CSE 417T
Introduction to Machine Learning

Lecture 4

Instructor: Chien-Ju (CJ) Ho

Logistics: Homework 1

- Due: **Feb 14 (Monday), 2022**
  - http://chienjuho.com/courses/cse417t/hw1.pdf
  - Intended deadline: Feb 10.
    - Recommend to work on it early to spare time for homework 2

  - Two submission links: Report and Code
    - 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	11:30am (Herbert Zhou)	4pm (Dean Yu)	
Tuesday	1pm (Ziqi Xu)	3:30pm (Neal Huang)	
Wednesday	1pm (Eddie Choi)	4:30pm (Weiwei Ma)	
Thursday	10am (Jackie Zhong)	3pm (Fankun Zeng)	
Friday	8am (Shohaib Shaffiey)	1pm (Yunfan Wang)	7pm (Hao Qin)
Sunday	1pm (Jonathan Ma)		

- 60 minutes per session
- Please follow **Piazza** for additional information
- Recommendation: Try to utilize the office hour early (way ahead of deadlines), you are likely to get more of TAs’ time this way

Recap

Common Notations

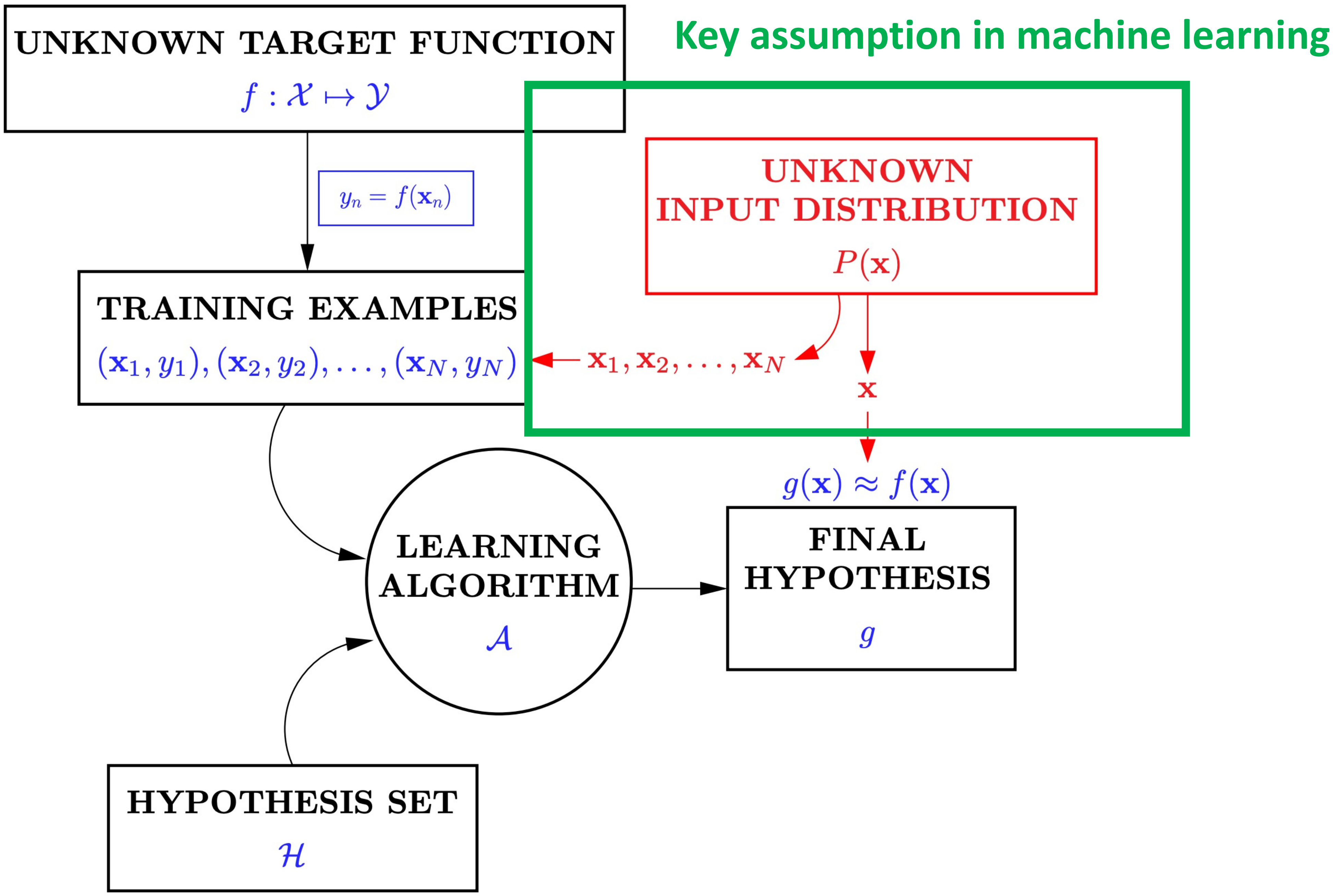
Note that by default, $\vec{x}$ is a column vector.

