_n))$
- Want $h : \mathcal{X} \rightarrow \mathcal{Y}$ minimizing $L_{\mathcal{D}_x, f}(h) = \Pr_{x \sim \mathcal{D}_x} (h(x) \neq f(x))$
- 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 $L_S(h)$



Back to basics: hypothesis class

- $x \sim \mathcal{D}_x$, a distribution over \mathcal{X} ; $y = f(x) \in \mathcal{Y}$; $S = ((x_1, y_1), \dots, (x_n, y_n))$
- Want $h : \mathcal{X} \rightarrow \mathcal{Y}$ minimizing $L_{\mathcal{D}_x, f}(h) = \Pr_{x \sim \mathcal{D}_x} (h(x) \neq f(x))$
- 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** \mathcal{H} of functions $h : \mathcal{X} \rightarrow \mathcal{Y}$

Choosing a hypothesis class

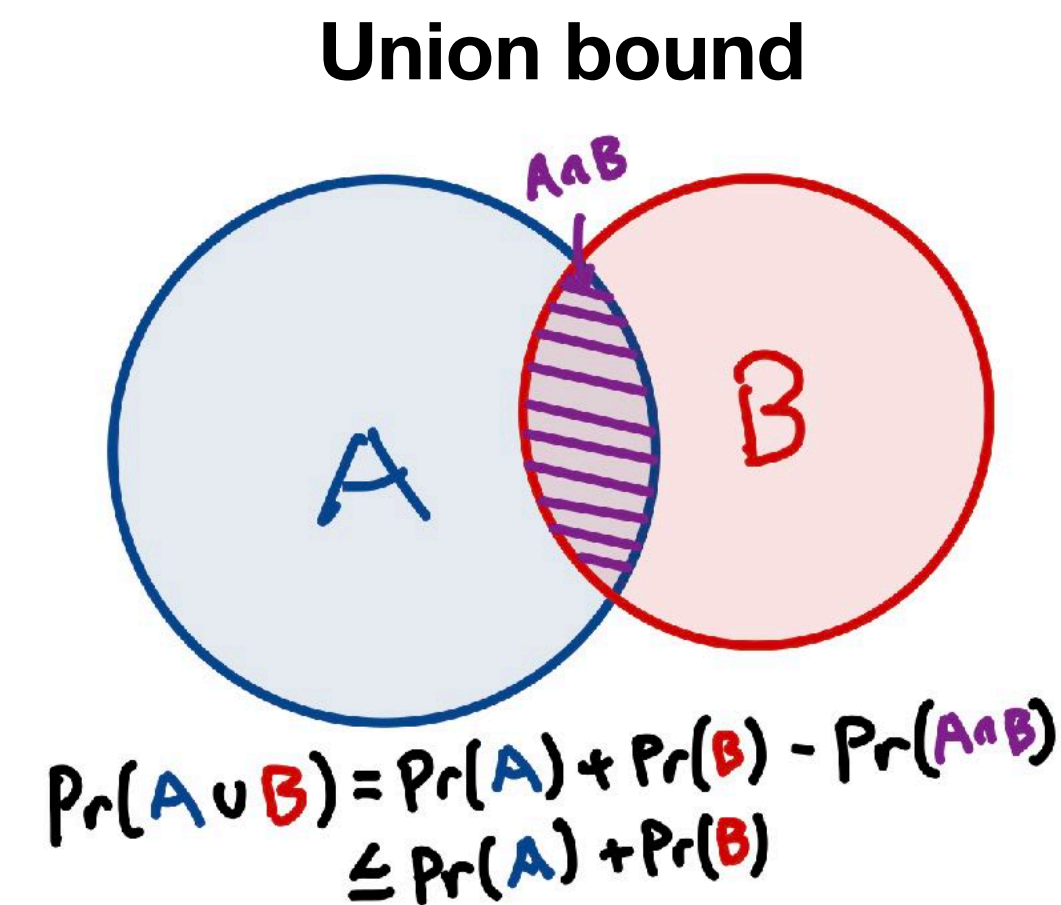
- In this basic setup, we should choose \mathcal{H} *before* looking at the data
 - Will avoid this later in the course (regularization / SRM)
- If \mathcal{H} is too “big”: leads to overfitting (superstitious pigeons)
- If too “small”: can’t learn things we need to (rats with poison indicator light)
- Let’s start simple: *finite* \mathcal{H}
 - Simple things like threshold functions aren’t finite
 - but thresholding at a float32 is
 - Also finite: “all Python programs of size < 1 GB”

Finite hypothesis class

- To start with something simple, assume **realizability**:
there is an $h^* \in \mathcal{H}$ with $L_{\mathcal{D}_x, f}(h^*) = 0$
- Implies (a.s.) that $L_S(h^*) = 0$
- Now, will ERM work?
- We might get a really unlucky S , e.g. every example has $y = -1$
 - but, hopefully, it will **probably** work (high probability over S)
- In continuous settings, we might not ever get *exactly* h^*
 - but, hopefully, we'll get something that's **approximately correct**
- **PAC framework**: Probably Approximately Correct

Realizable, finite \mathcal{H}

- $h_S \in \arg \min_{h \in \mathcal{H}} L_S(h)$: realizable means $L_S(h_S) = 0$, but maybe $L_{\mathcal{D}_x, f}(h_S) > 0$
- Would like to show $\Pr_S \left(L_{\mathcal{D}_x, f}(h_S) \leq \varepsilon \right) \geq 1 - \delta$, i.e. $\Pr(L(h_S) > \varepsilon) < \delta$
- Call \mathcal{H}_B the set of “bad” hypotheses, $\left\{ h \in \mathcal{H} : L_{\mathcal{D}_x, f}(h) > \varepsilon \right\}$
- $M = \left\{ S : \exists h \in \mathcal{H}_B . L_S(h) = 0 \right\}$ is set of “bad” samples
 - If $L_{\mathcal{D}_x, f}(h_S) > \varepsilon$, then $S \in M$
- $$\Pr(L(h_S) > \varepsilon) \leq \mathcal{D}_x^n(M) = \mathcal{D}_x^n \left(\bigcup_{h \in \mathcal{H}_B} \{S : L_S(h) = 0\} \right) \leq \sum_{h \in \mathcal{H}_B} \mathcal{D}_x^n \left(\{S : L_S(h) = 0\} \right)$$



This is where we stopped, halfway through the proof.
To be concluded!

Recap

- Learning theory – yay!
- Defined data distribution, loss function, true loss, empirical loss
- Empirical risk minimization: a semi-reasonable principle
- PAC learnability
 - Finite classes are (realizable-)PAC learnable with $\frac{1}{\epsilon} \log \left(\frac{|\mathcal{H}|}{\delta} \right)$ samples
- **Next time:** finish the proof + uniform convergence.