Introduction to Machine Learning

CMSC 422

Ramani Duraiswami

Basic Introductory Concepts in Supervised Learning

Where to…

- find the readings: A Course in Machine Learning
- view and submit assignments: Canvas
- check your grades: Canvas
- ask and answer questions, participate in discussions and surveys, contact the instructors, and everything else: Piazza
  - Please use piazza instead of email
- Read the lecture notes: Class website

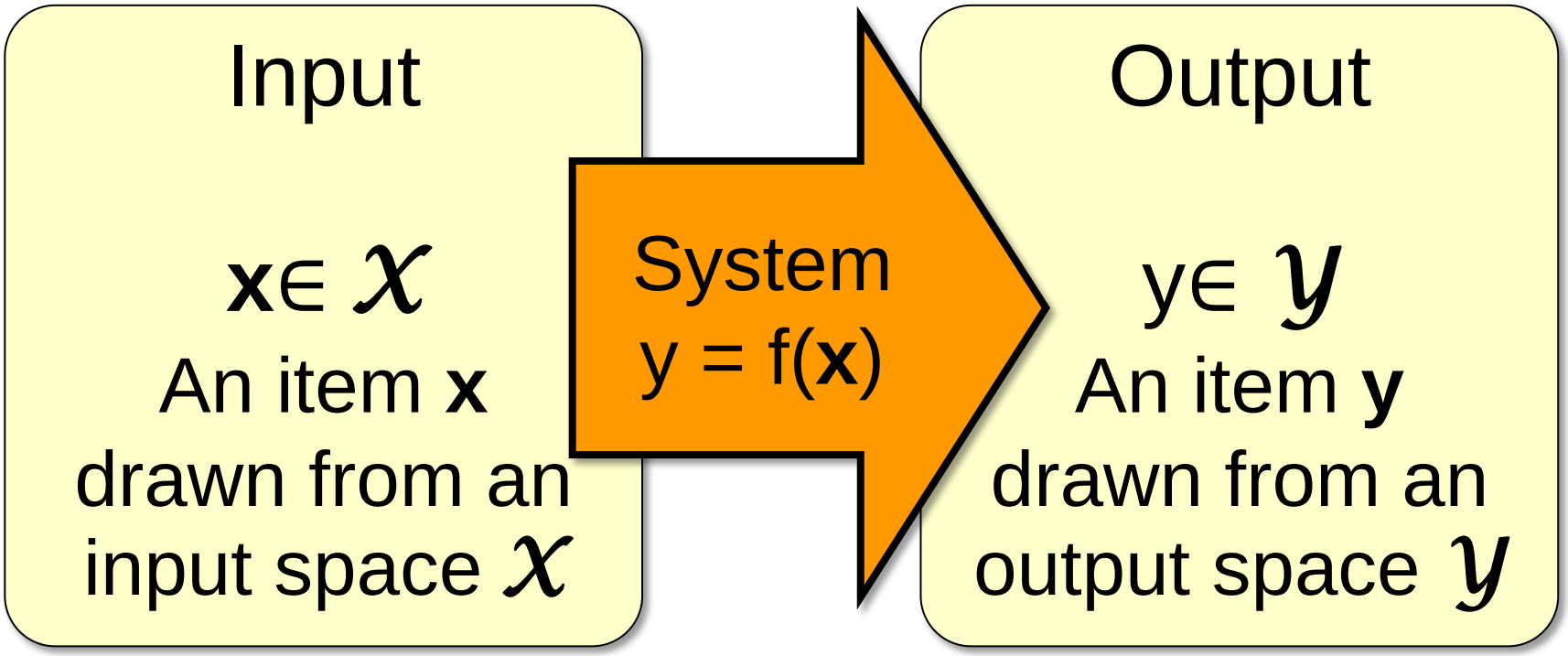
Homework -1

- Due Thursday before class
- Homework disappears
- Homework 1 – 5 attempts, no penalty
- Future homework – 2 attempts
- Where is the homework posted?

