CMSC 422 Introduction to Machine Learning

Lecture 3 Introduction

Furong Huang / furongh@umd.edu

UNIVERSITY OF MARYLAND

What is Learning?

- Ability to use previous data to perform future actions
- Biological systems do it all the time

A definition due to Simon (1983) is one of the best:

"Learning denotes changes in the system that are adaptive in the sense that they enable the system to do the task or tasks drawn from the same population more efficiently and more effectively the next time."

Simon, Herbert A. "Why should machines learn?." *Machine learning*. Springer Berlin Heidelberg, 1983. 25-37.

Topics

What does it mean to “learn by example”?

- Classification tasks
- Inductive bias
- Formalizing learning

Topics

What does it mean to “learn by example”?

- **Classification tasks**
- **Inductive bias**
- Formalizing learning

Classification tasks

- How would you write a program to distinguish a picture of me from a picture of someone else?

- Provide examples pictures of me and pictures of other people and let a **classifier** learn to distinguish the two.

Classification tasks

- How would you write a program to distinguish a sentence is grammatical or not?

- Provide examples of grammatical and ungrammatical sentences and let a **classifier** learn to distinguish the two.

Classification tasks

- How would you write a program to distinguish cancerous cells from normal cells?

- Provide examples of cancerous and normal cells and let a **classifier** learn to distinguish the two.

Let's try it out…

- Your task:

learn a classifier to distinguish

class A from class B

from examples

Examples of Class A

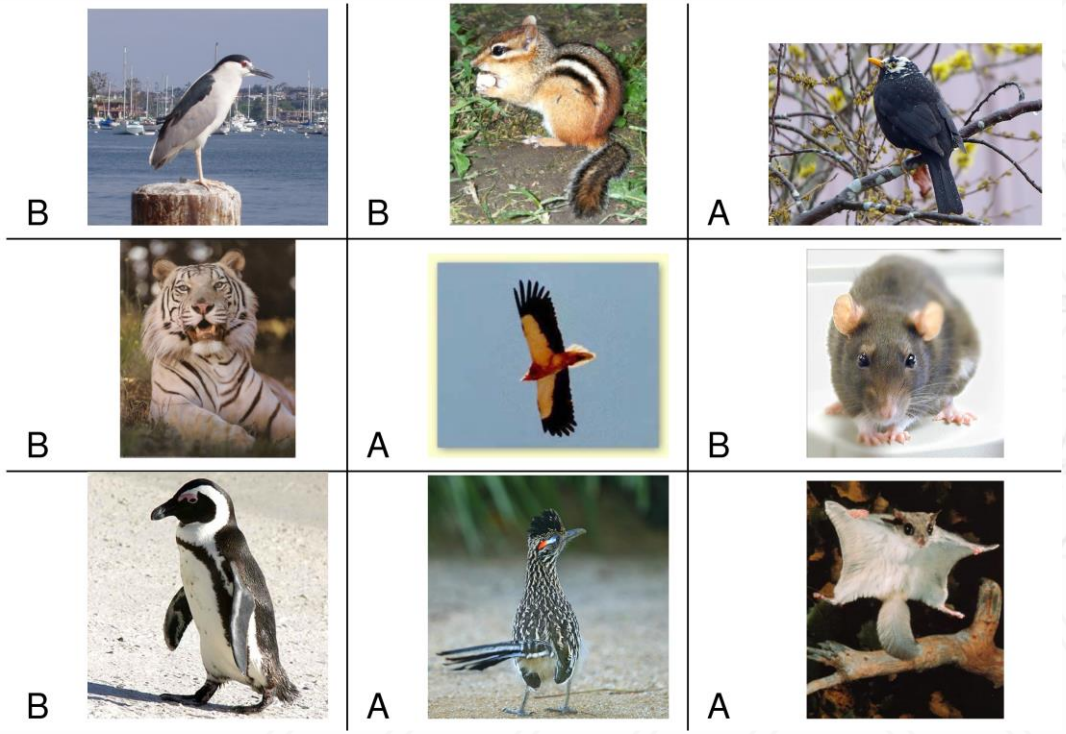
Examples of Class B

Let's try it out…

✓ Learn a classifier from examples

- Now: predict class on new examples using what you've learned

What if I told you…



Plausibly other hypotheses: is the background in focus or not?

But nobody comes up with them: why? Inductive bias --- some hypotheses are more probable than others.

