Machine Learning Theory: Overview

COMS 4995-1 Spring 2020 (Daniel Hsu)

Agenda

- Today:
  - About machine learning theory
  - About the course
  - Some examples of learning problems/settings
- Next time:
  - Concentration of measure

What is machine learning? (1)

What is machine learning? (1)

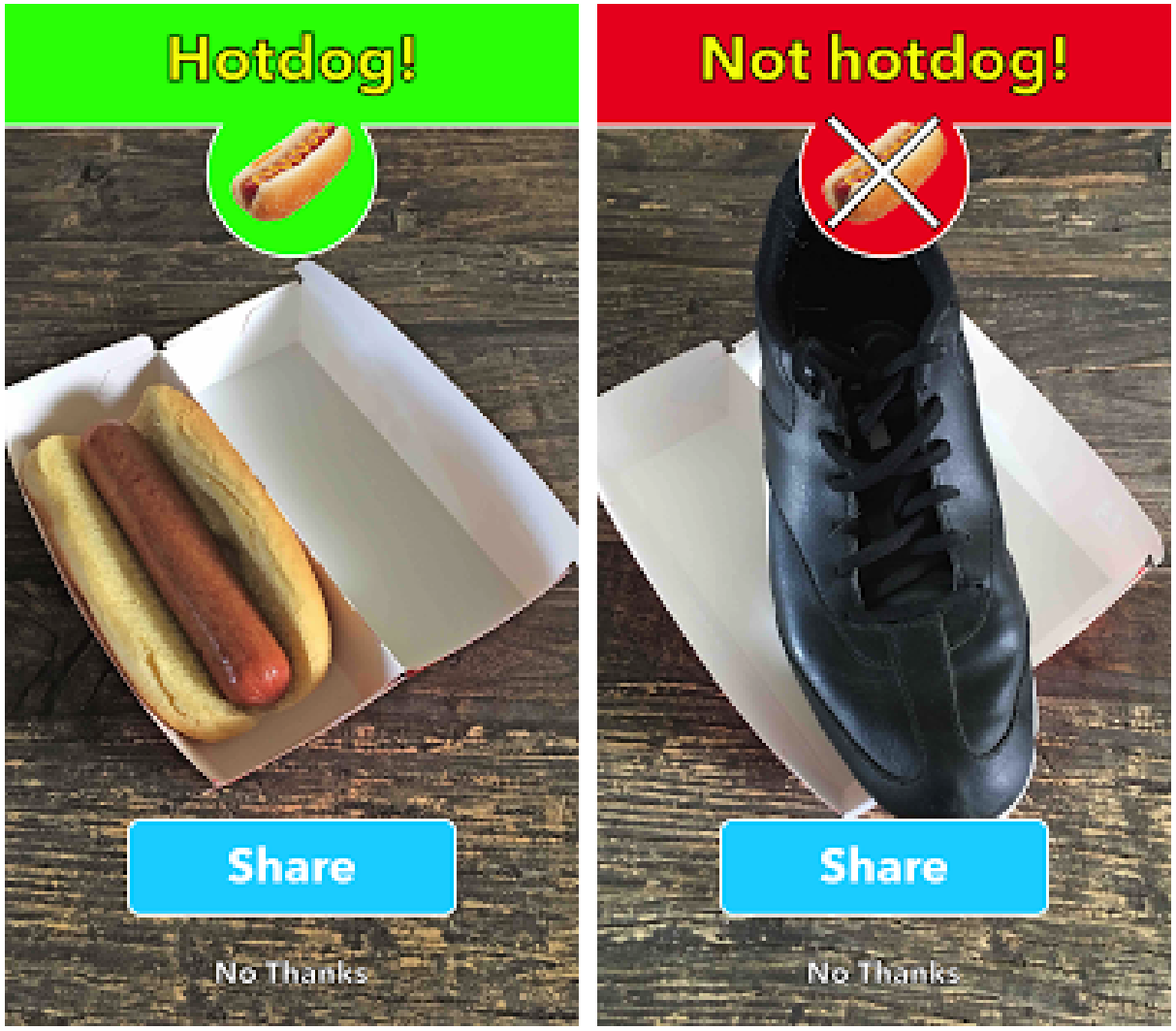


Image credit: http://www.seefoodtechnologies.com/nothotdog/

What is machine learning? (2)

Examples:

- Spam filtering (from email text)
- Ad click prediction (from user profile and context)
- Gene expression level prediction (from upstream DNA)
- Best-next-move prediction (from state of chess board)
- ...
- Programming-by-demonstration

What is machine learning? (2)

Examples:

- Spam filtering (from email text)
- Ad click prediction (from user profile and context)
- Gene expression level prediction (from upstream DNA)
- Best-next-move prediction (from state of chess board)
- ...
- Programming-by-demonstration

Note: This is *not* an introductory course in machine learning. Also, we won't be overly concerned with practical applications / methods.