Class Preparation

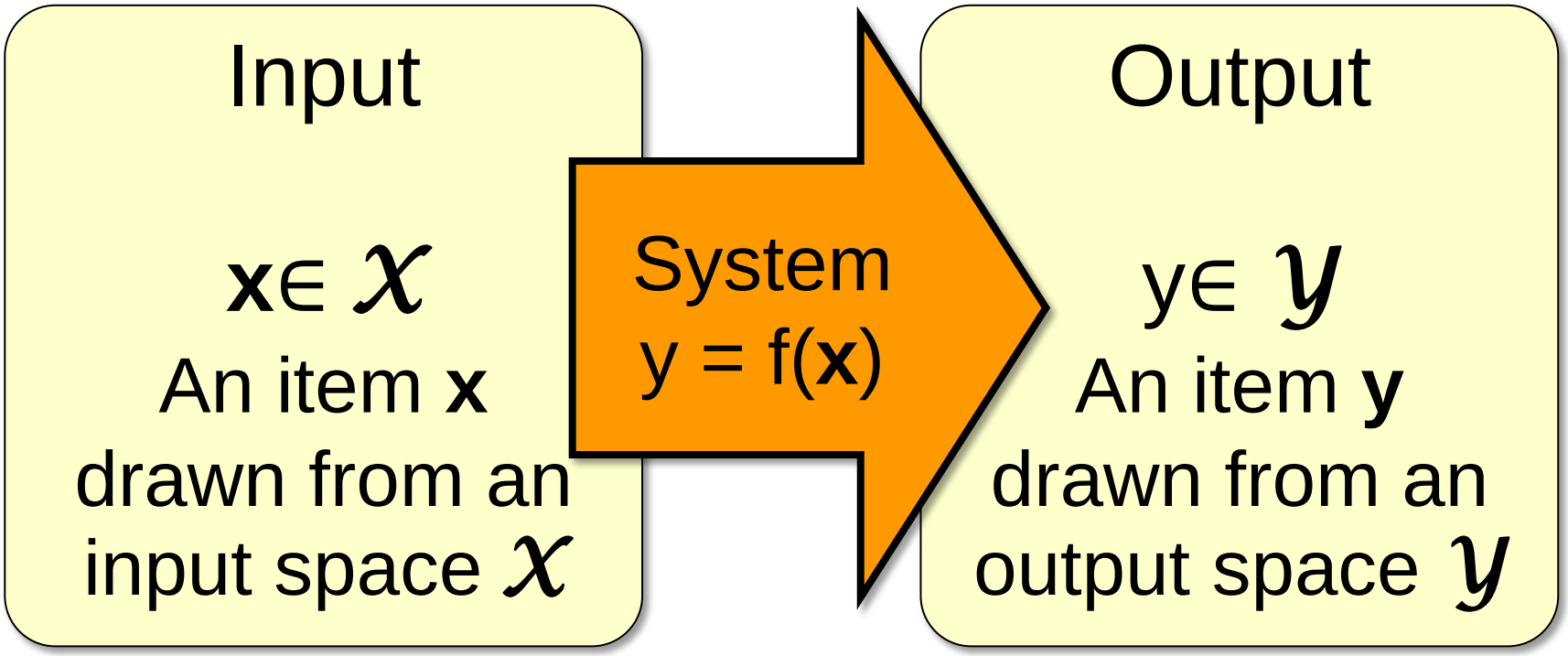
- Chapter 1 of a Course in Machine Learning
- Did you do your assigned reading?
- Textbook is free and online.
- http://ciml.info

Supervised Learning



- We consider systems that apply a function f() to input items **x** and return an output **y** = f(**x**).