More formally, we should write $\vec{x} = \begin{bmatrix} x_0 \\ \vdots \\ x_d \end{bmatrix}$.

For convenience, I usually write
$\vec{x} = (x_0, \ldots, x_d)$.

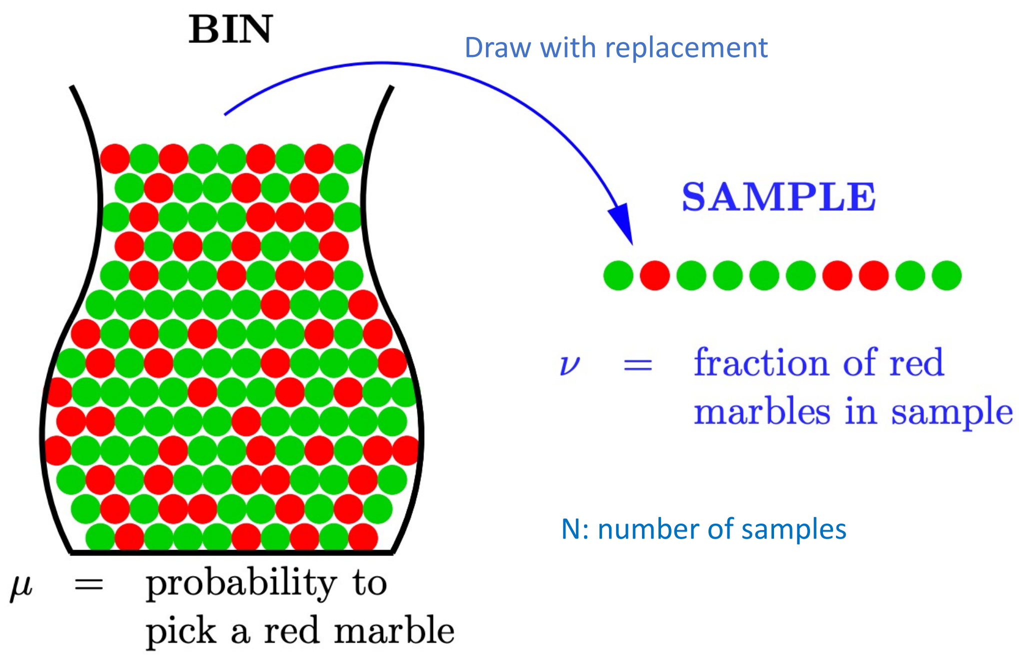
- Data point with augmented x_0: $\vec{x} = (x_0, \ldots, x_d)$
  - We often use d to specify the dimensions of data points
  - We augment $x_0 = 1$ for each data point (Check Lecture 1 for the reasoning)
- Dataset: $D = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_N, y_N)\}$
  - We often use N to specify the number of data points in the dataset
- Hypothesis set H
  - We use $h \in H$ to specify an arbitrary hypothesis
  - We use $g \in H$ to specify the hypothesis output by the learning algorithm
- Indicator variable:
  - $\mathbb{I}[\text{event}] = \begin{cases} 1 & \text{if event is true} \\ 0 & \text{if event is false} \end{cases}$

Example: $\mathbb{I}[h(\vec{x}) \neq f(\vec{x})] = \begin{cases} 1 & \text{if } h(\vec{x}) \neq f(\vec{x}) \\ 0 & \text{if } h(\vec{x}) = f(\vec{x}) \end{cases}$



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



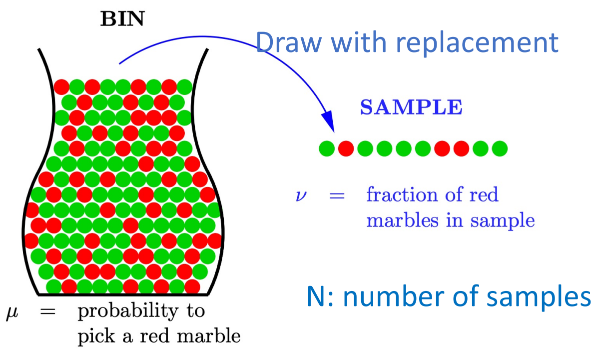
$$\Pr[|\mu - \nu| > \epsilon] \leq 2e^{-2\epsilon^2 N}$$