Please see COMS 4771 for general non-theoretical introduction and COMS 4995-11 for an applied introduction.

What is learning theory?

- Design/analysis of machine learning algorithms/problems
  - Computational resources: running time, memory, ...
  - Data resources: sample size, rounds of interaction, ...
- Many different models for theoretical analysis
  - Statistical learning
  - Online learning
  - Learning with queries
  - Finding planted structures
  - ...

Why study learning theory? (1)

Relevance to machine learning practice

- Breiman (1995) "Reflections After Refereeing Papers for NIPS"

2. USES OF THEORY

- **Comfort:** We knew it worked, but it's nice to have a proof.
- **Insight:** Aha! So that's why it works.
- **Innovation:** At last, a mathematically proven idea that applies to data.
- **Suggestion:** Something like this might work with data.

Why study learning theory? (1)

Relevance to machine learning practice

- Breiman (1995) "Reflections After Refereeing Papers for NIPS"

2. USES OF THEORY

- **Comfort:** We knew it worked, but it's nice to have a proof.
- **Insight:** Aha! So that's why it works.
- **Innovation:** At last, a mathematically proven idea that applies to data.
- **Suggestion:** Something like this might work with data.

Breiman's "Post World War II" Examples (in 1995):

- Asymptotic analyses of decision trees, nearest neighbor, universal approximation
- Nonparametric regression, sparsity in inverse problems
- Spectral analysis in time series, information theory, bootstrap
- Theory-inspired heuristics for function fitting

Why study learning theory? (2)

Relevance to machine learning practice

- Breiman (1995) "Reflections After Refereeing Papers for NIPS"

> **Mathematical theory is not critical to the development of machine learning.**
>
> ***But scientific inquiry is.***

3.5 INQUIRY

INQUIRY = sensible and intelligent efforts to understand what is going on. For example:

- mathematical heuristics
- simplified analogies (like the Ising Model)
- simulations
- comparisons of methodologies
- devising new tools
- theorems where useful (rare!)
- shunning panaceas

Why study learning theory? (3)

Insights into general phenomenon of learning

- Valiant (1984) "A Theory of the Learnable"

ABSTRACT: *Humans appear to be able to learn new concepts without needing to be programmed explicitly in any conventional sense. In this paper we regard learning as the phenomenon of knowledge acquisition in the absence of explicit programming. We give a precise methodology for studying this phenomenon from a computational viewpoint.*

Why study learning theory? (3)

Insights into general phenomenon of learning

- Valiant (1984) "A Theory of the Learnable"

ABSTRACT: *Humans appear to be able to learn new concepts without needing to be programmed explicitly in any conventional sense. In this paper we regard learning as the phenomenon of knowledge acquisition in the absence of explicit programming. We give a precise methodology for studying this phenomenon from a computational viewpoint.*

My suggestions

- Study learning theory for its breadth of topics and the wide applicability of its methods
- View theorems as demonstrations of careful understanding
- Find/develop your own personal motivation

About this course

- COMS 4995-1
  - Will be "COMS 4773" in the future; can count towards degree program requirements as such.
- Website (with syllabus, schedule, etc):
  http://www.cs.columbia.edu/~djhsu/LT/
- Topics:
  - Statistical learning (e.g., generalization theory)
  - Online learning (e.g., learning with experts, multi-arm bandits)
  - Unsupervised learning (e.g., clustering models), **if time permits**
- Learning goals:
  - Rigorously analyze ML problems/algorithms
  - Read/understand research papers in ML theory

Course requirements

Prerequisites

- Mathematical maturity; reading and writing proofs
- Probability, linear algebra, a bit of convex analysis
- Prior exposure to machine learning (maybe just for motivation)

Requirements

- Reading assignments – schedule on website
  - Primarily from a few textbooks, available on the website
- Homework assignments – will be posted on website
  - 75% of overall grade
- Project – instructions on website
  - 25% of overall grade

Course requirements

Prerequisites

- Mathematical maturity; reading and writing proofs
- Probability, linear algebra, a bit of convex analysis
- Prior exposure to machine learning (maybe just for motivation)

Requirements

- Reading assignments – schedule on website
  - Primarily from a few textbooks, available on the website
- Homework assignments – will be posted on website
  - 75% of overall grade
- Project – instructions on website
  - 25% of overall grade

[Do you want to do scribe notes (perhaps for extra credit)?]