Key ingredients needed for learning

Training vs. test examples

- Memorizing the training examples is not enough!
- Need to generalize to make good predictions on test examples

Inductive bias

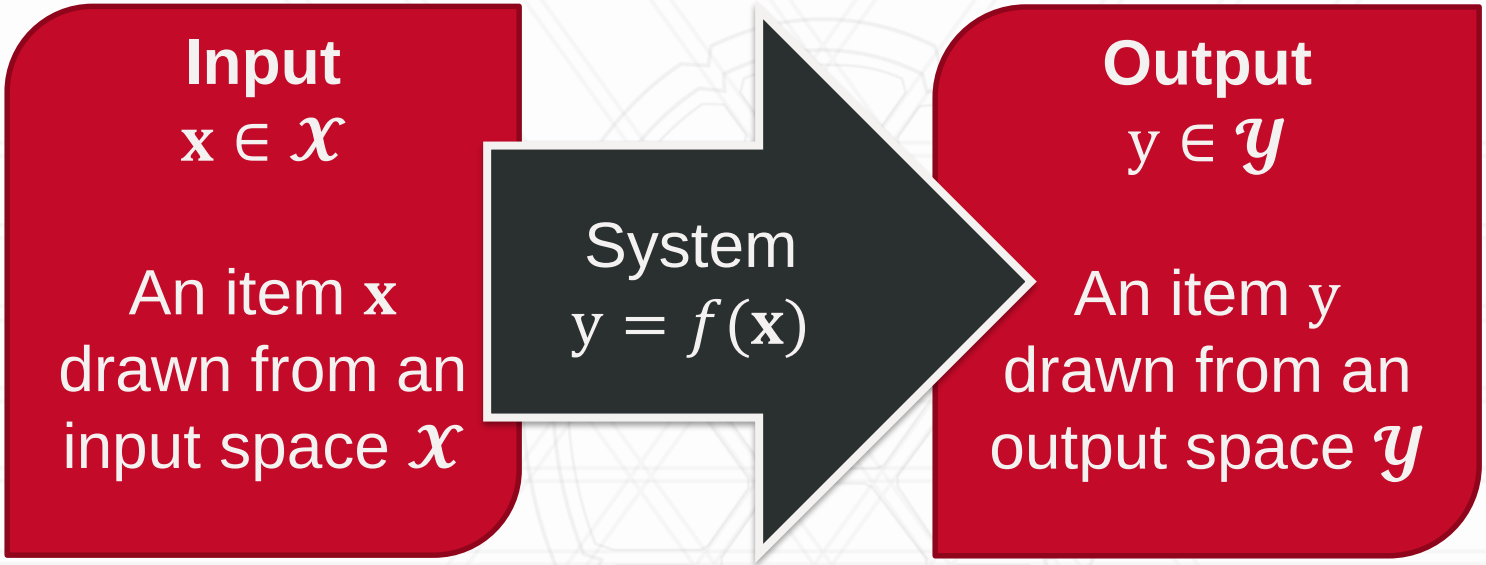
- Many classifier hypotheses are plausible
- Need assumptions about the nature of the relation between examples and classes

Topics

What does it mean to “learn by example”?

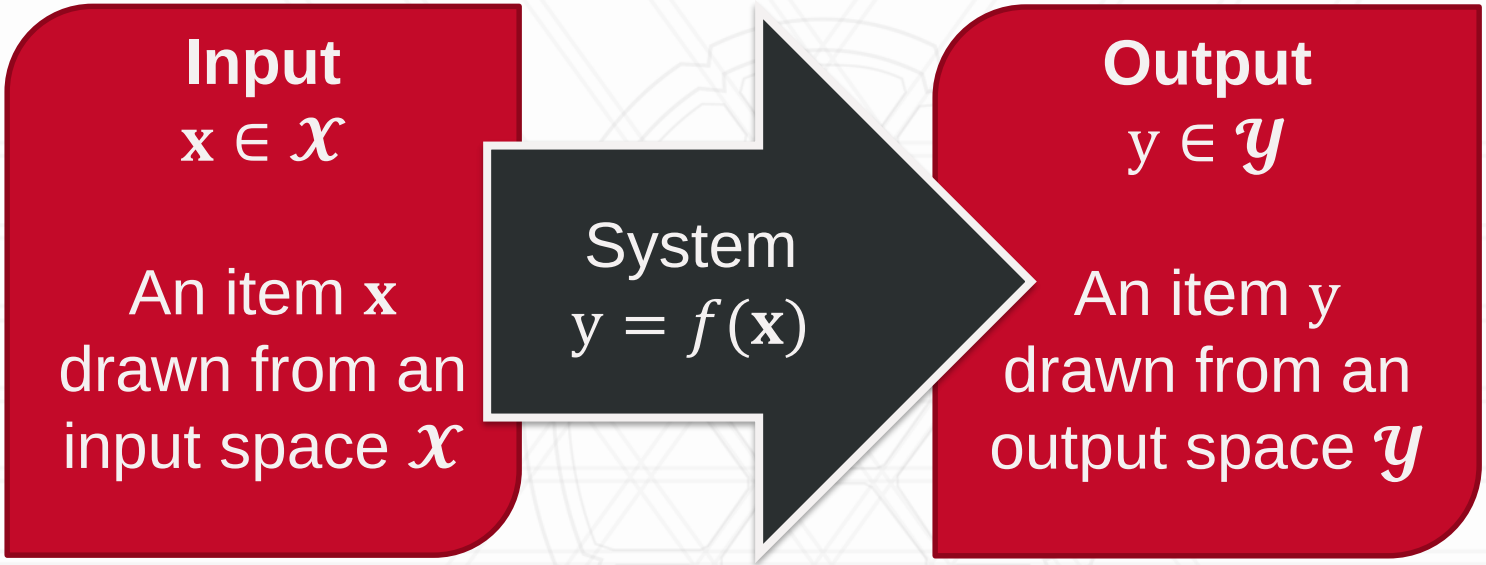
- Classification tasks
- Inductive bias
- **Formalizing learning**