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
  - $E_{in}(h) \stackrel{\text{def}}{=} \frac{1}{N}\sum_{n=1}^{N} \mathbb{I}[h(\vec{x}_n) \neq f(\vec{x}_n)]$ [In-sample error, analogy to ν]
  - $E_{out}(h) \stackrel{\text{def}}{=} \Pr_{\vec{x}\sim P(\vec{x})}[h(\vec{x}) \neq f(\vec{x})]$ [Out-of-sample error, analogy to μ]
- Apply Hoeffding's inequality

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

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

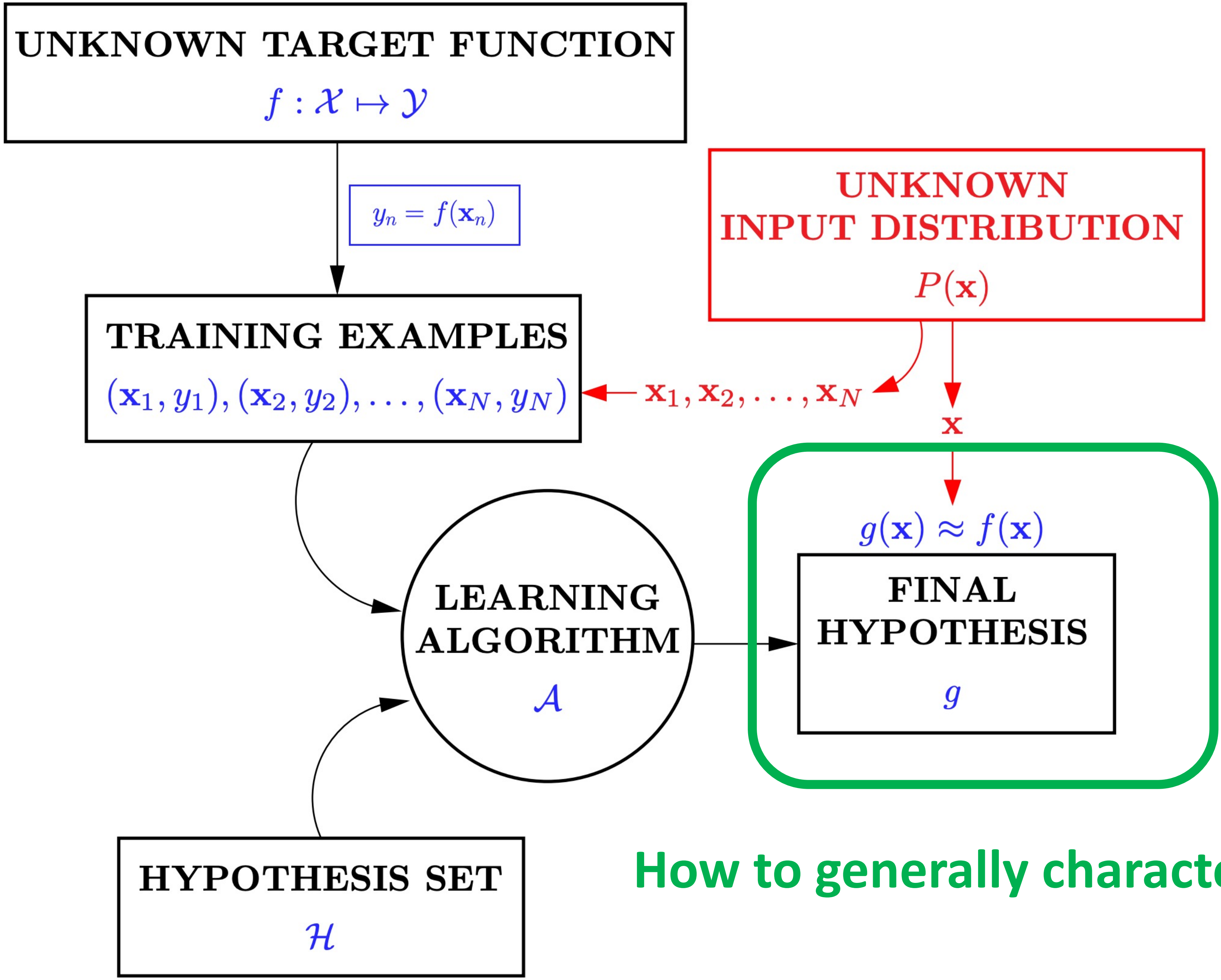
   each of the $\Pr[B(h_m)]$ follows Hoeffding's inequality

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): $e(h(\vec{x}), f(\vec{x})) = \left(f(\vec{x}) - h(\vec{x})\right)^2$

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

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)