Example: Pattern recognition 1

What is the pattern (input/output relationship)?

Input	**Output**
$(3, 5)$	$+1$
$(9, 1)$	-1
$(2, 7)$	$+1$
$(4, 2)$	-1
$(8, 8)$	-1
$(5, 2)$	-1
$(3, 1)$	-1
$(1, 3)$	$+1$
$(2, 5)$	$+1$
$(5, 3)$	-1
$(4, 2)$	-1

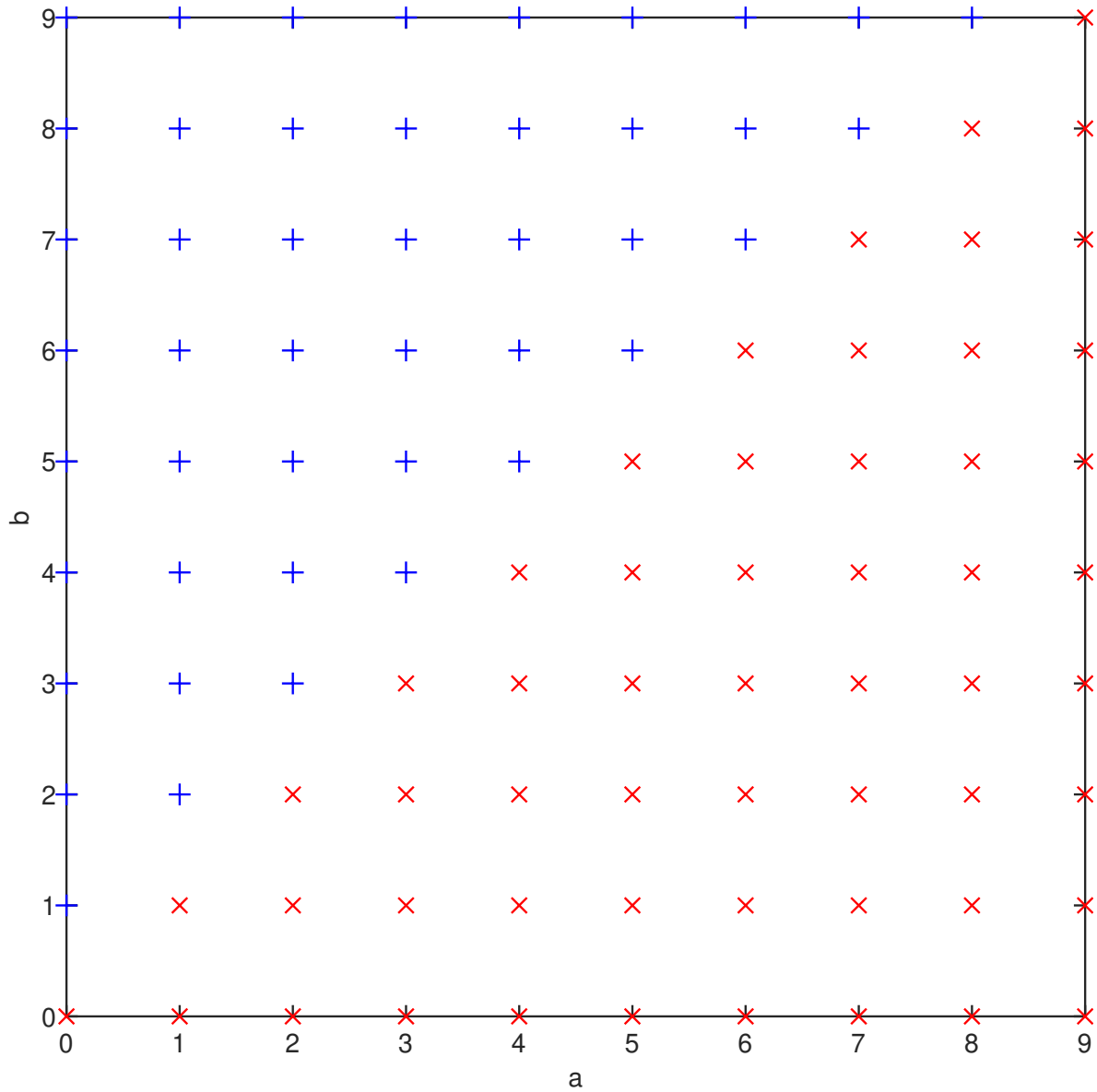
Example: Pattern recognition 1

What is the pattern (input/output relationship)?

Input	**Output**
$(3, 5)$	$+1$
$(9, 1)$	-1
$(2, 7)$	$+1$
$(4, 2)$	-1
$(8, 8)$	-1
$(5, 2)$	-1
$(3, 1)$	-1
$(1, 3)$	$+1$
$(2, 5)$	$+1$
$(5, 3)$	-1
$(4, 2)$	-1

Answer: "First number is less than the second number"