Supervised Learning



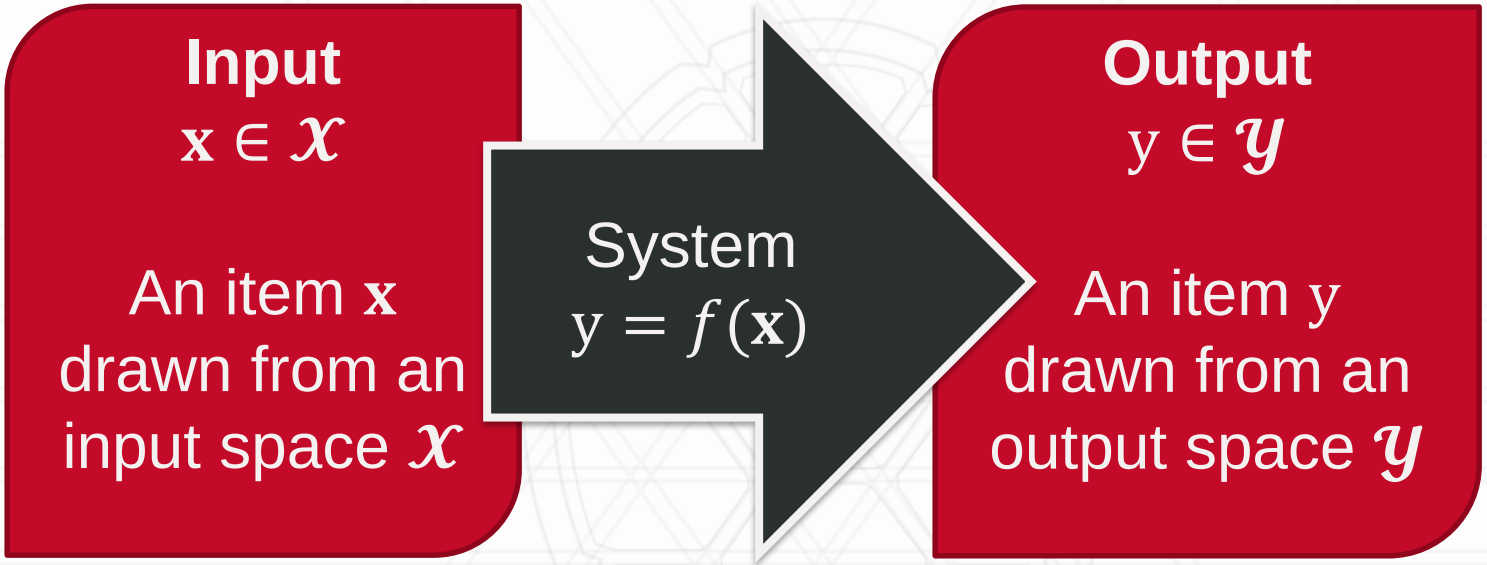
Consider systems that apply a function $f()$ to input items $\mathbf{x}$ and return outputs $\boldsymbol{y} = f(\mathbf{x})$.