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

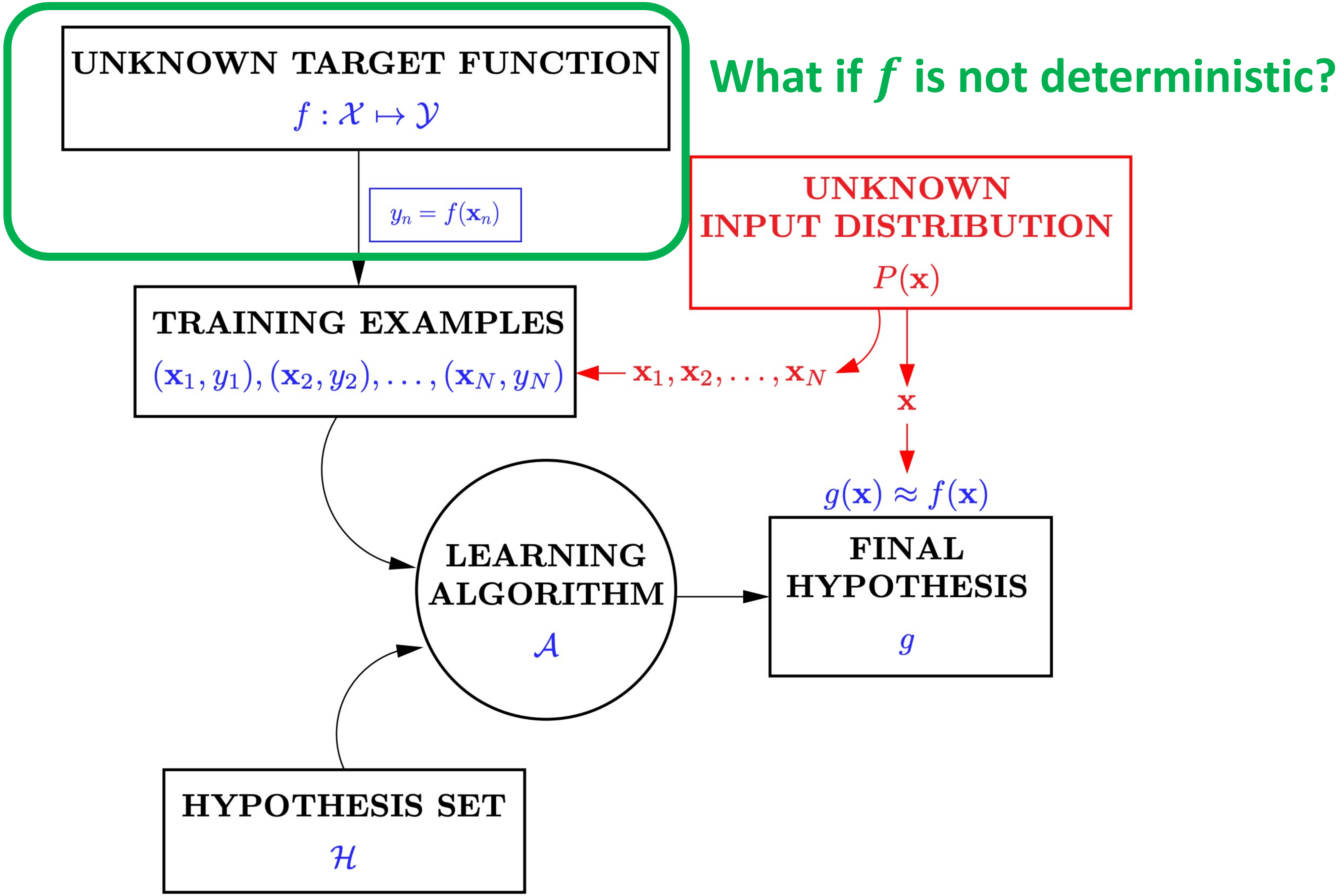
  - Choose the error that is “easier” to optimize
    - e.g., if the error function is continuous, differentiable, and convex, we usually have efficient algorithms

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



UNKNOWN TARGET FUNCTION
$f : \mathcal{X} \mapsto \mathcal{Y}$
$y_n = f(\mathbf{x}_n)$
What if f is not deterministic?
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
$\mathcal{H}$

