CSE 417T
Introduction to Machine Learning

Lecture 4

Instructor: Chien-Ju (CJ) Ho

Logistics: Homework 1

- Due: **September 23 (Friday), 2022**
  - http://chienjuho.com/courses/cse417t/hw1.pdf
  - Two submission links: Report and Code (The links will be up over the weekend)
    - Report: Answer all questions, including the implementation question
      - **Grades are based on the report**
    - Code: Complete and submit **hw1.py** for Problem 2
      - The code will only be used for correctness checking (when in doubts) and plagiarism checking
  - Reserve time if you never used Gradescope.
    - Make sure to **specify the pages for each problem**. You **won't get points** otherwise

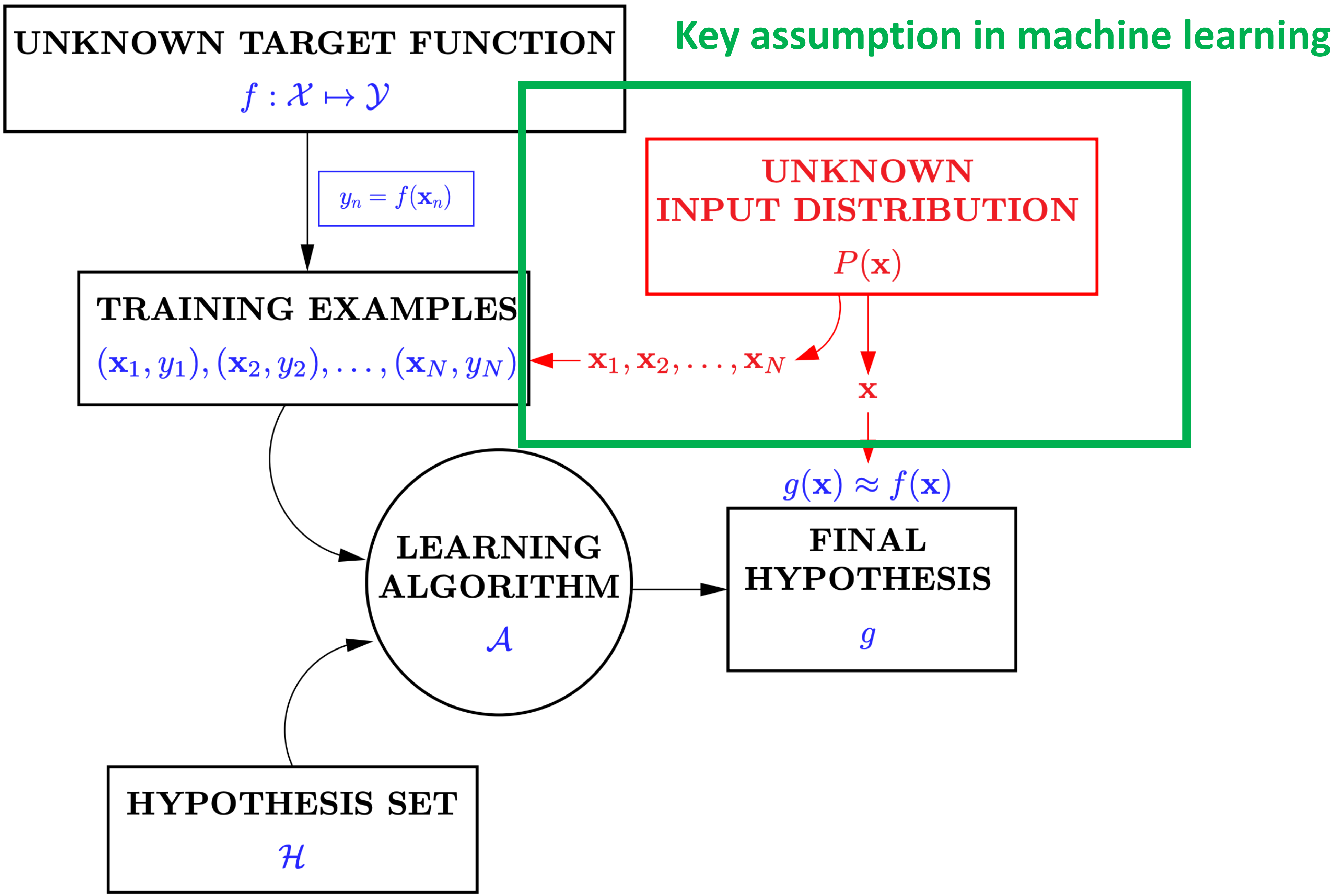
Logistics: Office Hours

- Tentative schedule of TA office hours (starting next Monday)

Monday	9:30am Asher Baraban	3pm Qihang Zhao	
Tuesday	10am Di Huang	1pm Andrew Ruttenberg	4pm Quinn Wai Wong
Wednesday	1pm Wenxuan Zhu	3pm William Sepesi	4:30pm Sylvia Tang
Thursday	11:30am Yuan Liu	4pm Elyse Tang	7pm Fankun Zen
Friday	11am Riggie Kong	3pm Nan Huang	5:30pm Weiwei Ma
Sunday	Noon Jonathan Ma	1:30pm Kenneth Li	

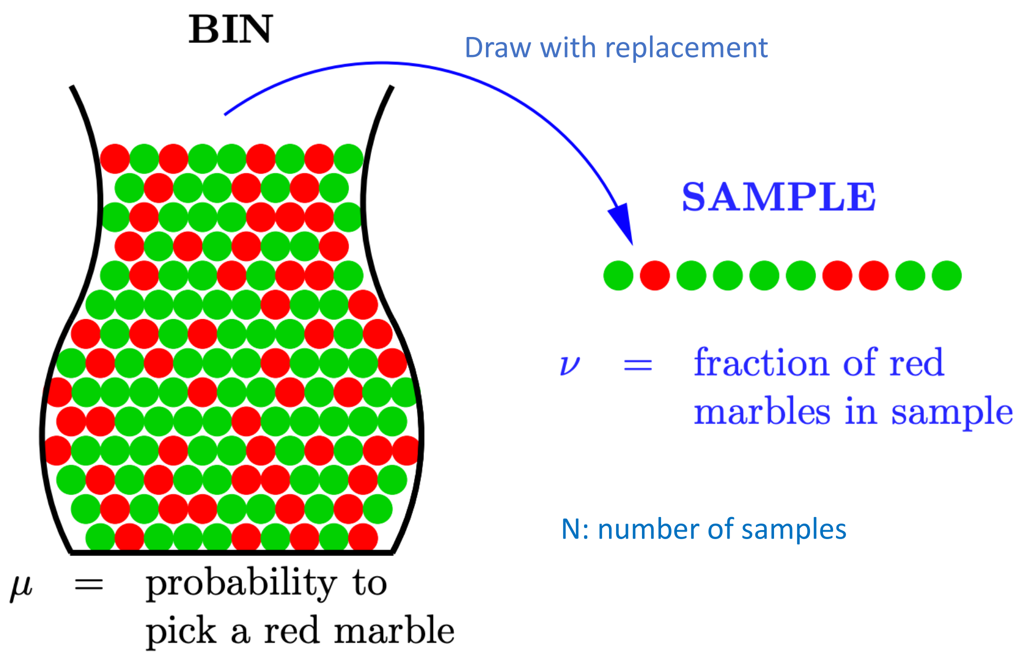
- 60 minutes per session; In-person office hours are highlighted in orange
- Please follow **Piazza** for additional information (location, zoom link, etc)
- Recommendation: Try to utilize the office hour early (way ahead of deadlines), you are likely to get more of TAs' time this way