Example: Pattern recognition 1 - all points



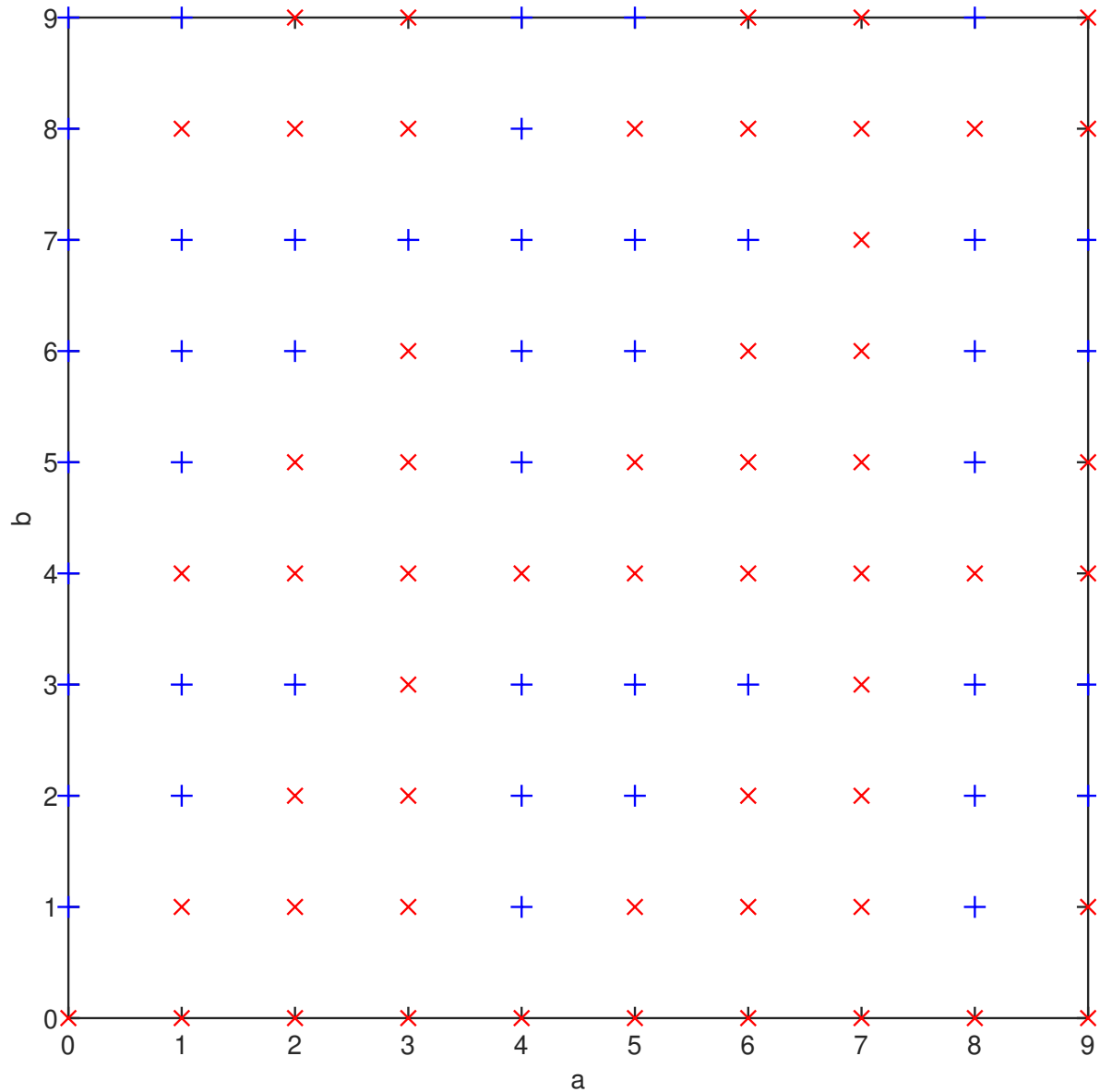
$$h(a, b) = \text{sign}(-a + b)$$

Example: Pattern recognition 2

What is the pattern (input/output relationship)?