Supervised Learning



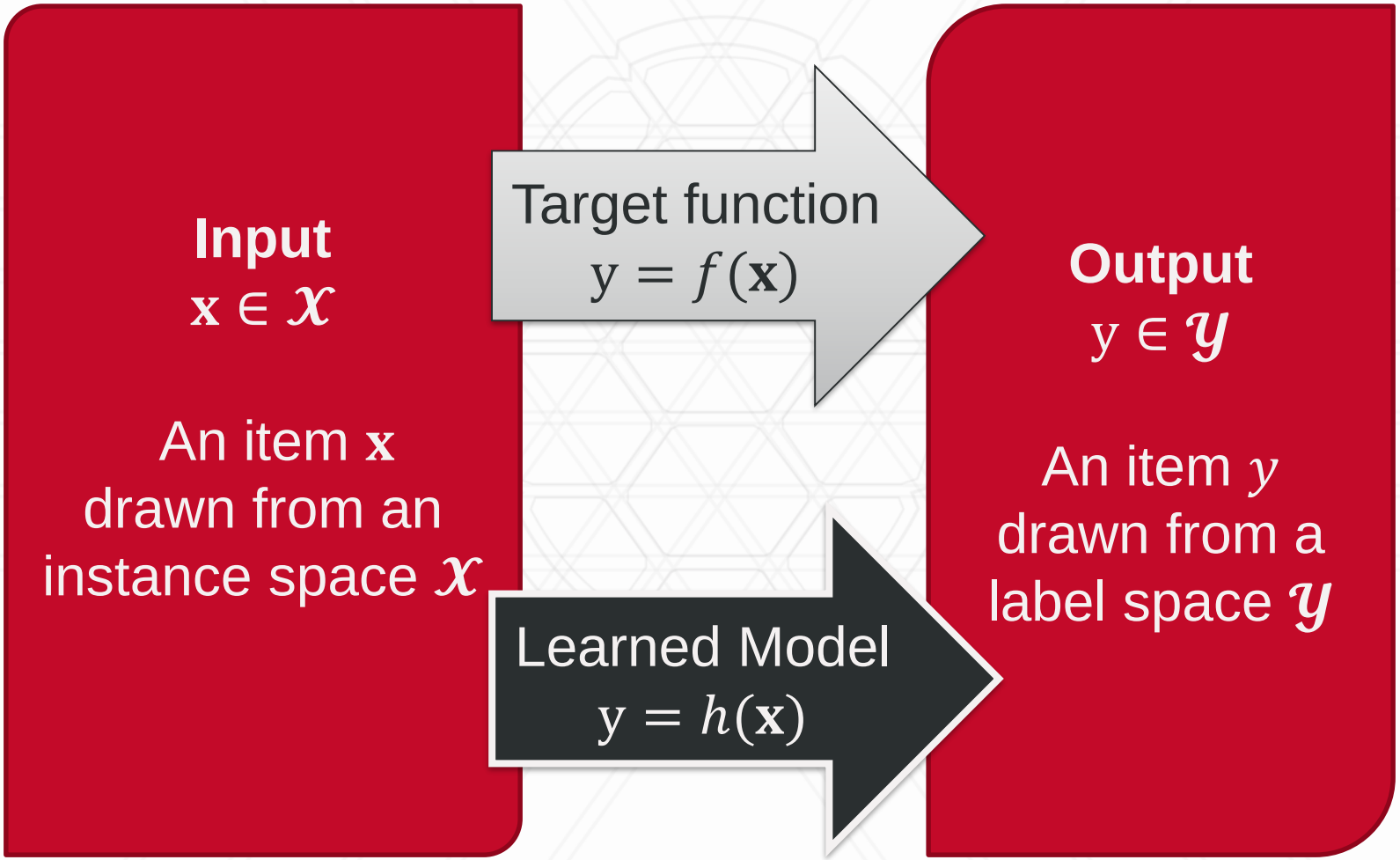
In (supervised) machine learning, we deal with systems whose $f(\mathbf{x})$ is **learned from examples.**