Supervised Learning

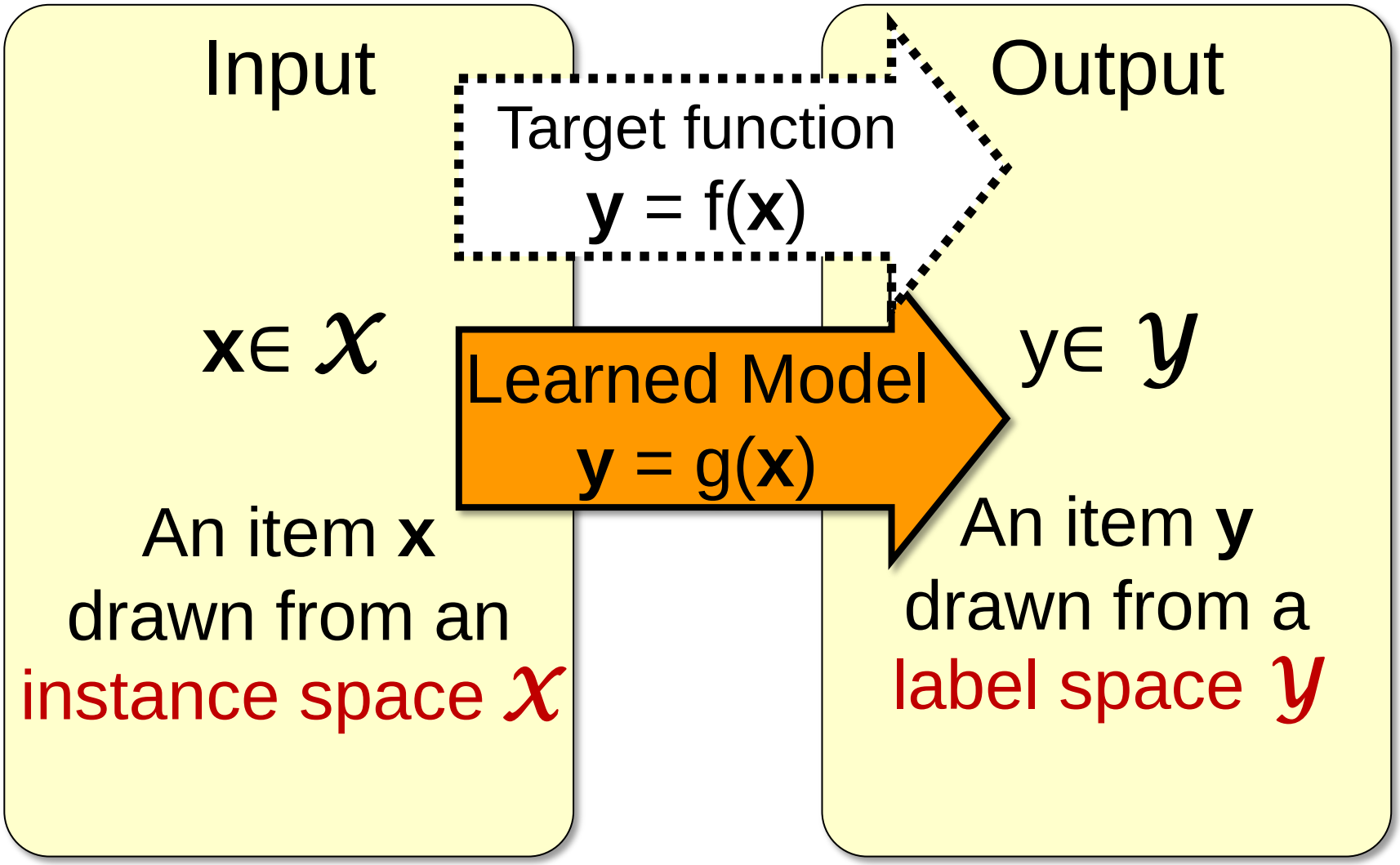


- In (supervised) machine learning, we deal with systems whose f(**x**) is learned from examples.

Why use learning?