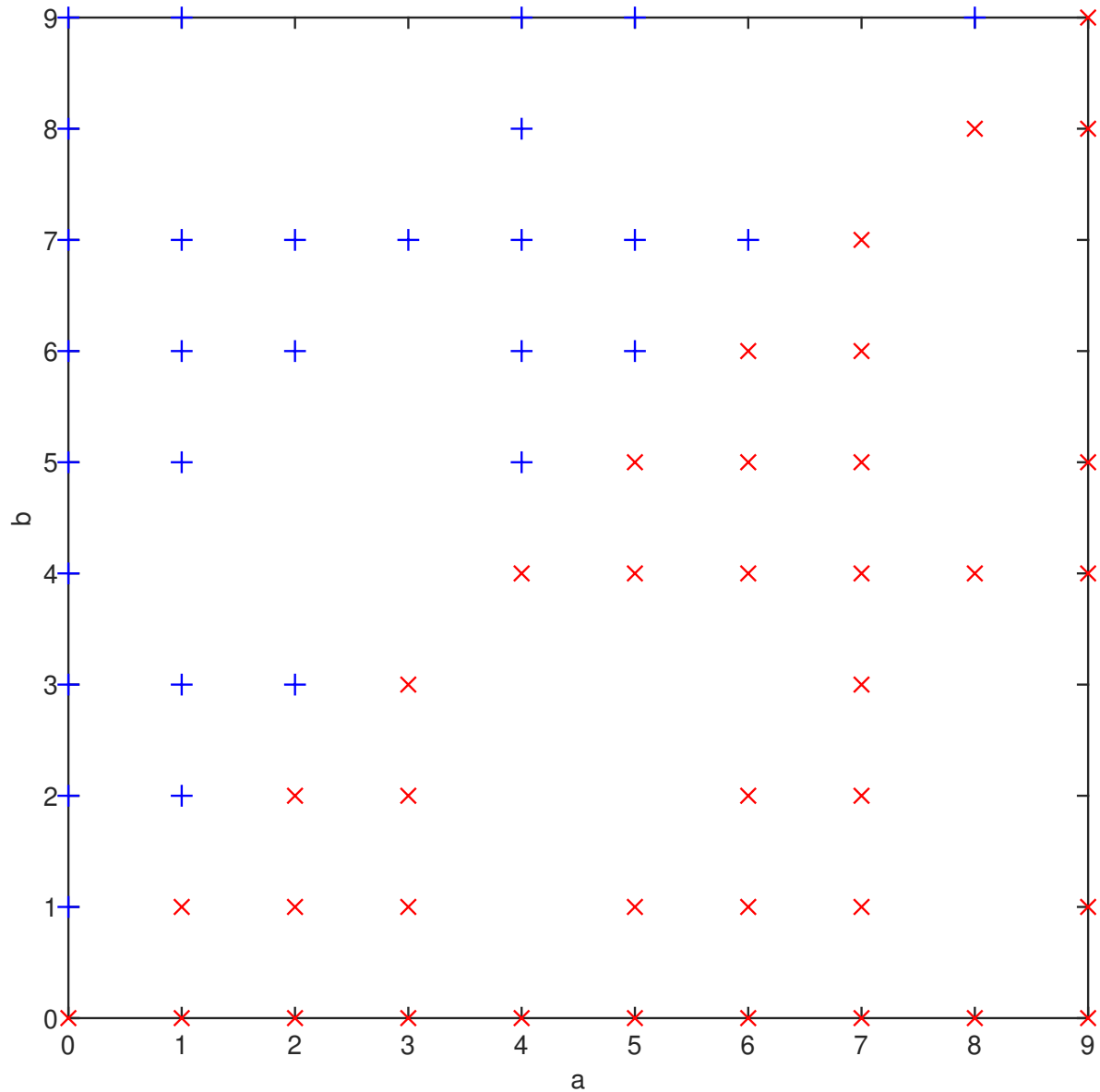
Input	**Output**
$(3, 5)$	-1
$(9, 1)$	-1
$(2, 7)$	$+1$
$(4, 2)$	$+1$
$(8, 8)$	-1
$(5, 2)$	$+1$
$(3, 1)$	-1
$(1, 3)$	$+1$
$(2, 5)$	-1
$(5, 3)$	$+1$
$(4, 2)$	$+1$