Supervised Learning

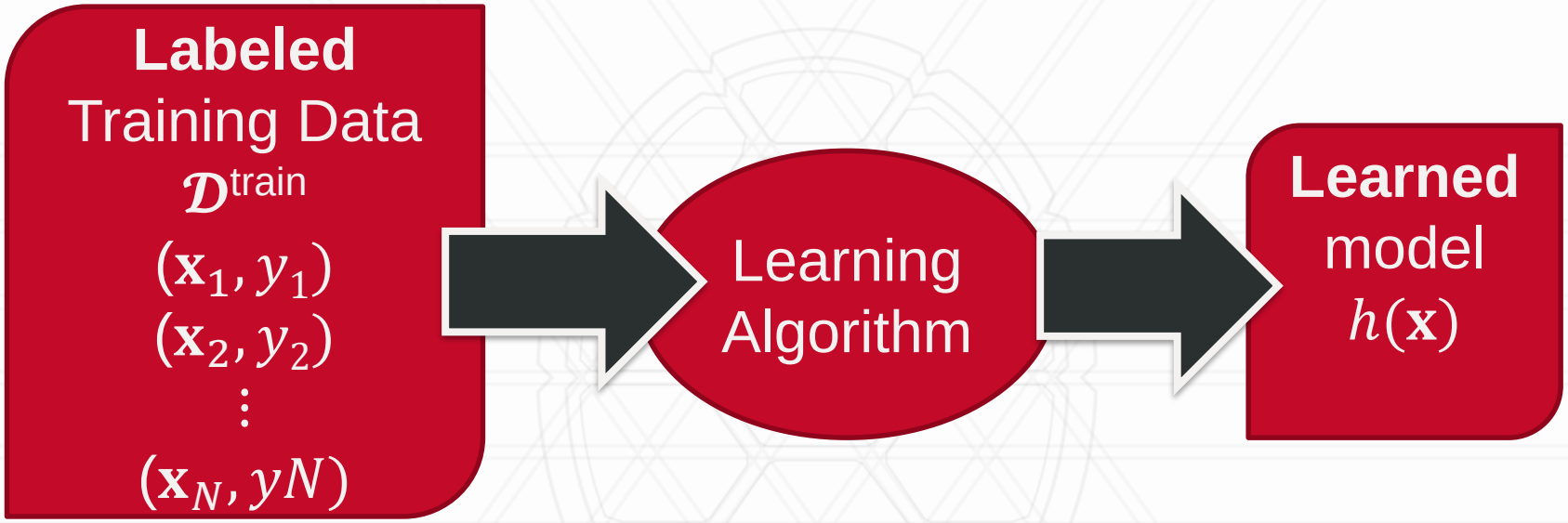


We typically use machine learning when the function $f(\mathbf{x})$ we want the system to apply is unknown to us, and we cannot "think" about it.

Supervised Learning



Supervised Learning: Training



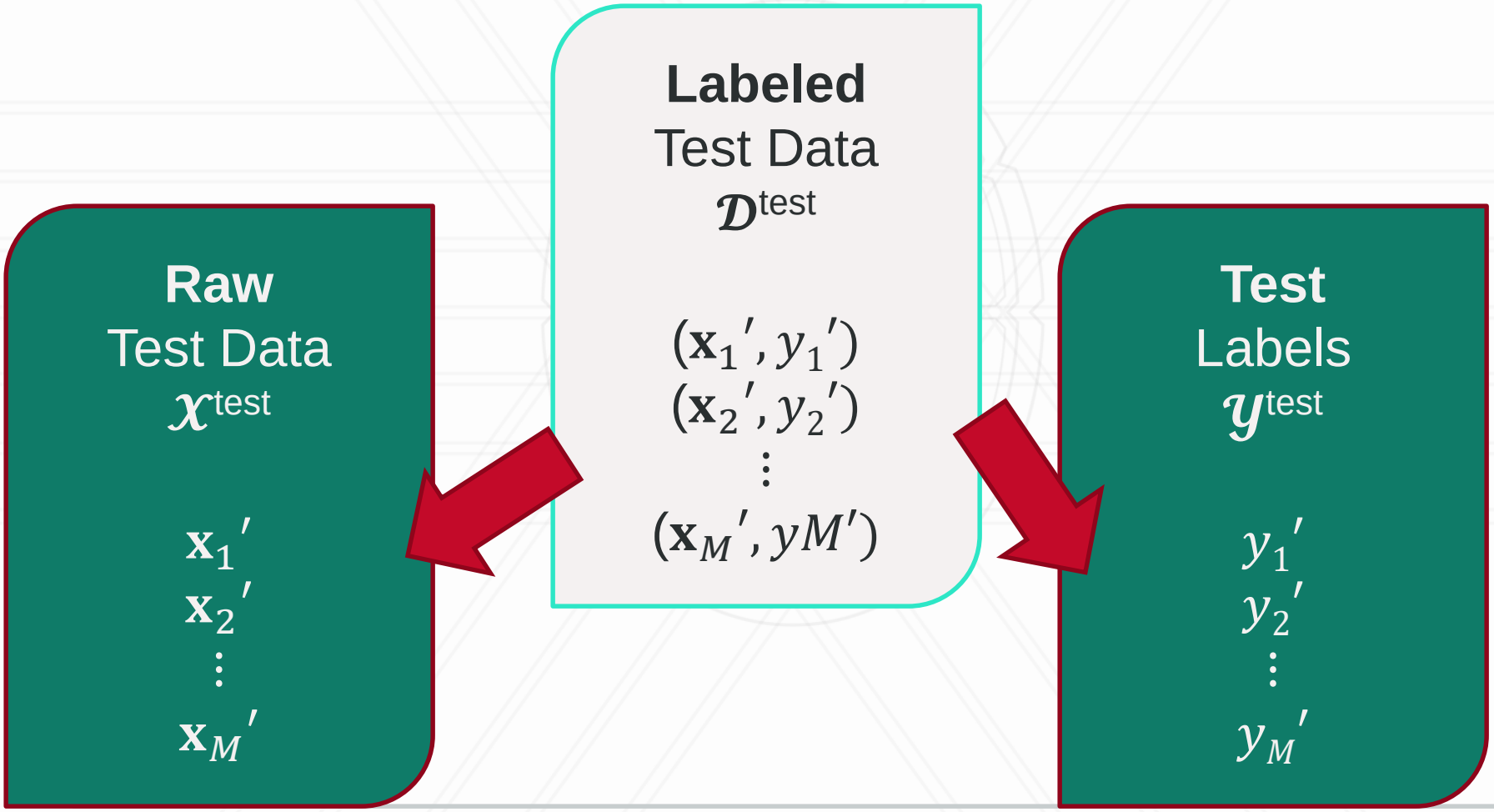
Given the training examples in $\mathcal{D}^{\text{train}}$
The learner returns a model $h(\mathbf{x})$