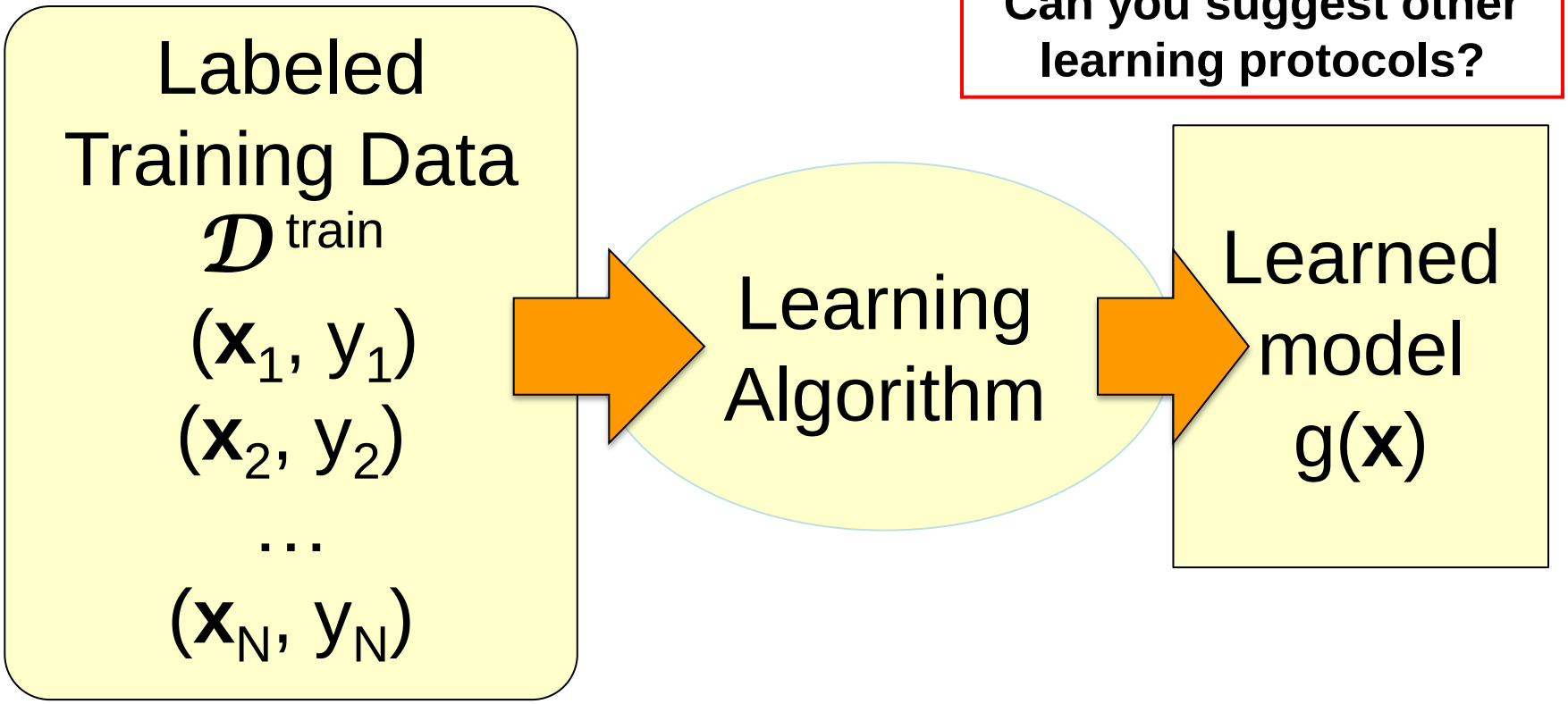
- We typically use machine learning when the function f(**x**) we want the system to apply is unknown to us, and we cannot "think" about it. The function could actually be simple.
- Different from tasks such as polynomial interpolation
  - Though those tasks can be considered in a learning setting

Supervised learning



Supervised learning: Training

Can you suggest other learning protocols?



- Give the learner examples in $\mathcal{D}^{\text{train}}$
- The learner returns a model $g(\mathbf{x})$