Example: Pattern recognition 2 - all points



???

Example: Pattern recognition 2 - subset of points



What if you only saw these examples?

Basic questions

Some basic questions when thinking about pattern recognition / prediction problems

- What kinds of patterns are we considering? Hypothesis class
  - E.g., neural networks
- What is the nature of our data? Data model
  - E.g., iid sample from P
- When has "learning" been achieved? Success criterion
  - E.g., $P(h(X) \neq Y)$ is small

Example: Online learning of linear classifiers

- Hypothesis class: linear classifiers in $\mathbb{R}^d$

$$\boldsymbol{x} \mapsto \operatorname{sign}(\boldsymbol{x}^\mathsf{T}\boldsymbol{w})$$

- Data model: arbitrary bounded sequence of examples $((\boldsymbol{x}_t, y_t))_{t\geq 1}$ from $\mathbb{R}^d \times \{\pm 1\}$; promise that $\exists \boldsymbol{w}^\star \in \mathbb{R}^d$ s.t.

$$y_t = \operatorname{sign}(\boldsymbol{x}_t^\mathsf{T}\boldsymbol{w}^\star) \quad (\forall t \geq 1)$$

- Success criterion:
  - Examples $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots$ revealed one-by-one
  - For t-th example $(\boldsymbol{x}_t, y_t)$:
    - Learner first sees $\boldsymbol{x}_t$ and must make prediction $\hat{y}_t \in \{\pm 1\}$
    - Then, learner sees the correct y_t.
  - Learner is successful if total number of mistakes M_t after t examples is $o(t)$.

Example: Online learning of linear classifiers

- Hypothesis class: linear classifiers in $\mathbb{R}^d$

$$\boldsymbol{x} \mapsto \operatorname{sign}(\boldsymbol{x}^\mathsf{T}\boldsymbol{w})$$

- Data model: arbitrary bounded sequence of examples $((\boldsymbol{x}_t, y_t))_{t\geq 1}$ from $\mathbb{R}^d \times \{\pm 1\}$; promise that $\exists \boldsymbol{w}^\star \in \mathbb{R}^d$ s.t.

$$y_t = \operatorname{sign}(\boldsymbol{x}_t^\mathsf{T}\boldsymbol{w}^\star) \quad (\forall t \geq 1)$$

- Success criterion:
  - Examples $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots$ revealed one-by-one
  - For t-th example $(\boldsymbol{x}_t, y_t)$:
    - Learner first sees $\boldsymbol{x}_t$ and must make prediction $\hat{y}_t \in \{\pm 1\}$
    - Then, learner sees the correct y_t.
  - Learner is successful if total number of mistakes M_t after t examples is $o(t)$.

No learner is always successful.

Example: Online learning of linear classifiers with margins

- Hypothesis class: linear classifiers in $\mathbb{R}^d$

$$\boldsymbol{x} \mapsto \operatorname{sign}(\boldsymbol{x}^\mathsf{T}\boldsymbol{w})$$

- Data model: arbitrary bounded sequence of examples $((\boldsymbol{x}_t, y_t))_{t\geq 1}$ from $\mathbb{R}^d \times \{\pm 1\}$; promise that $\exists \boldsymbol{w}^\star \in \mathbb{R}^d$ s.t.

$$y_t \boldsymbol{x}_t^\mathsf{T} \boldsymbol{w}^\star \geq 1 \quad (\forall t \geq 1)$$

- Success criterion:
  - Examples $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots$ revealed one-by-one
  - For t-th example $(\boldsymbol{x}_t, y_t)$:
    - Learner first sees $\boldsymbol{x}_t$ and must make prediction $\hat{y}_t \in \{\pm 1\}$
    - Then, learner sees the correct y_t.
  - Learner is successful if total number of mistakes M_t after t examples is $o(t)$.