Supervised Learning: Testing

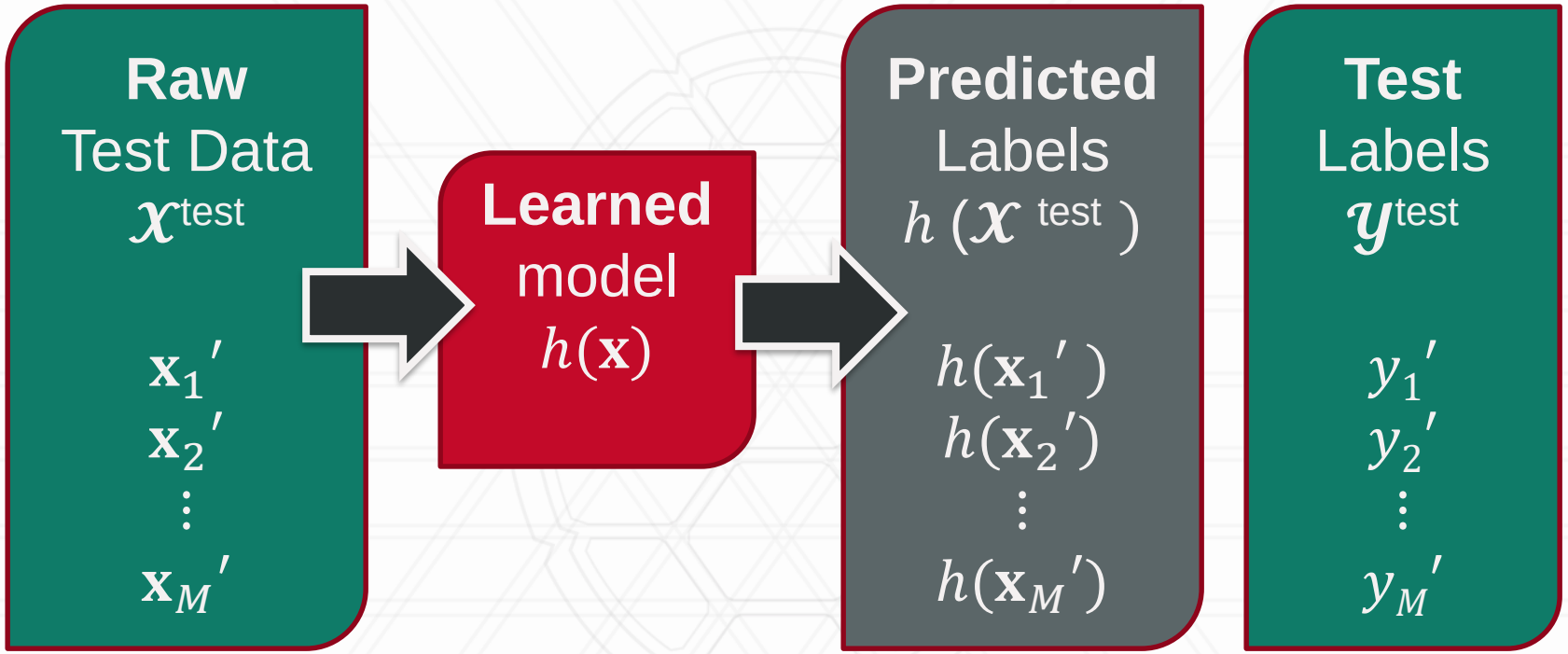
Labeled
Test Data
$\mathcal{D}^{\text{test}}$