Recap



Key assumption in machine learning
UNKNOWN TARGET FUNCTION
$f : \mathcal{X} \mapsto \mathcal{Y}$
$y_n = f(\mathbf{x}_n)$
UNKNOWN INPUT DISTRIBUTION
$P(\mathbf{x})$
TRAINING EXAMPLES
$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)$
$\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$
$\mathbf{x}$
$g(\mathbf{x}) \approx f(\mathbf{x})$
LEARNING ALGORITHM
$\mathcal{A}$
FINAL HYPOTHESIS
g
HYPOTHESIS SET
$\mathcal{H}$

Hoeffding's Inequality



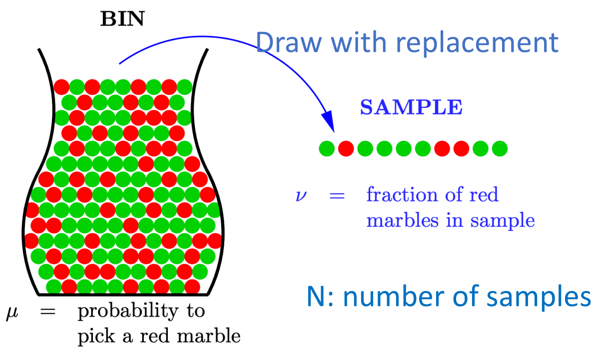
$$\mathbf{Pr}[|\mu - \nu| > \epsilon] \leq 2e^{-2\epsilon^2 N}$$

Define $\delta = \Pr[|\mu - \nu| > \epsilon]$

- Fix δ, ϵ decreases as N increases
- Fix ϵ, δ decreases as N increases
- Fix N, δ decreases as ϵ increases

Informal intuitions of notations
N: # sample
δ: probability of "bad" event
ϵ: error of estimation

Connection to Learning



- Given dataset $D = \{(\vec{x}_1, y_1), \dots, (\vec{x}_N, y_N)\}$.
- Fix a hypothesis h
  - $\boldsymbol{E_{in}(h)} \stackrel{\text{def}}{=} \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}[h(\vec{x}_n) \neq f(\vec{x}_n)]$ [In-sample error, analogy to ν]
  - $\boldsymbol{E_{out}(h)} \stackrel{\text{def}}{=} \Pr_{\vec{x} \sim P(\vec{x})}[h(\vec{x}) \neq f(\vec{x})]$ [Out-of-sample error, analogy to μ]
- Apply Hoeffding's inequality

$$\boldsymbol{Pr[|E_{out}(h) - E_{in}(h)| > \epsilon] \leq 2e^{-2\epsilon^2 N}}$$

- This is *verification*, not *learning*

Connection to "Real" Learning

- Given a finite hypothesis set $H = \{h_1, \dots, h_M\}$
- Apply some learning algorithm on D, output a $g \in H$

- What can we say about $E_{out}(g)$ from $E_{in}(g)$?

$$\boldsymbol{Pr[|E_{out}(g) - E_{in}(g)| > \epsilon] \leq 2Me^{-2\epsilon^2 N}} \quad \textbf{for any } \boldsymbol{\epsilon > 0}$$

<u>Intuitions:</u>

1. Bad event $B(g) \subseteq B(h_1) \cup B(h_2) \dots \cup B(h_M)$

   g is selected within $\{h_1, \dots, h_M\}$
   => bad event of g is within the union of the bad events of $h_1, \dots, h_M$

2. $\Pr[B(g)] \leq \Pr[B(h_1)] + \cdots + \Pr[B(h_M)]$

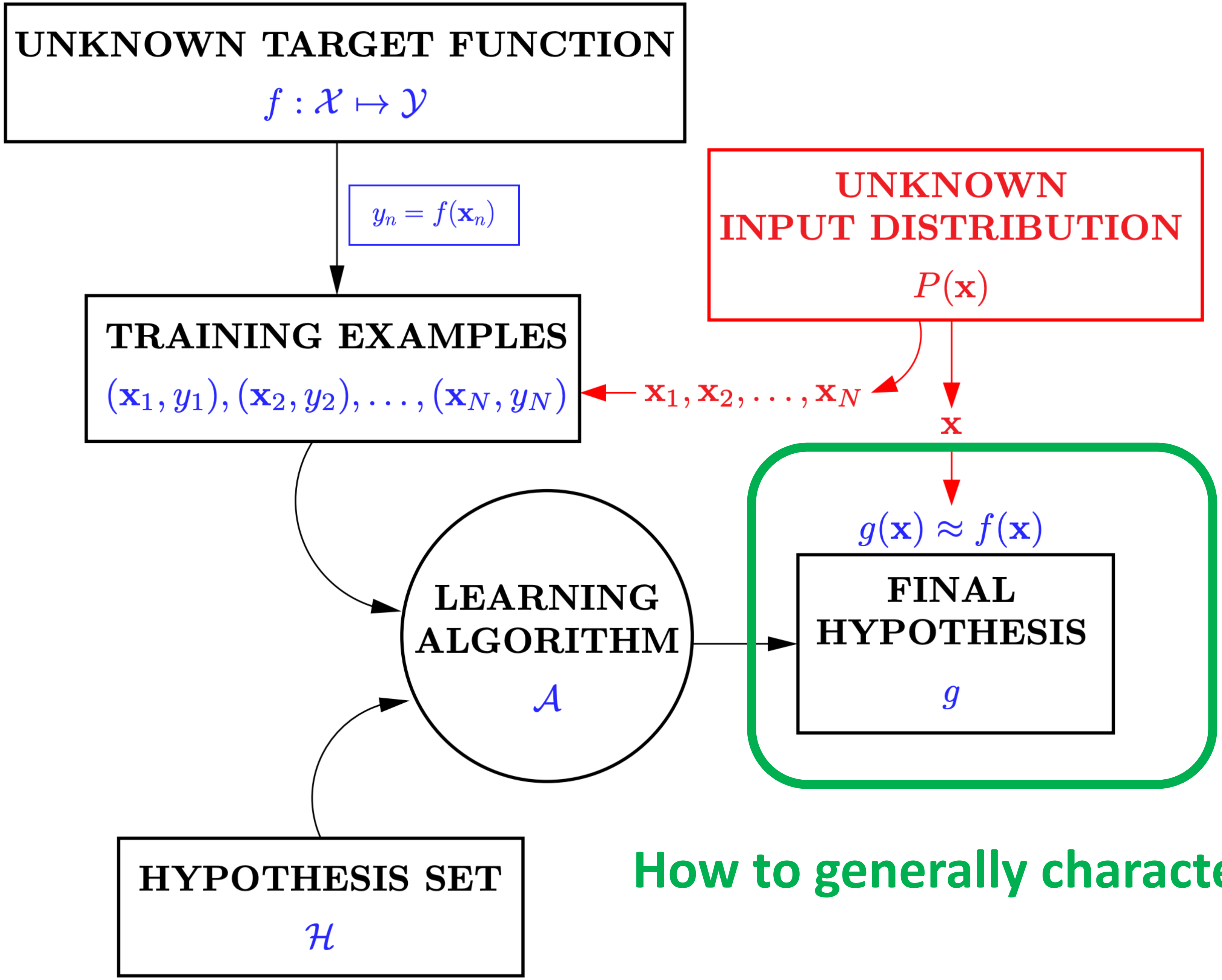
   each of the $\Pr[B(h_m)]$ follows Hoeffding's inequality

Today’s Lecture

The notes are not intended to be comprehensive. They should be accompanied by lectures and/or textbook.