Example: Online learning of linear classifiers with margins

- Hypothesis class: linear classifiers in $\mathbb{R}^d$

$$\boldsymbol{x} \mapsto \operatorname{sign}(\boldsymbol{x}^\mathsf{T}\boldsymbol{w})$$

- Data model: arbitrary bounded sequence of examples $((\boldsymbol{x}_t, y_t))_{t\geq 1}$ from $\mathbb{R}^d \times \{\pm 1\}$; promise that $\exists \boldsymbol{w}^\star \in \mathbb{R}^d$ s.t.

$$y_t \boldsymbol{x}_t^\mathsf{T} \boldsymbol{w}^\star \geq 1 \quad (\forall t \geq 1)$$

- Success criterion:
  - Examples $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots$ revealed one-by-one
  - For t-th example $(\boldsymbol{x}_t, y_t)$:
    - Learner first sees $\boldsymbol{x}_t$ and must make prediction $\hat{y}_t \in \{\pm 1\}$
    - Then, learner sees the correct y_t.
  - Learner is successful if total number of mistakes M_t after t examples is $o(t)$.

"Perceptron" algorithm is always successful.

(Online) Perceptron

- Start with $\boldsymbol{w}^{(0)} := \mathbf{0}$
- For $t = 1, 2, \ldots$:
  - Predict $\hat{y}_t := \operatorname{sign}(\boldsymbol{x}_t^\intercal \boldsymbol{w}^{(t-1)})$
  - Update:

$$\boldsymbol{w}^{(t)} := \begin{cases} \boldsymbol{w}^{(t-1)} + y_t \boldsymbol{x}_t & \text{if } \hat{y}_t \neq y_t \\ \boldsymbol{w}^{(t-1)} & \text{otherwise} \end{cases}$$

(Online) Perceptron

- Start with $\boldsymbol{w}^{(0)} := \mathbf{0}$
- For $t = 1, 2, \ldots$:
  - Predict $\hat{y}_t := \operatorname{sign}(\boldsymbol{x}_t^\mathsf{T} \boldsymbol{w}^{(t-1)})$
  - Update:

$$\boldsymbol{w}^{(t)} := \begin{cases} \boldsymbol{w}^{(t-1)} + y_t \boldsymbol{x}_t & \text{if } \hat{y}_t \neq y_t \\ \boldsymbol{w}^{(t-1)} & \text{otherwise} \end{cases}$$

Theorem (Novikoff, 1962). Fix any sequence $((\boldsymbol{x}_t, y_t))_{t \geq 1}$ from $\mathbb{R}^d \times \{\pm 1\}$ such that (i) $L := \sup_t \|\boldsymbol{x}_t\|_2 < +\infty$, and (ii) there exists $\boldsymbol{w}^\star \in \mathbb{R}^d$ satisfying

$$y_t \boldsymbol{x}_t^\mathsf{T} \boldsymbol{w}^\star \geq 1 \quad (\forall t \geq 1).$$

On this sequence, Perceptron has $M_t \leq L^2 \|\boldsymbol{w}^\star\|_2^2$ for all $t \geq 1$.

[See a demo? Go through the proof?]

Example: Linear regression

- Hypothesis class: linear functions in $\mathbb{R}^d$

$$\boldsymbol{x} \mapsto \boldsymbol{x}^\mathsf{T}\boldsymbol{w}$$

- Data model: iid random examples $((\boldsymbol{X}_i, Y_i))_{i=1}^n$ from $\mathbb{R}^d \times \mathbb{R}$ (at least satisfying moment conditions s.t. stuff below is finite)
- Success criterion:
  - Let $(\boldsymbol{X}, Y)$ be an independent copy of $(\boldsymbol{X}_1, Y_1)$
  - Learner returns a linear function $\hat{\boldsymbol{w}} = \hat{\boldsymbol{w}}((\boldsymbol{X}_1, Y_1), \dots, (\boldsymbol{X}_n, Y_n))$
  - Learner is successful if the "risk" of $\hat{\boldsymbol{w}}$, defined by

$$\mathcal{R}(\boldsymbol{w}) := \mathbb{E}[(\boldsymbol{X}^\mathsf{T}\boldsymbol{w} - Y)^2] \quad (\forall \boldsymbol{w} \in \mathbb{R}^d),$$

  is not much larger than $\min_{\boldsymbol{w} \in \mathbb{R}^d} \mathcal{R}(\boldsymbol{w})$.