$(\mathbf{x}_1', y_1')$
$(\mathbf{x}_2', y_2')$
$\vdots$
$(\mathbf{x}_M', yM')$

Supervised Learning: Testing



Supervised Learning: Testing



- Apply the model to the raw test data
- Evaluate by comparing predicted labels against the test labels

Can we use the test data otherwise?

Learning formally

- Given: examples $(\mathbf{x}, y = f(\mathbf{x}))$ of unknown function f
- Find: A good approximation of f
- $\mathbf{x}$ provides some representation of the input
  - Feature extraction: the process of mapping a domain element into a representation.
  - $\mathbf{x} \in \{0,1\}^n$, $\mathbf{x} \in \mathbb{R}^n$
- The target function $f()$ (label)
  - $f(\mathbf{x}) \in \{-1, +1\}$ Binary classification
  - $f(\mathbf{x}) \in \{1,2,3, \ldots, k-1\}$ Multi-class classification
  - $f(\mathbf{x}) \in \mathbb{R}$ Regression

Learning formally - continued

- Hypothesis space
  - Set of possible instances $\mathrm{X} = \{\mathbf{x} \in \boldsymbol{\mathcal{X}}\}$
  - Set of possible outputs $\mathrm{Y} = \{\mathrm{y} \in \boldsymbol{\mathcal{Y}}\}$
  - The set of function hypotheses that provide some mapping from the input to output space $H = \{h \mid h\colon \mathbf{x} \to \mathrm{y}, \mathbf{x} \in \boldsymbol{\mathcal{X}}, \mathrm{y} \in \boldsymbol{\mathcal{Y}}\}$
  - Size of hypothesis space

Machine Learning as Function Approximation

- Problem setting
  - Set of possible instances $\mathrm{X} = \{\mathbf{x} \in \boldsymbol{\mathcal{X}}\}$
  - Unknown target function $f: \mathrm{Y} = \{\mathrm{y} \in \boldsymbol{\mathcal{Y}}\}$
  - Set of function hypotheses $H = \{h \mid h: \mathbf{x} \rightarrow \mathrm{y}, \mathbf{x} \in \boldsymbol{\mathcal{X}}, \mathrm{y} \in \boldsymbol{\mathcal{Y}}\}$
- Input
  - Training examples $\{(\mathbf{x}^{(1)}, y^{(1)}), \ldots (\mathbf{x}^{(N)}, y^{(N)})\}$ of unknown target function f
- Output
  - Hypothesis $h \in H$ that best approximates target function f

Formalizing induction: Loss Function

$l(y, h(x))$ where $y\ (= f(x))$ is the truth and $h(x)$ is the system's prediction

$$\text{e.g. } l(y, h(x)\ ) = \begin{cases} 0 & if\ y = h(x) \\ 1 & otherwise \end{cases}$$

Captures our notion of what is important to learn

Formalizing induction: Data generating distribution

Where does the data come from?

- Data generating distribution

  A probability distribution D over (x, y) pairs

- We don't know what D is!

  We only get a random sample from it: our training data

Formalizing induction: Expected loss

- h should make good predictions
  - as measured by loss l
  - on **future** examples that are also drawn from D
- Formally
  - ε , the expected loss of h over D with respect to l should be small

$$\varepsilon \triangleq \mathbb{E}_{(x,y)\sim D}\{l(y,h(x))\} = \sum_{(x,y)} D(x,y)l(y,h(x))$$

Formalizing induction: Training error

- We can't compute expected loss because we don't know what D is
- We only have a sample of D
  - training examples $\{(x^{(1)}, y^{(1)}), \ldots (x^{(N)}, y^{(N)})\}$
- All we can compute is the training error

$$\hat{\varepsilon} \triangleq \sum_{n=1}^{N} \frac{1}{N} l(y^{(n)}, h(x^{(n)}))$$