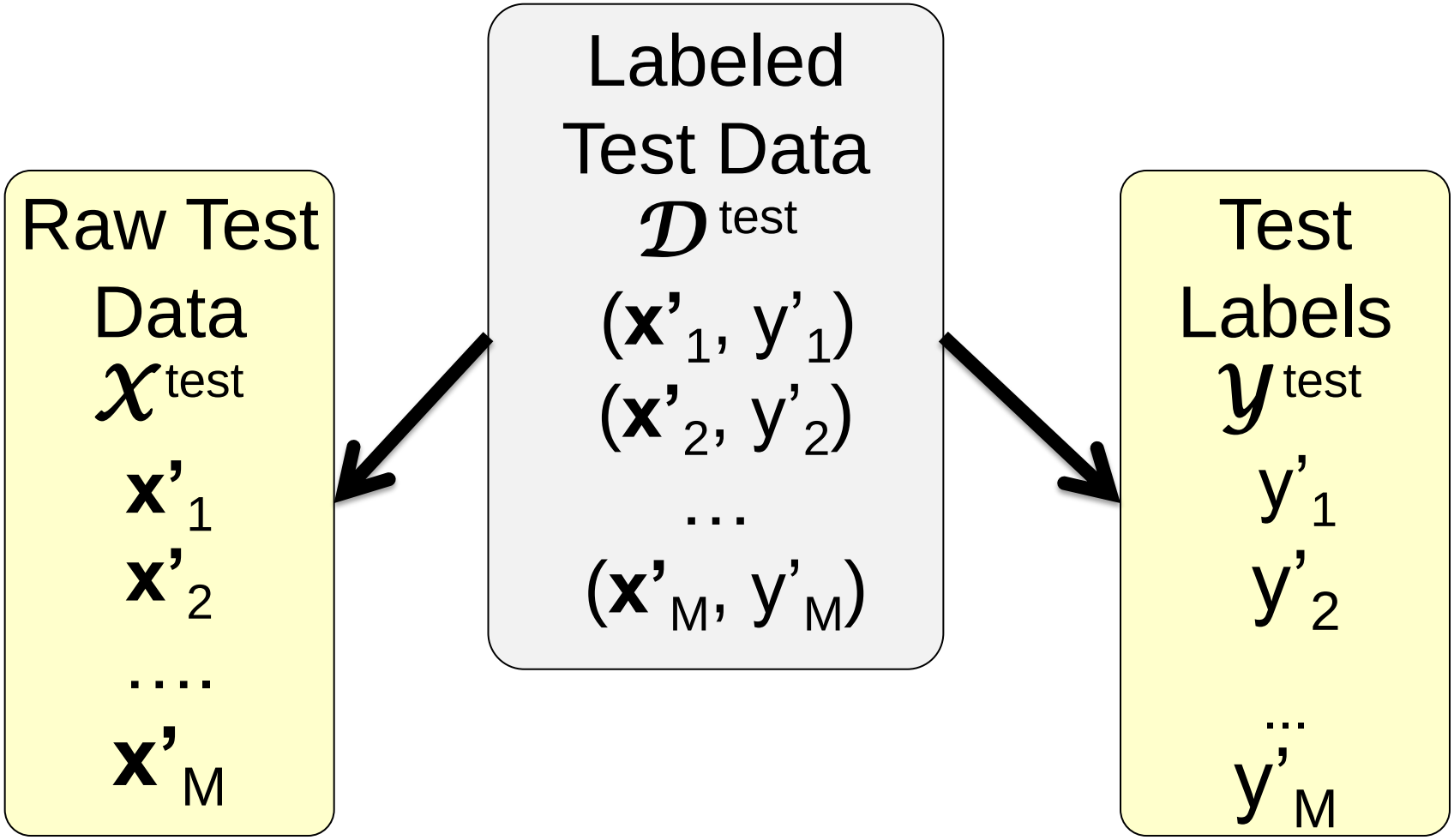
Supervised learning: Testing

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

$(\mathbf{x'}_1, y'_1)$
$(\mathbf{x'}_2, y'_2)$
…
$(\mathbf{x'}_M, y'_M)$

- Reserve some labeled data for testing

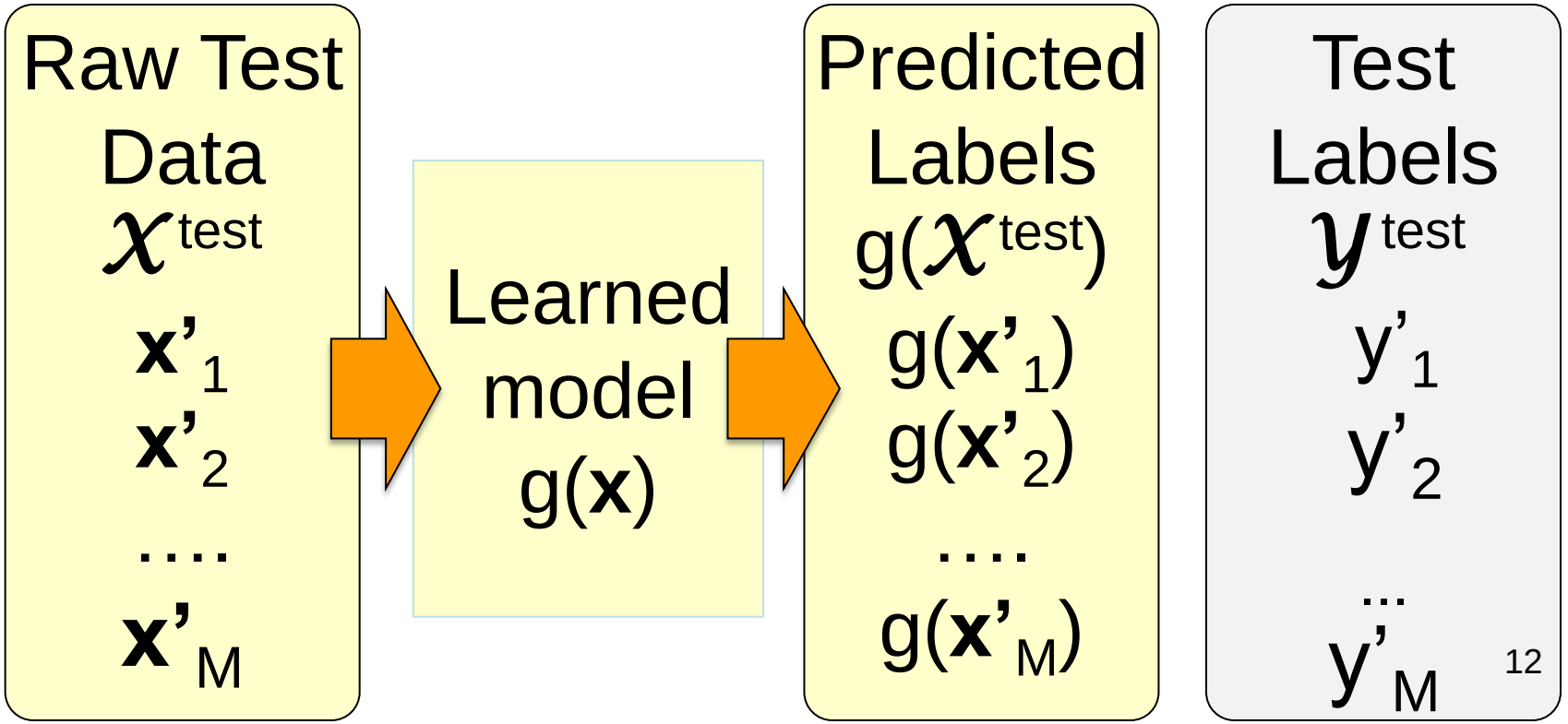
Supervised learning: Testing



Supervised learning: Testing

Can you use the test data otherwise?

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



Learning formally

- Given: Examples (x,f(x)) of some unknown function f
- Find: A good approximation of f
- x provides some representation of the input
  - The process of mapping a domain element into a representation is called Feature Extraction. (Hard; ill-understood; important)
  - $x \in \{0,1\}^n$ or $x \in \mathcal{R}^n$
- The target function (label)
  - $f(x) \in \{-1,+1\}$ Binary Classification
  - $f(x) \in \{1,2,3,..,k-1\}$ Multi-class classification
  - $f(x) \in \mathcal{R}$ Regression

Examples

- Disease diagnosis
  - x: Properties of patient (symptoms, lab tests)
  - f : Disease (or maybe: recommended therapy)
- Part-of-Speech tagging
  - x: An English sentence (e.g., The can will rust)
  - f : The part of speech of a word in the sentence
- Face recognition
  - x: Bitmap picture of person's face