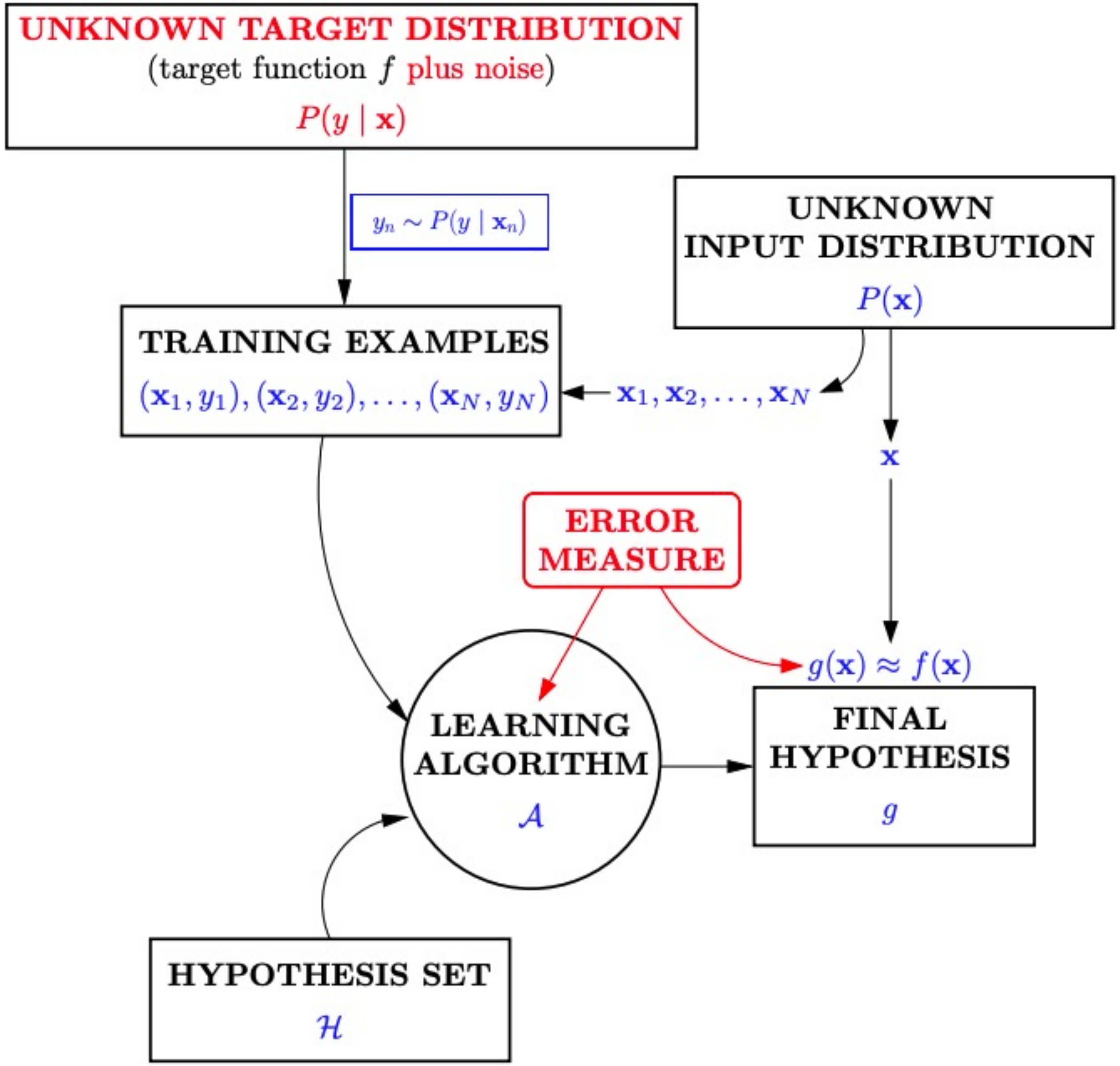
Noisy Target

- 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 $E[y|\vec{x}]$
    - ϵ is zero-mean noise $y - E[y|\vec{x}]$

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 $E_{in}(g)$?

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

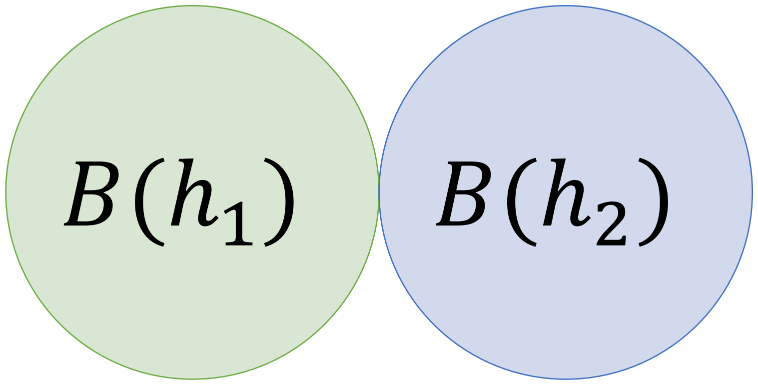
What if M is infinite?