Let me know if you spot errors.

Revisit the learning problem



How to generally characterize $g \approx f$

Goal: $g \approx f$

- A general approach:
  - Define an error function $E(h, f)$ that quantify how far away h is to f
  - choose $g = \underset{h \in \mathcal{H}}{\operatorname{argmin}} E(h, f)$
- A major component of ML is optimization
- E is usually defined in terms of a pointwise error function $e(h(\vec{x}), f(\vec{x}))$
  - Binary error (classification): $e(h(\vec{x}), f(\vec{x})) = \mathbb{I}[h(\vec{x}_n) \neq f(\vec{x}_n)]$
  - Squared error (regression): $e(h(\vec{x}), f(\vec{x})) = \big(f(\vec{x}) - h(\vec{x})\big)^2$

$E_{in}(h) = \frac{1}{N}\sum_{n=1}^{N} e(h(\vec{x}_n), f(\vec{x}_n))$
$E_{out}(h) = \mathbb{E}_{\vec{x}}[e(h(\vec{x}), f(\vec{x}))]$

The discussion on the Hoeffding's inequality applies for general (bounded) error functions.

How to choose the error function?

- Consideration 1: Properties of domain applications
- Example: Fingerprint recognition
  - Input: fingerprints
  - Outputs: whether the person is authorized

		$f(\vec{x})$	
		+1	-1
$h(\vec{x})$	+1	No error	False positive
	-1	False negative	No error

Supermarket

		$f(\vec{x})$	
		$+1$	-1
$h(\vec{x})$	$+1$	0	Small
	-1	Large	0

FBI

		$f(\vec{x})$	
		$+1$	-1
$h(\vec{x})$	$+1$	0	Large
	-1	Small	0

How to choose the error function?

- Consideration 1: Properties of application problems

- Consideration 2: Computation
  - ML algorithms are essentially performing optimization (finding g with smallest error)

$$g = \operatorname*{argmin}_{h \in \mathcal{H}} E(h, f)$$

  - Choose the error that is “easier” to optimize
    - e.g., if the error function is continuous, differentiable, and convex, we usually have efficient algorithms

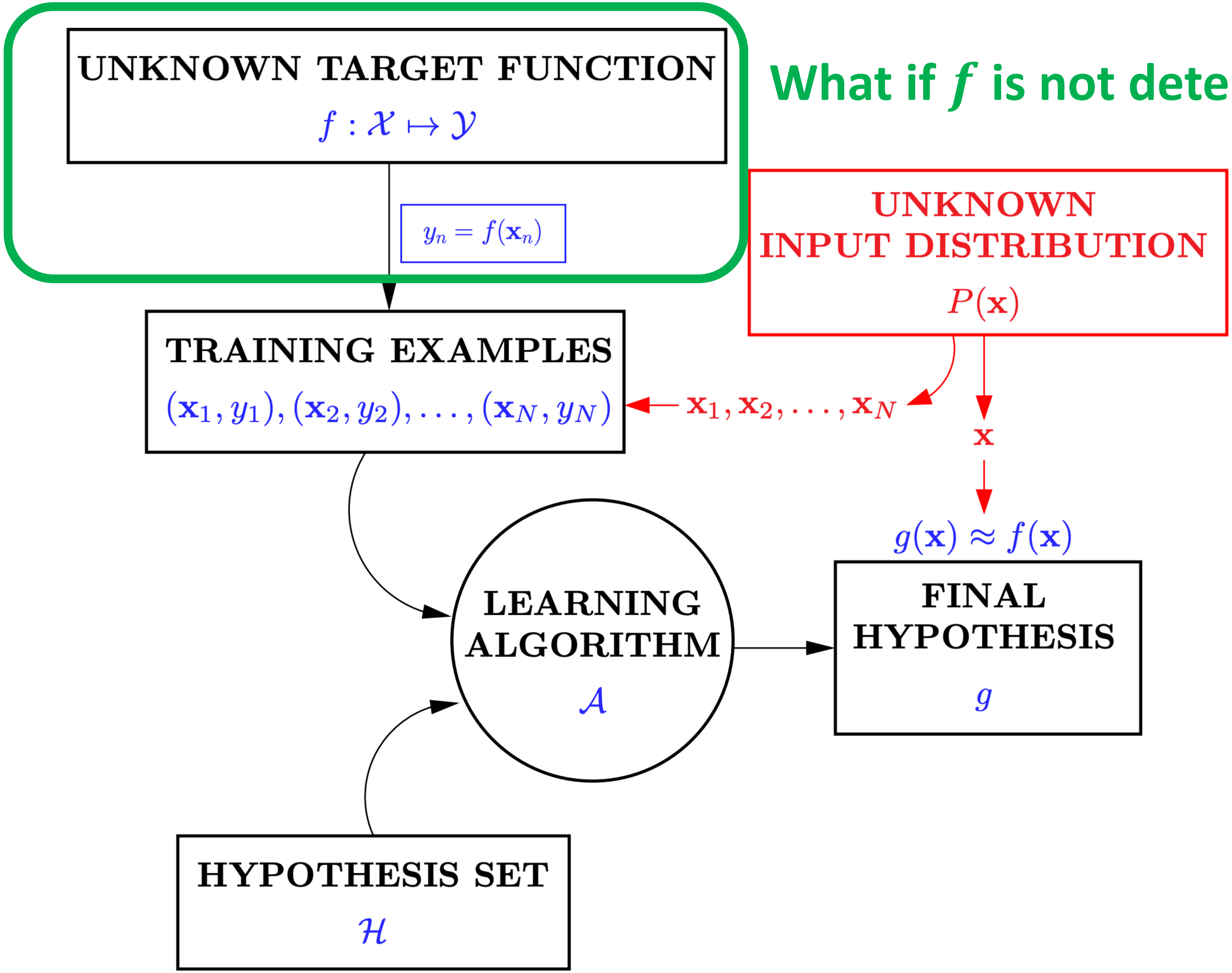
How to choose the error function?

- Consideration 1: Properties of application problems

- Consideration 2: Computation

- Specifying the error function is part of setting up the learning problem
  - It impacts what you eventually learn

What if f is not deterministic?