Formalizing Induction

- Given
  - a loss function l
  - a sample from some unknown data distribution D

- Our task is to compute a function f that has low expected error over D with respect to l.

$$\mathbb{E}_{(x,y)\sim D}\{l(y,h(x))\} = \sum_{(x,y)} D(x,y)l(y,h(x))$$

Recap: introducing machine learning

- What does “learning by example” mean?
- Classification tasks
- Learning requires examples + inductive bias
- Generalization vs. memorization
- Formalizing the learning problem
  - Function approximation
  - Learning as minimizing expected loss

Next: Decision Trees

- **What is a decision tree?**
- How to learn a decision tree from data?
- What is the inductive bias?
- Generalization?

Entropy

Entropy $H(X)$ of a random variable X

# of possible values for X

$$H(X) = -\sum_{i=1}^{n} P(X=i) \log_2 P(X=i)$$

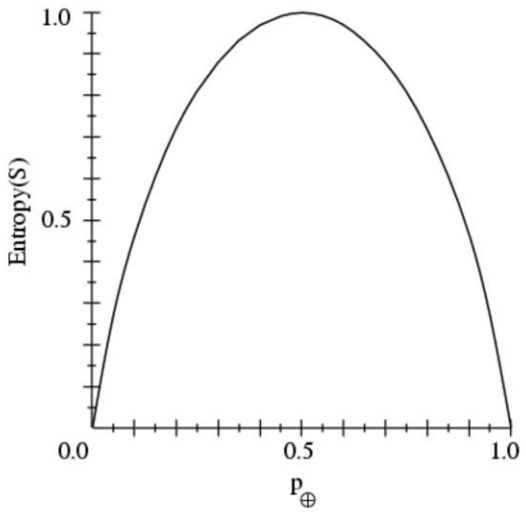
$H(X)$ is the expected number of bits needed to encode a randomly drawn value of X (under most efficient code)

Why? Information theory:

- Most efficient possible code assigns $-\log_2 P(X=i)$ bits to encode the message $X=i$
- So, expected number of bits to code one random X is:

$$\sum_{i=1}^{n} P(X=i)(-\log_2 P(X=i))$$

Sample Entropy



High Entropy –
High level of uncertainty

Low Entropy –
Low level of uncertainty

- S is a sample of training examples
- $p_{\oplus}$ is the proportion of positive examples in S
- $p_{\ominus}$ is the proportion of negative examples in S
- Entropy measures the impurity of S

$$H(S) \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Information Gain

- The information gain of an attribute a is the expected reduction in entropy caused by partitioning on this attribute

$$Gain(S, a) = Entropy(S) - \sum_{v \in values(a)} \frac{|S_v|}{|S|} Entropy(S_v)$$

  - Original set is S.
  - S_v is the subset of S for which attribute a has value v.
  - The entropy of partitioning the data is calculated by weighing the entropy of each partition by its size relative to the original set.
    - Partitions of low entropy (imbalanced splits) lead to high gain

An example training set

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$Gain(S, a) = Entropy(S) - \sum_{v \in values(a)} \frac{|S_v|}{|S|} Entropy(S_v)$$

An example training set

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$Proportion\ of\ positive\ examples = p_+ = \frac{9}{14}$$

$$Proportion\ of\ negative\ examples = p_- = \frac{5}{14}$$

Sample Entropy

$$Entropy(S) = -p_+ \log_2 p_+ - p_- \log_2 p_- = -(9/14 * \log_2(9/14) + (5/14) * \log_2(5/14)) = 0.94$$

An example training set

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$Gain(S, a = Humidity) = Entropy(S) - \sum_{v \in values(a)} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$v = \{High, Normal\}$$

An example training set

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$Gain(S, a = Humidity) = Entropy(S) - \sum_{v \in values(a)} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$E.g., v = High, \quad |S_v| = 7, \quad Entropy(S_v) = -\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7}$$

An example training set

Day	Outlook	Temperature	Humidity	Wind	PlayTennis?
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$Gain(S, a = Humidity) = Entropy(S) - \sum_{v \in values(a)} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$E.g., v = Normal, \quad |S_v| = 7, \quad Entropy(S_v) = -\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log_2\frac{1}{7}$$

Exercise

Find attribute a s.t. Gain(S, a) is maximized.

- **Check out course webpage, Canvas, Piazza**
- **Do the readings!**

UNIVERSITY OF MARYLAND

Furong Huang
4124 IRB, College Park, MD 20740
301.405.8010 / furongh@umd.edu