$Pr[|E_{out}(g) - E_{in}(g)| > \epsilon] \leq 2Me^{-2\epsilon^2 N}$ 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 $\{h_1, h_2\}$
    - $B(g) \subseteq B(h_1) \cup B(h_2)$
    - $\Pr[B(g)] \le \Pr[B(h_1) \text{ or } B(h_2)]$

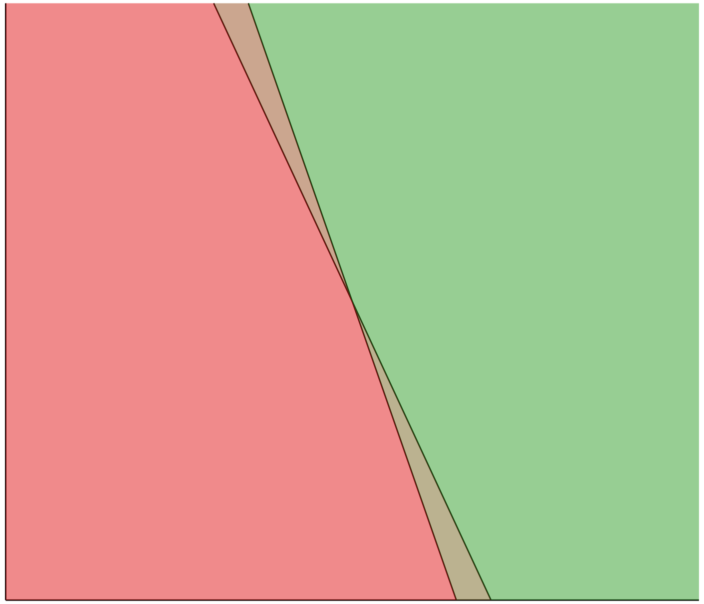
$$\le \Pr[B(h_1)] + \Pr[B(h_2)] \qquad \text{(Union Bound)}$$



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

Do Bad Events Overlap?

- Oftentimes, they overlap a lot!



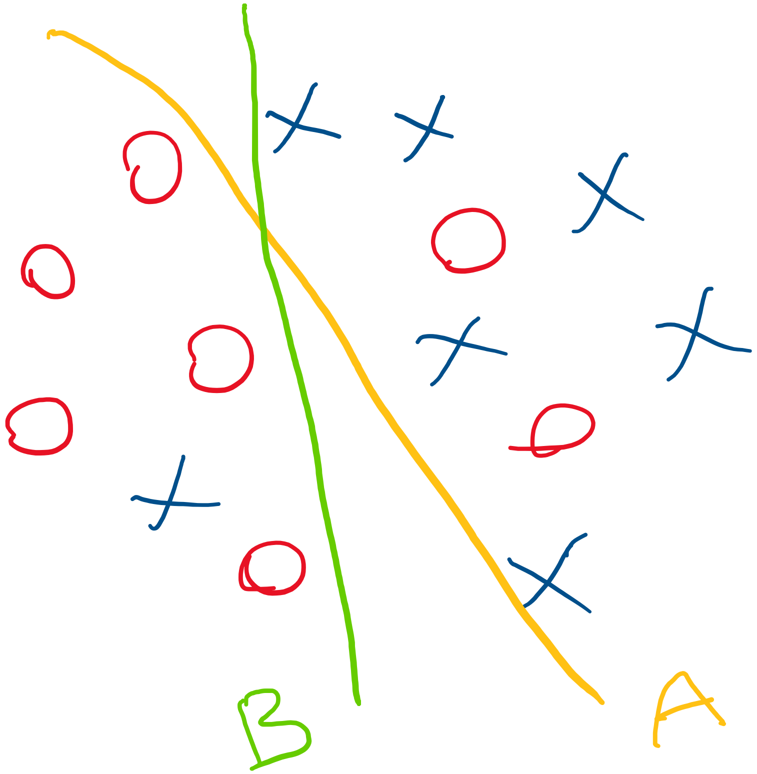
The two linear separators on the left make the same predictions for most points.

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 not.

They make the same predictions 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 $(\vec{x}_1, \dots, \vec{x}_N)$