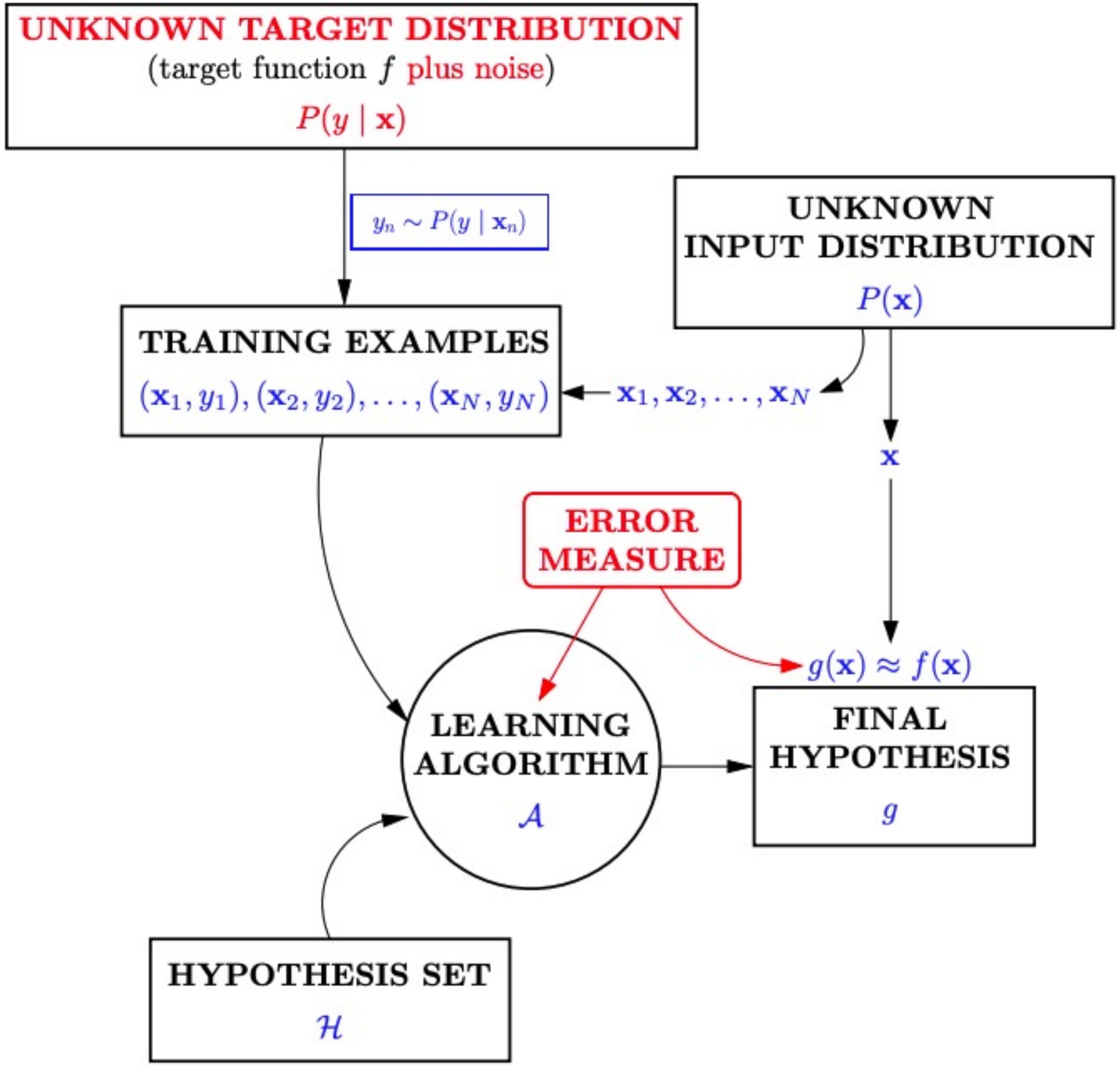
Noisy Target

- What if there doesn't exist f such that $y = f(\vec{x})$?
  - f is stochastic instead of deterministic
  - (even if two customers have exactly the same attributes, one might be a good customer for bank, and the other might not be)

- Common approach
  - Instead of a target function, define a target **distribution**
  - Instead of $y = f(\vec{x})$, y is drawn from a conditional distribution $P(y|\vec{x})$
  - $y = f(\vec{x}) + \epsilon$
    - $f(\vec{x})$ is the mean of the distribution $\mathbb{E}[y|\vec{x}]$
    - ϵ is zero-mean noise $y - \mathbb{E}[y|\vec{x}]$

The discussion on the Hoeffding's inequality applies for noisy targets.

General Setup of (Supervised) Learning



Theory of Generalization

Revisit the "Multi-Hypothesis" Bound

- Given a finite hypothesis set $H = \{h_1, \dots, h_M\}$
- Apply some learning algorithm on D, output a $g \in H$

- What can we say about $E_{out}(g)$ from $E_{in}(g)$?

$$\boldsymbol{Pr[|E_{out}(g) - E_{in}(g)| > \epsilon] \leq 2Me^{-2\epsilon^2 N}} \quad \text{for any } \boldsymbol{\epsilon > 0}$$

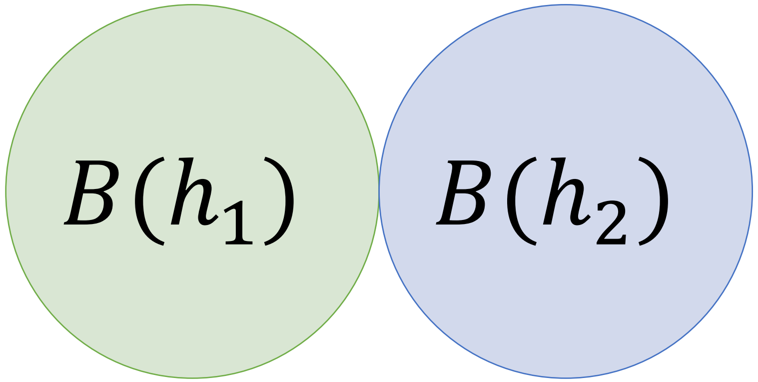
What if M is infinite?

$Pr[|E_{out}(g) - E_{in}(g)| > \epsilon] \leq 2Me^{-2\epsilon^2 N}$ don't seem to carry any meanings

Key Intuitions in the Multi-Hypothesis Analysis

- Define ”bad event of h” $B(h)$ *as* $|E_{out}(h) - E_{in}(h)| > \epsilon$
- If g is selected from $\{h_1, h_2\}$
  - $B(g) \subseteq B(h_1) \cup B(h_2)$
  - $\Pr[B(g)] \leq \Pr[B(h_1) \text{ or } B(h_2)]$

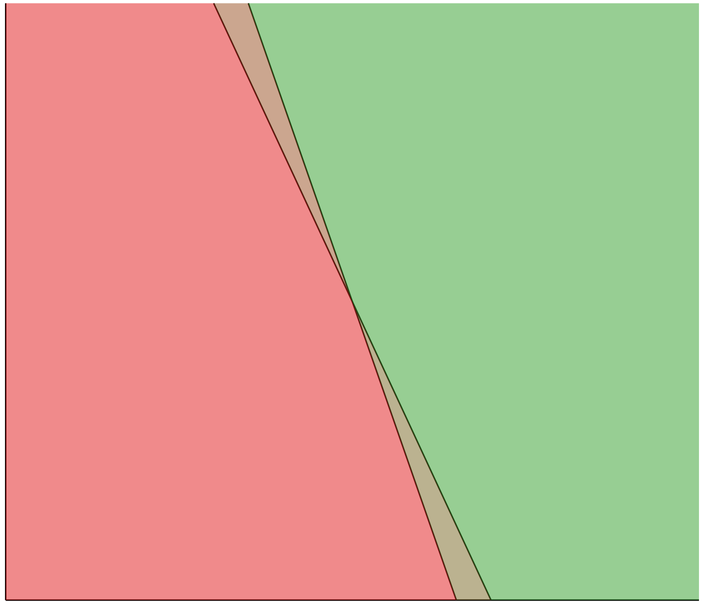
$$\leq \Pr[B(h_1)] + \Pr[B(h_2)]$$ (Union Bound)



  - Union bound considers the worst case: Bad events don’t overlap

Do Bad Events Overlap?