  - f : Name the person (or maybe: a property of)
- Automatic Steering
  - x: Bitmap picture of road surface in front of car
  - f : Degrees to turn the steering wheel

Size of the hypothesis space

- Given all possible values of **x**$\in \mathcal{X}$
- , and all possible values of y$\in \mathcal{Y}$
- There are many functions that provide some mapping y = f(**x**)
- **Our goal is to choose the one best for the training data**

A Supervised Learning Problem

- Consider a simple, Boolean dataset:
  - $f : X \rightarrow Y$
  - $X = \{0,1\}^4$
  - $Y = \{0,1\}$

- Question 1: How should we pick the *hypothesis space*, the set of possible functions f?
- Question 2: How do we find the best f in the hypothesis space?

Dataset:

Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

Most General Hypothesis Space

- Consider all possible boolean functions over four input features!

- 2^{16} possible hypotheses
- 2^{9} are consistent with our dataset
- How do we choose the best one?

x_1	x_2	x_3	x_4	y
0	0	0	0	?
0	0	0	1	?
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	?
1	0	0	0	?
1	0	0	1	1
1	0	1	0	?
1	0	1	1	?
1	1	0	0	0
1	1	0	1	?
1	1	1	0	?
1	1	1	1	?

- Dataset:

Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

A Restricted Hypothesis Space

Consider all conjunctive boolean functions.