Example: Linear regression

- Hypothesis class: linear functions in $\mathbb{R}^d$

$$\boldsymbol{x} \mapsto \boldsymbol{x}^\mathsf{T}\boldsymbol{w}$$

- Data model: iid random examples $((\boldsymbol{X}_i, Y_i))_{i=1}^n$ from $\mathbb{R}^d \times \mathbb{R}$ (at least satisfying moment conditions s.t. stuff below is finite)
- Success criterion:
  - Let $(\boldsymbol{X}, Y)$ be an independent copy of $(\boldsymbol{X}_1, Y_1)$
  - Learner returns a linear function $\hat{\boldsymbol{w}} = \hat{\boldsymbol{w}}((\boldsymbol{X}_1, Y_1), \dots, (\boldsymbol{X}_n, Y_n))$
  - Learner is successful if the "risk" of $\hat{\boldsymbol{w}}$, defined by

$$\mathcal{R}(\boldsymbol{w}) := \mathbb{E}[(\boldsymbol{X}^\mathsf{T}\boldsymbol{w} - Y)^2] \quad (\forall \boldsymbol{w} \in \mathbb{R}^d),$$

  is not much larger than $\min_{\boldsymbol{w} \in \mathbb{R}^d} \mathcal{R}(\boldsymbol{w})$.

What is a good strategy here?

Empirical risk minimization

- Let P be the probability distribution of $(\boldsymbol{X}, Y)$.
  - Risk $\mathcal{R}(\boldsymbol{w})$ is mean squared prediction error of $\boldsymbol{w}$ w.r.t. P.
  - If we know P, then in principle we can just minimize $\mathcal{R}$.
  - What if we just have the random examples $((\boldsymbol{X}_i, Y_i))_{i=1}^n$?

Empirical risk minimization

- Let P be the probability distribution of $(\boldsymbol{X}, Y)$.
  - Risk $\mathcal{R}(\boldsymbol{w})$ is mean squared prediction error of $\boldsymbol{w}$ w.r.t. P.
  - If we know P, then in principle we can just minimize $\mathcal{R}$.
  - What if we just have the random examples $((\boldsymbol{X}_i, Y_i))_{i=1}^n$?
- **Plug-in principle**: pretend P_n is P, where

$$P_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{(\boldsymbol{X}_i, Y_i)}$$

  is the "empirical distribution", and proceed as above.

Empirical risk minimization

- Let P be the probability distribution of $(\boldsymbol{X}, Y)$.
  - Risk $\mathcal{R}(\boldsymbol{w})$ is mean squared prediction error of $\boldsymbol{w}$ w.r.t. P.
  - If we know P, then in principle we can just minimize $\mathcal{R}$.
  - What if we just have the random examples $((\boldsymbol{X}_i, Y_i))_{i=1}^n$?
- **Plug-in principle**: pretend P_n is P, where

$$P_n = \frac{1}{n} \sum_{i=1}^{n} \delta_{(\boldsymbol{X}_i, Y_i)}$$

  is the "empirical distribution", and proceed as above.

This general approach—not just for linear functions—is called "Empirical Risk Minimization (ERM)".

[Go through an analysis?]

Example: Planted bisection model

- Hypothesis class: groupings of discrete items (e.g., clustering)
- Data model: graph on n vertices
  - vertices are partitioned into two groups of $n/2$ each
  - appearance of edges are random and independent
  - $\{u, v\}$ appears with probability p if u, v in same group
  - $\{u, v\}$ appears with probability $q < p$ if u, v in different groups
  - graph is observed; bisection is "hidden"
- Success criterion:
  - (Approximately) recover the bisection

Recap

- Recap:
  - About machine learning theory
  - About the course
  - Some examples of learning problems/settings
- Next time:
  - Concentration of measure
  - Please read UML Chapter 1 and Appendix B
  - Homework 0 (required!) is out; due next week Jan 31