- Oftentimes, they overlap a lot!



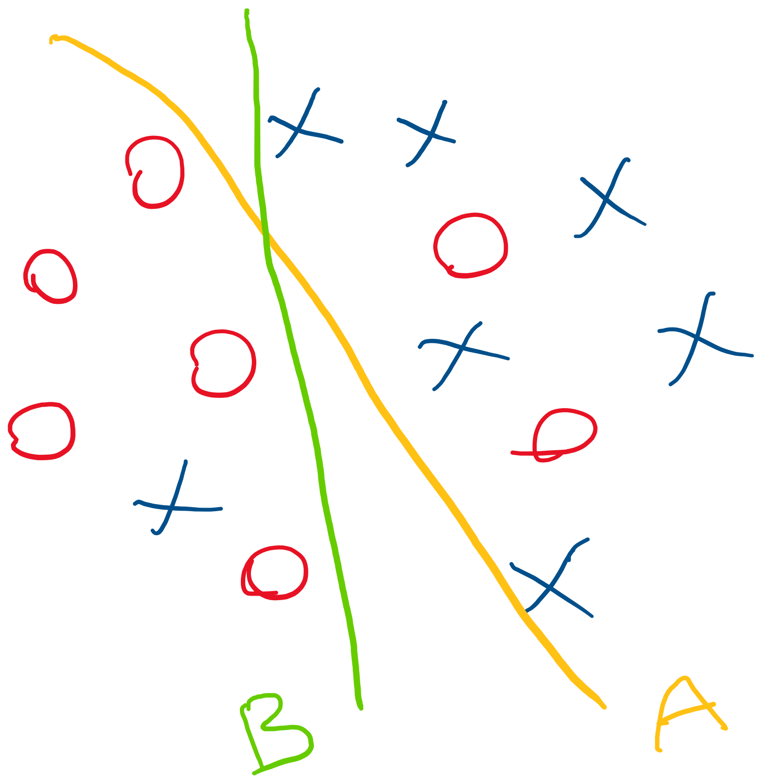
The two linear separators on the left make the same predictions for most points.

If it's a bad event for one, it's likely to be a bad event for the other.

"bad event of h" $B(h)$: $|E_{out}(h) - E_{in}(h)| > \epsilon$

Recall: Informally, you can interpret "bad event of h" as the event that we draw a "unrepresentative dataset D" that makes the in-sample errors of h to be far away from out-of-sample error of h

What Can We Do?



For this dataset,
any difference between A and B?

For this dataset, probably no difference.

They make the same prediction for every data point in this dataset.

What Can We Do?

- Let’s define “data-dependent” hypothesis, call it dichotomy.

> di·chot·o·my
> /dīˈkädəmē/
>
> *noun*
>
> a division or contrast between two things that are or are represented as being opposed or entirely different.
> "a rigid **dichotomy between** science and mysticism"

- A hypothesis $h: X \rightarrow \{-1, +1\}$
- A dichotomy for a set of data points $(\vec{x}_1, \dots, \vec{x}_N)$:
  - Assign either +1 or -1 for each of the data points
    (divide the data points into two groups)

- Why dichotomies?
  - It helps us count “effective number of hypothesis” (to replace M)

More Formal Definitions

- Dichotomies
  - Informally, consider a dichotomy as a "data-dependent" hypothesis
  - Characterized by both hypothesis set H and N data points $(\vec{x}_1, \dots, \vec{x}_N)$

$$H(\vec{x}_1, \dots \vec{x}_N) = \{(h(\vec{x}_1), \dots, h(\vec{x}_N)) | h \in H\}$$

  - The set of possible prediction combinations $h \in H$ can induce on $\vec{x}_1, \dots, \vec{x}_N$

- Growth function
  - Largest number of dichotomies H can induce across all possible data sets of size N

$$m_H(N) = \max_{(\vec{x}_1, \dots, \vec{x}_N)} |H(\vec{x}_1, \dots, \vec{x}_N)|$$

Example: H = Positive Rays

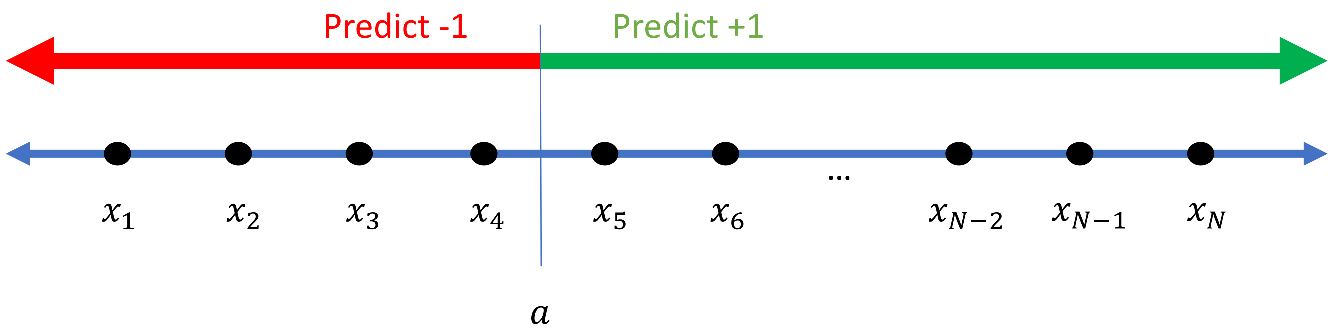
- Dichotomies
  - Informally, consider a dichotomy as a “data-dependent” hypothesis
  - Characterized by both hypothesis set H and N data points $(\vec{x}_1, \ldots, \vec{x}_N)$

$$H(\vec{x}_1, \ldots \vec{x}_N) = \{(h(\vec{x}_1), \ldots, h(\vec{x}_N)) | h \in H\}$$

  - The set of possible prediction combinations $h \in H$ can induce on $\vec{x}_1, \ldots, \vec{x}_N$
- Growth function
  - Largest number of dichotomies H can induce across all possible data sets of size N

$$m_H(N) = \max_{(\vec{x}_1, \ldots, \vec{x}_N)} |H(\vec{x}_1, \ldots, \vec{x}_N)|$$

- Data points are in one-dimensional space
- Positive rays: $h(x) = sign(x - a)$



- What is $H(\vec{x}_1, \ldots, \vec{x}_N)$?
- What is $m_H(N)$?