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

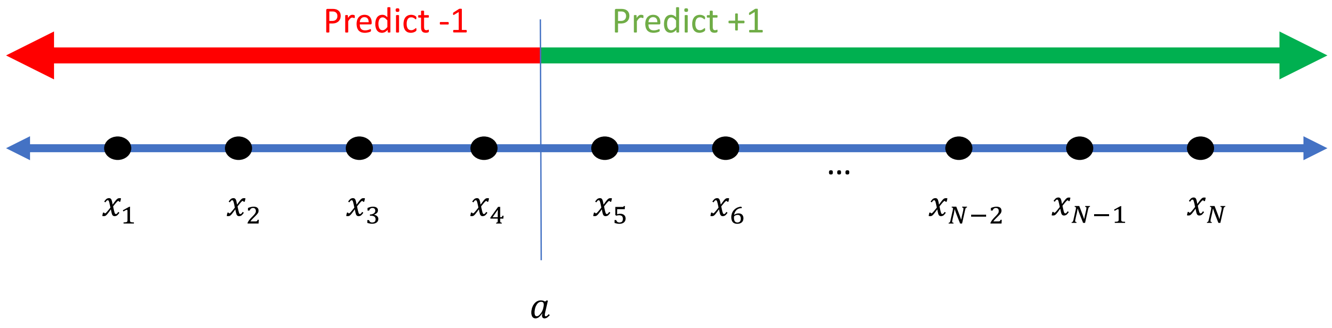
  - The set of possible prediction combinations $h \in H$ can induce on $\vec{x}_1, \dots, \vec{x}_N$

- 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

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



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

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

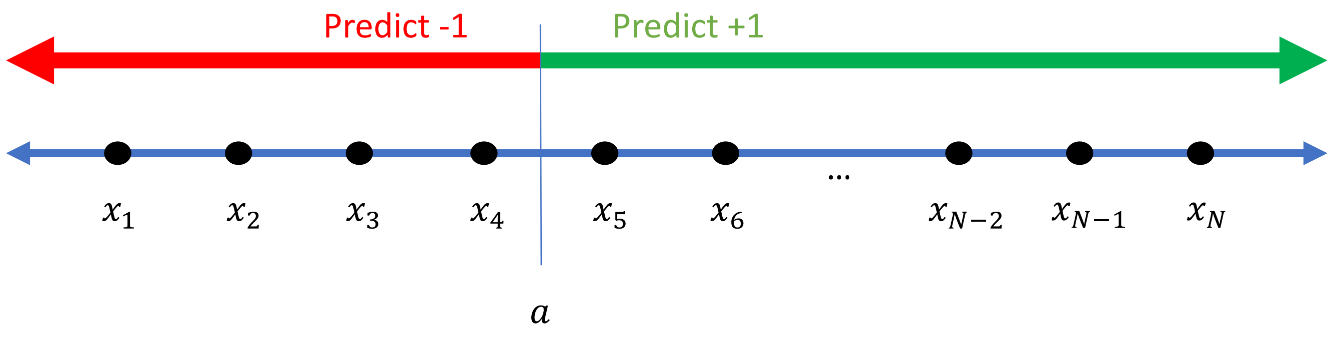
$$H(\vec{x}_1, \dots \vec{x}_N) = \{(h(\vec{x}_1), \dots, h(\vec{x}_N)) | h \in H\}$$

  - 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

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

Example: H = Positive Rays

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



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

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

- What is $m_H(N)$?

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

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

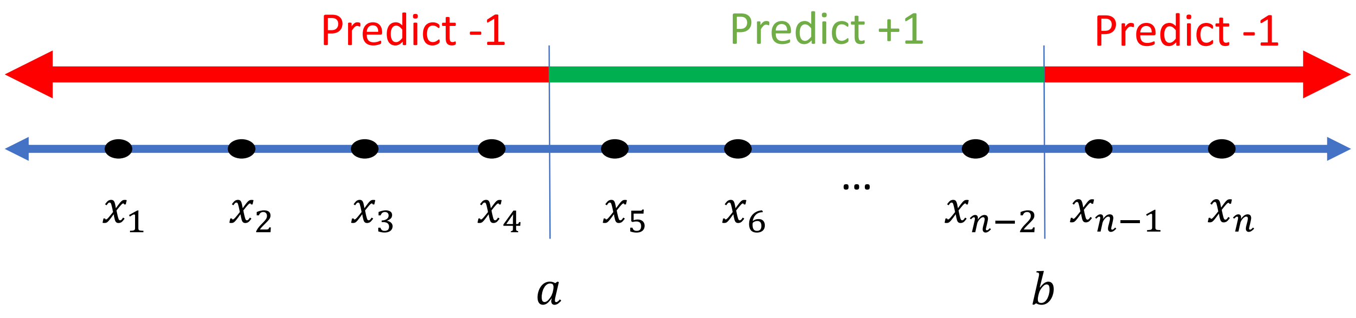
$$H(\vec{x}_1, \dots \vec{x}_N) = \{(h(\vec{x}_1), \dots, h(\vec{x}_N)) | h \in H\}$$

  - 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

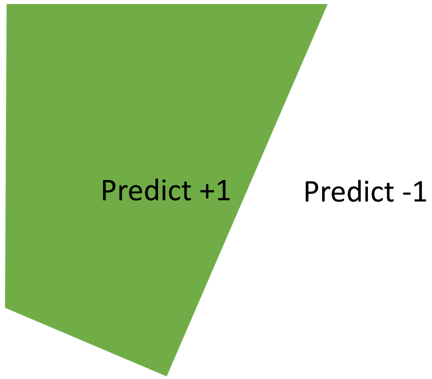
$$m_H(N) = \max_{(\vec{x}_1, \dots, \vec{x}_N)} |H(\vec{x}_1, \dots, \vec{x}_N)|$$

What is $m_H(N)$?