- 16 possible hypotheses
- None are consistent with our dataset
- How do we choose the best one?

Rule	Counterexample
$\Rightarrow y$	1
$x_1 \Rightarrow y$	3
$x_2 \Rightarrow y$	2
$x_3 \Rightarrow y$	1
$x_4 \Rightarrow y$	7
$x_1 \wedge x_2 \Rightarrow y$	3
$x_1 \wedge x_3 \Rightarrow y$	3
$x_1 \wedge x_4 \Rightarrow y$	3
$x_2 \wedge x_3 \Rightarrow y$	3
$x_2 \wedge x_4 \Rightarrow y$	3
$x_3 \wedge x_4 \Rightarrow y$	4
$x_1 \wedge x_2 \wedge x_3 \Rightarrow y$	3
$x_1 \wedge x_2 \wedge x_4 \Rightarrow y$	3
$x_1 \wedge x_3 \wedge x_4 \Rightarrow y$	3
$x_2 \wedge x_3 \wedge x_4 \Rightarrow y$	3
$x_1 \wedge x_2 \wedge x_3 \wedge x_4 \Rightarrow y$	3

Dataset:

Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

Machine Learning as Function Approximation

Problem setting

- Provided data
- Converted into a set of possible instances/features X
- Labels/outputs available for training and test data Y
- We want to "learn" the unknown target function $f: X \rightarrow Y$
- Method of learning provides a set of function hypotheses from which we can select a suitable hypothesis for the function

$$H = \{h \mid h: X \rightarrow Y\}$$

Input

- Training examples $\{(x^{(1)}, y^{(1)}), \ldots (x^{(N)}, y^{(N)})\}$ of unknown target function f

Output

- Hypothesis $h \in H$ that best approximates target function f

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

Formalizing induction: Loss Function

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

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

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

- f should make good predictions
  - as measured by loss l
  - on **future** examples that are also drawn from D

- Formally
  - ε , the expected loss of f over D with respect to l should be small

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

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 $\{\left(x^{(1)}, y^{(1)}\right), \ldots \left(x^{(N)}, y^{(N)}\right)\}$
- All we can compute is the training error

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

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.

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

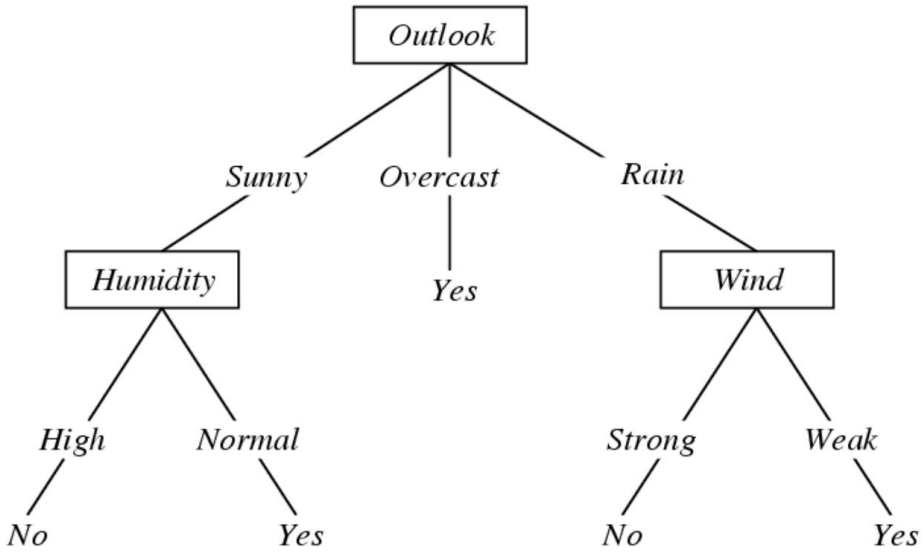
Learning Algorithm: Decision Trees

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

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

A decision tree to decide whether to play tennis



Decision Trees

- Representation
  - Each internal node tests a feature
  - Each branch corresponds to a feature value
  - Each leaf node assigns a classification
    - or a probability distribution over classifications

- Decision trees represent functions that map examples in X to classes in Y

f: <Outlook, Temperature, Humidity, Wind> → PlayTennis?

Your tasks before next class

- Check out course webpage, Canvas, Piazza
- Submit HW01
  - due Thursday 10:59am
- Do the readings!