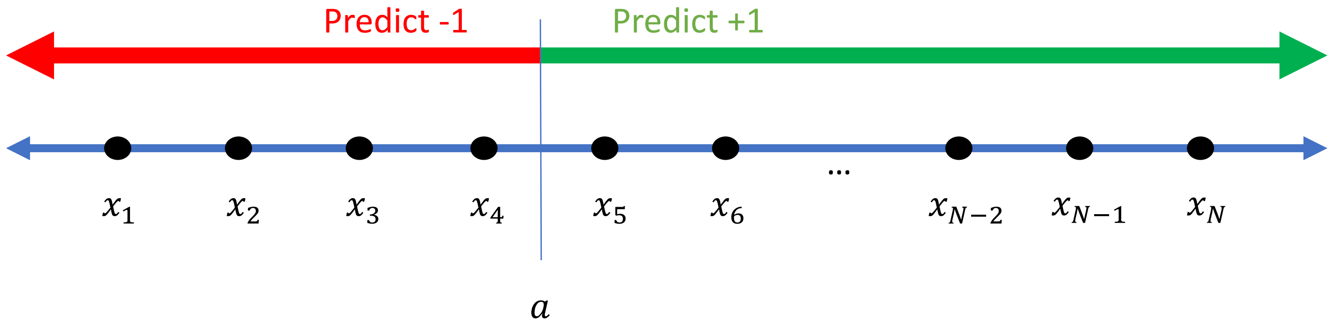
Example: H = Positive Rays

- Dichotomies
  - Informally, consider a dichotomy as a "data-dependent" hypothesis
  - Characterized by both hypothesis set H and N data points $(\vec{x}_1, \dots, \vec{x}_N)$

    $$H(\vec{x}_1, \dots \vec{x}_N) = \{(h(\vec{x}_1), \dots, h(\vec{x}_N)) | h \in H\}$$
  - The set of possible prediction combinations $h \in H$ can induce on $\vec{x}_1, \dots, \vec{x}_N$
- Growth function
  - Largest number of dichotomies H can induce across all possible data sets of size N

    $$m_H(N) = \max_{(\vec{x}_1, \dots, \vec{x}_N)} |H(\vec{x}_1, \dots, \vec{x}_N)|$$

- Data points are in one-dimensional space
- Positive rays: $h(x) = sign(x - a)$



- What is $H(\vec{x}_1, \dots, \vec{x}_N)$?

$$\begin{aligned} H(\vec{x}_1, \dots, \vec{x}_N) = \{&(+1, +1, \dots, +1), \\ &(-1, +1, \dots, +1), \\ &\dots \\ &(-1, -1, \dots, -1)\} \end{aligned}$$

- What is $m_H(N)$?

$$m_H(N) = N + 1$$

What is $m_H(N)$?

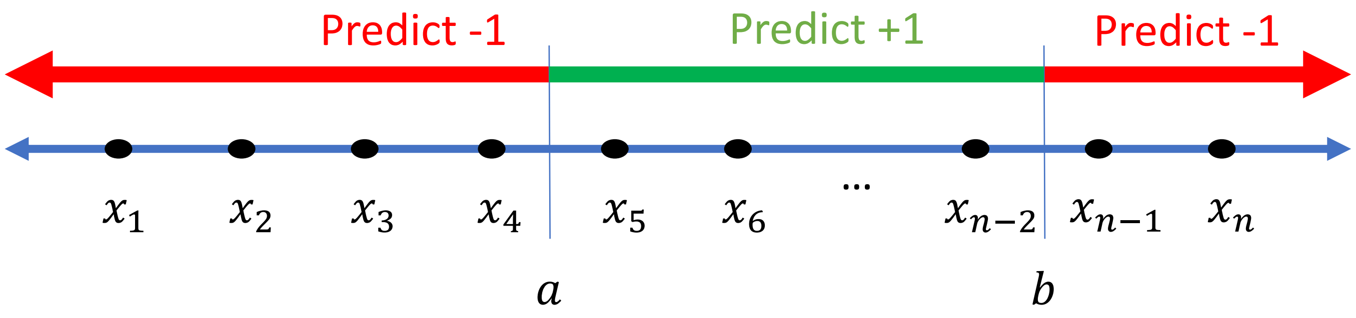
- Dichotomies
  - Informally, consider a dichotomy as a "data-dependent" hypothesis
  - Characterized by both hypothesis set H and N data points $(\vec{x}_1, \dots, \vec{x}_N)$

$$H(\vec{x}_1, \dots \vec{x}_N) = \{(h(\vec{x}_1), \dots, h(\vec{x}_N)) | h \in H\}$$

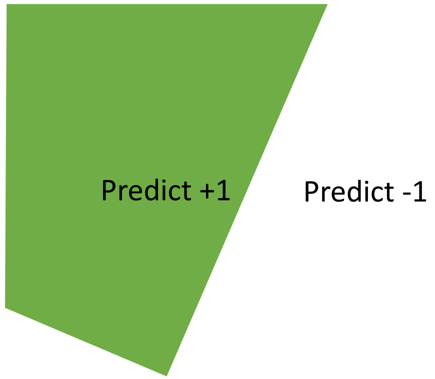
  - The set of possible prediction combinations $h \in H$ can induce on $\vec{x}_1, \dots, \vec{x}_N$
- Growth function
  - Largest number of dichotomies H can induce across all possible data sets of size N

$$m_H(N) = \max_{(\vec{x}_1, \dots, \vec{x}_N)} |H(\vec{x}_1, \dots, \vec{x}_N)|$$

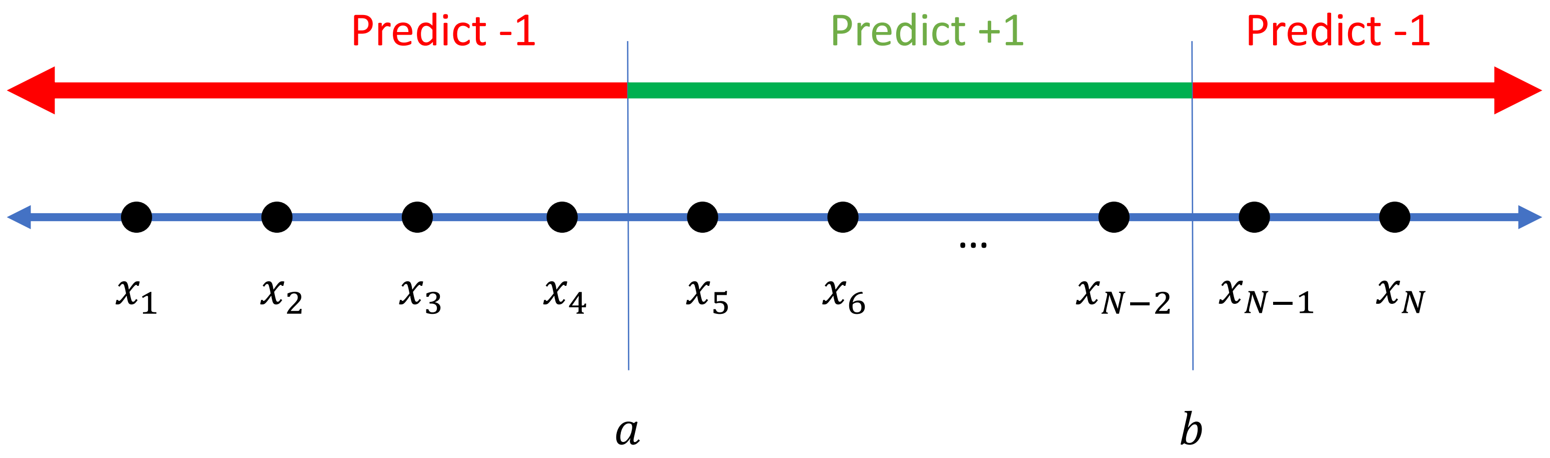
- H = Positive Intervals
  - Data points are in one-dimensional space
  - Choose two thresholds. Predict +1 within the interval, -1 outside



- H = Convex Sets
  - Data points are in 2-dimensional space
  - Hypothesis is represented by a convex set