- 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



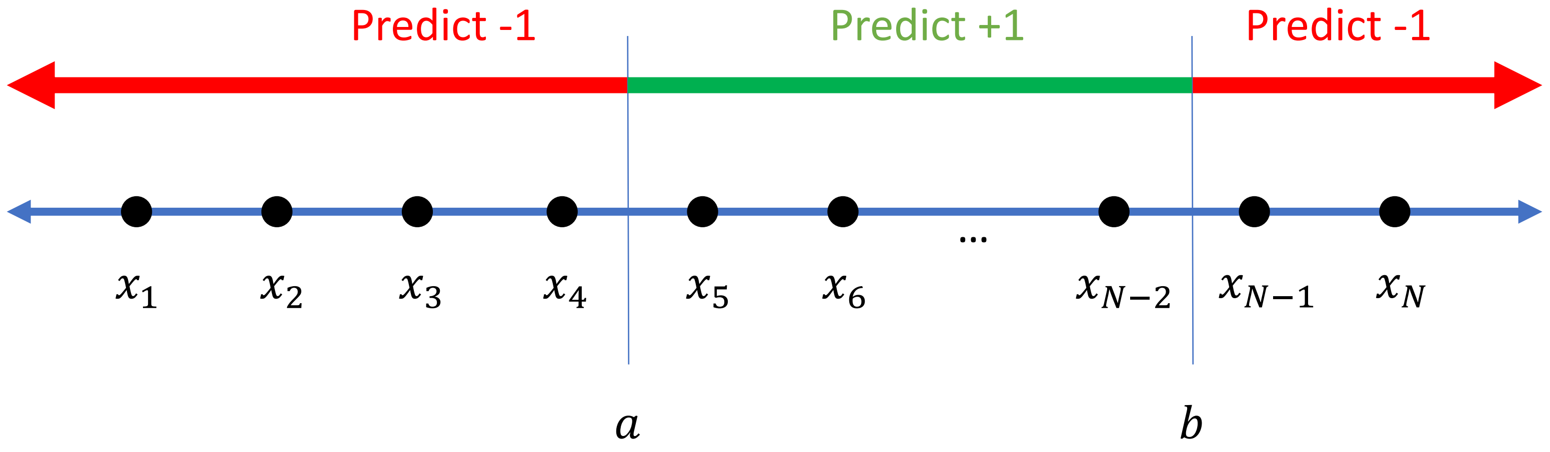
- Dichotomies
  - Informally, consider a dichotomy as a "data-dependent" hypothesis
  - Characterized by both hypothesis set H and N data points $(\vec{x}_1, \dots, \vec{x}_N)$

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

  - 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

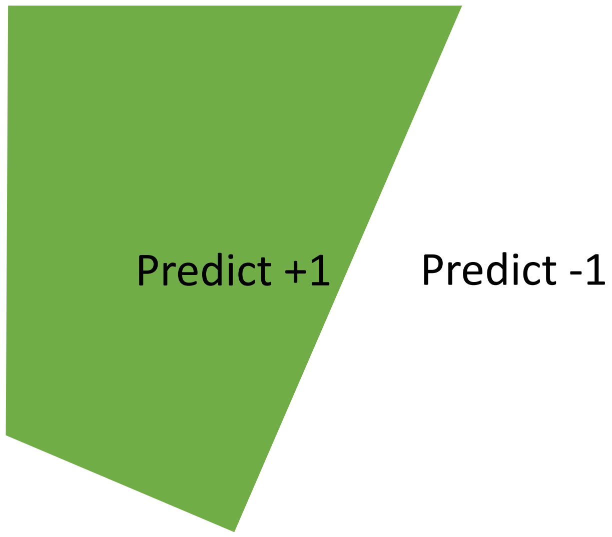
$$m_H(N) = \max_{(\vec{x}_1, \dots, \vec{x}_N)} |H(\vec{x}_1, \dots, \vec{x}_N)|$$

Example: H = Positive 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?

- 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 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 $1 - \delta$

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

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

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

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 $(\vec{x}_1, \ldots, \vec{x}_N)$

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

  - 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

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

- Exactly calculating the growth function is generally hard!
- Next lecture
  - Discuss how we can “bound” the growth function
  - Introduce the notion of VC dimension