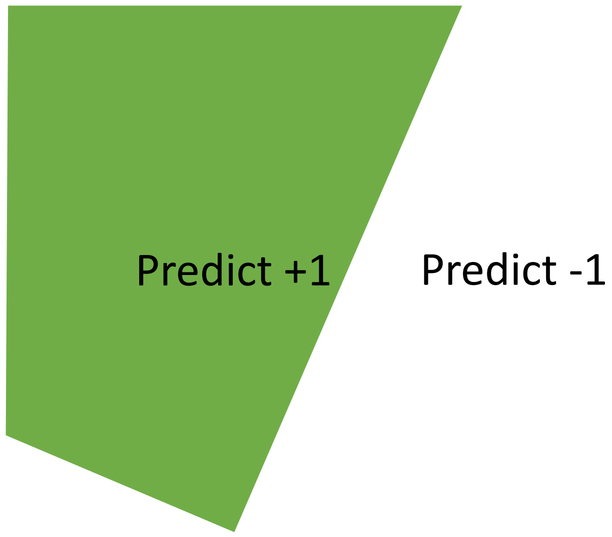


Example: H = Positive Intervals



- What is $m_H(N)$?
  - $m_H(N) = \binom{N+1}{2} + 1 = \frac{N^2}{2} + \frac{N}{2} + 1$

Example: H = Convex Sets



- What is $m_H(N)$?
  - $m_H(N) = 2^N$

Note:
$m_H(N) \leq 2^N$ for all H and all N
(There are only 2^N possible label combinations for N $points$)

Why Growth Function?

- Growth function $m_H(N)$
  - Largest number of "effective" hypothesis H can induce on N data points
  - A more precise "complexity" measure for H
  - Goal: Replace M in finite-hypothesis analysis with $m_H(N)$
    - With prob $1-\delta$, $E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{1}{2N} ln \frac{2M}{\delta}}$

- Theorem: VC Inequality (1971)

  With prob $1-\delta$

$$E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} ln \frac{4m_H(2N)}{\delta}}$$

Growth Functions for Other H

- H = 2-D Perceptron
  - What is $m_H(3)$
  - What is $m_H(4)$

- Dichotomies
  - Informally, consider a dichotomy as a “data-dependent” hypothesis
  - Characterized by both hypothesis set H and N data points $(\vec{x}_1, \dots, \vec{x}_N)$

$$H(\vec{x}_1, \dots \vec{x}_N) = \{(h(\vec{x}_1), \dots, h(\vec{x}_N)) | h \in H\}$$

  - The set of possible prediction combinations $h \in H$ can induce on $\vec{x}_1, \dots, \vec{x}_N$

- Growth function
  - Largest number of dichotomies H can induce across all possible data sets of size N

$$m_H(N) = \max_{(\vec{x}_1, \dots, \vec{x}_N)} |H(\vec{x}_1, \dots, \vec{x}_N)|$$

Growth Functions for Other H

- H = 2-D Perceptron
  - What is $m_H(3)$
  - What is $m_H(4)$

- <u>Dichotomies</u>
  - Informally, consider a dichotomy as a “data-dependent” hypothesis
  - Characterized by both hypothesis set H and N data points $(\vec{x}_1, \dots, \vec{x}_N)$

$$H(\vec{x}_1, \dots \vec{x}_N) = \{(h(\vec{x}_1), \dots, h(\vec{x}_N)) | h \in H\}$$

  - The set of possible prediction combinations $h \in H$ can induce on $\vec{x}_1, \dots, \vec{x}_N$

- <u>Growth function</u>
  - Largest number of dichotomies H can induce across all possible data sets of size N

$$m_H(N) = \max_{(\vec{x}_1, \dots, \vec{x}_N)} |H(\vec{x}_1, \dots, \vec{x}_N)|$$

- Exactly calculating the growth function is generally hard!

- Goal: “bound” the growth function using some proxy

Bounding Growth Function

- More definitions….
  - Shatter:
    - H **shatters** $(\vec{x}_1, \dots, \vec{x}_N)$ if $|H(\vec{x}_1, \dots, \vec{x}_N)| = 2^N$
    - H can induce all label combinations for $(\vec{x}_1, \dots, \vec{x}_N)$
  - Break point
    - k is a **break point** for H if no data set of size k can be shattered by H
- A peek at the key result (take this as a fact for now)
  - If there are no break points for H, $m_H(N) = 2^N$
  - If k is a break point for H, $m_H(N)$ is polynomial in N.

    In particular, $m_H(N) = O(N^{k-1})$

A bit more accurately:

- $m_H(N) \leq \sum_{i=1}^{k-1} \binom{N}{i}$, or
- $m_H(N) \leq N^{k-1} + 1$

Practice

- Dichotomies
  - Informally, consider a dichotomy as a "data-dependent" hypothesis
  - Characterized by both hypothesis set H and N data points $(\vec{x}_1, \dots, \vec{x}_N)$

$$H(\vec{x}_1, \dots \vec{x}_N) = \{(h(\vec{x}_1), \dots, h(\vec{x}_N)) | h \in H\}$$

  - The set of possible prediction combinations $h \in H$ can induce on $\vec{x}_1, \dots, \vec{x}_N$

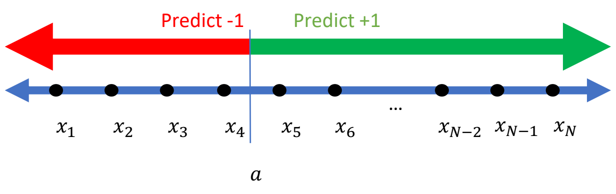
- Growth function
  - Largest number of dichotomies H can induce across all possible data sets of size N

$$m_H(N) = \max_{(\vec{x}_1, \dots, \vec{x}_N)} |H(\vec{x}_1, \dots, \vec{x}_N)|$$

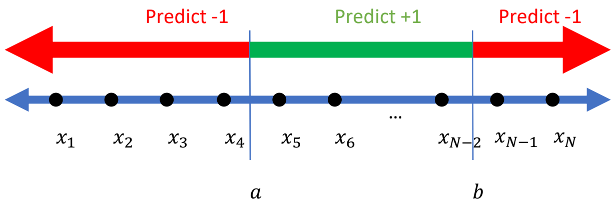
- Shatter:
  - H **shatters** $(\vec{x}_1, \dots, \vec{x}_N)$ if $|H(\vec{x}_1, \dots, \vec{x}_N)| = 2^N$
  - H can induce all label combinations for $(\vec{x}_1, \dots, \vec{x}_N)$
- Break point
  - k is a **break point** for H if no data set of size k can be shattered by H

- What are the break points for

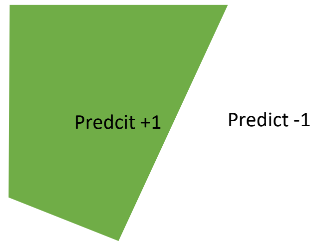
1. Positive Rays



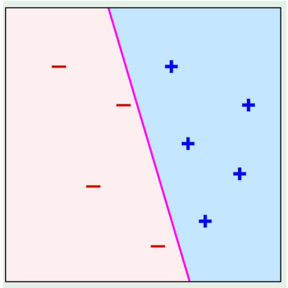
2. Positive Intervals



3. Convex Sets



4. 2-D Perceptron



Practice

- <u>Dichotomies</u>
  - Informally, consider a dichotomy as a "data-dependent" hypothesis
  - Characterized by both hypothesis set H and N data points $(\vec{x}_1, \dots, \vec{x}_N)$

$$H(\vec{x}_1, \dots \vec{x}_N) = \{(h(\vec{x}_1), \dots, h(\vec{x}_N)) | h \in H\}$$

  - The set of possible prediction combinations $h \in H$ can induce on $\vec{x}_1, \dots, \vec{x}_N$

- <u>Growth function</u>
  - Largest number of dichotomies H can induce across all possible data sets of size N

$$m_H(N) = \max_{(\vec{x}_1, \dots, \vec{x}_N)} |H(\vec{x}_1, \dots, \vec{x}_N)|$$

- <u>Shatter:</u>
  - H **shatters** $(\vec{x}_1, \dots, \vec{x}_N)$ if $|H(\vec{x}_1, \dots, \vec{x}_N)| = 2^N$
  - H can induce all label combinations for $(\vec{x}_1, \dots, \vec{x}_N)$
- <u>Break point</u>
  - k is a **break point** for H if no data set of size k can be shattered by H

$\boldsymbol{m_H(N)}$

$\boldsymbol{m_H(N)}$		N=1	N=2	N=3	N=4	N=5	Break Points
$N+1$	Positive Rays						
$\frac{N^2}{2} + \frac{N}{2} + 1$	Positive Intervals						
2^N	Convex Sets						
	2D Perceptron						

Practice

- <u>Dichotomies</u>
  - Informally, consider a dichotomy as a "data-dependent" hypothesis
  - Characterized by both hypothesis set H and N data points $(\vec{x}_1, \dots, \vec{x}_N)$

$$H(\vec{x}_1, \dots \vec{x}_N) = \{(h(\vec{x}_1), \dots, h(\vec{x}_N)) | h \in H\}$$

  - The set of possible prediction combinations $h \in H$ can induce on $\vec{x}_1, \dots, \vec{x}_N$

- <u>Growth function</u>
  - Largest number of dichotomies H can induce across all possible data sets of size N

$$m_H(N) = \max_{(\vec{x}_1, \dots, \vec{x}_N)} |H(\vec{x}_1, \dots, \vec{x}_N)|$$

- <u>Shatter:</u>
  - H **shatters** $(\vec{x}_1, \dots, \vec{x}_N)$ if $|H(\vec{x}_1, \dots, \vec{x}_N)| = 2^N$
  - H can induce all label combinations for $(\vec{x}_1, \dots, \vec{x}_N)$
- <u>Break point</u>
  - k is a **break point** for H if no data set of size k can be shattered by H

$\boldsymbol{m_H(N)}$

$\boldsymbol{m_H(N)}$		N=1	N=2	N=3	N=4	N=5	Break Points
$N+1$	Positive Rays	2	3	4	5	6	$k = 2,3,4, \dots$
$\frac{N^2}{2} + \frac{N}{2} + 1$	Positive Intervals						
2^N	Convex Sets						
	2D Perceptron						

Practice

- <u>Dichotomies</u>
  - Informally, consider a dichotomy as a "data-dependent" hypothesis
  - Characterized by both hypothesis set H and N data points $(\vec{x}_1, \dots, \vec{x}_N)$

$$H(\vec{x}_1, \dots \vec{x}_N) = \{(h(\vec{x}_1), \dots, h(\vec{x}_N)) | h \in H\}$$

  - The set of possible prediction combinations $h \in H$ can induce on $\vec{x}_1, \dots, \vec{x}_N$

- <u>Growth function</u>
  - Largest number of dichotomies H can induce across all possible data sets of size N

$$m_H(N) = \max_{(\vec{x}_1, \dots, \vec{x}_N)} |H(\vec{x}_1, \dots, \vec{x}_N)|$$

- <u>Shatter:</u>
  - H **shatters** $(\vec{x}_1, \dots, \vec{x}_N)$ if $|H(\vec{x}_1, \dots, \vec{x}_N)| = 2^N$
  - H can induce all label combinations for $(\vec{x}_1, \dots, \vec{x}_N)$
- <u>Break point</u>
  - k is a **break point** for H if no data set of size k can be shattered by H

$\boldsymbol{m_H(N)}$

$\boldsymbol{m_H(N)}$		N=1	N=2	N=3	N=4	N=5	Break Points
$N+1$	Positive Rays	2	3	4	5	6	$k = 2,3,4, \dots$
$\frac{N^2}{2} + \frac{N}{2} + 1$	Positive Intervals	2	4	7	11	16	$k = 3,4,5, \dots$
2^N	Convex Sets	2	4	8	16	32	None
	2D Perceptron	2	4	8	14	?	$k = 4,5,6, \dots$

Why Break Points?

- Theorem statement (Again, take it as a fact for now)
  - If there is no break point for H, then $m_H(N) = 2^N$ for all N.
  - If k is a break point for H, i.e., if $m_H(k) < 2^k$ for some value k, then

$$m_H(N) \leq \sum_{i=0}^{k-1} \binom{N}{i}$$

- Rephrase the above theorem
  - If there is no break point for H, then $m_H(N) = 2^N$ for all N.
  - If k is a break point for H, the following statements are true
    - $m_H(N) \leq N^{k-1} + 1$ [Can be proven using induction. See LFD Problem 2.5]
    - $m_H(N) = O(N^{k-1})$
    - $m_H(N)$ is polynomial in N
- We can "bound" the growth function without knowing it exactly.
  - Find break point!

Why Break Points?

- Rephrase the above theorem
  - If there is no break point for H, then $m_H(N) = 2^N$ for all N.
  - If k is a break point for H, the following statements are true
    - $m_H(N) \leq N^{k-1} + 1$ [Can be proven using induction. See LFD Problem 2.5]
    - $m_H(N) = O(N^{k-1})$
    - $m_H(N)$ is polynomial in N

- VC Generalization Bound

  With prob $1 - \delta$

$$E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} ln \frac{4m_H(2N)}{\delta}}$$

- In the following discussion, we treat δ as a constant [i.e., with high probability, the following is true]

$$E_{out}(g) \leq E_{in}(g) + O\left(\sqrt{\frac{1}{N} \ln m_H(N)}\right)$$

[For example, we can set δ to be a small constant, say 0.01. Then every time we wrote the above inequality, we mean that it is true with probability at least 99%.]

Applying Break Points in VC Bound

- VC Bound:

$$E_{out}(g) \le E_{in}(g) + O\left(\sqrt{\frac{1}{N}\ln m_H(N)}\right)$$

➕

- Rephrase the above theorem
  - If there is no break point for H, then $m_H(N) = 2^N$ for all N.
  - If k is a break point for H, the following statements are true
    - $m_H(N) \le N^{k-1} + 1$ [Can be proven using induction. See LFD Problem 2.5]
    - $m_H(N) = O(N^{k-1})$
    - $m_H(N)$ is polynomial in N

- If there are no break point ($m_H(N) = 2^N$)

$$E_{out}(g) \le E_{in}(g) + O(1)$$

(This implies that we can't infer E_{out} from E_{in} even when $N \to \infty$)

- If k is a break point for H, i.e., $m_H(N) = O(N^{k-1})$

$$E_{out}(g) \le E_{in}(g) + O\left(\sqrt{(k-1)\frac{\ln N}{N}}\right)$$