Lecture 2: (Supervised) Machine Learning Concepts

COMP 411, Fall 2021
Victoria Manfredi

WESLEYAN
UNIVERSITY

Acknowledgements: These slides are based primarily on content from the book "Machine Learning" by Tom Mitchell, slides created by Vivek Srikumar (U of Utah), Dan Roth (UPenn), and lectures by Balaraman Ravindran (IIT Madras), as well as from courses at Stanford (http://cs229.stanford.edu/) and UC Berkeley (http://ai.berkeley.edu/)

Today's Topics

Homework 1 out

- Due Thursday, September 16 by 11:59p

Formalizing supervised learning

- Instance space and features
  - What are inputs to the learning problem?
- Label space
  - Clustering
- Hypothesis space
  - What is being learned?

vumanfredi@wesleyan.edu

Supervised Learning

INSTANCE SPACE AND FEATURES

Let’s play

Name	Label
Norman Danner	+
Karen Collins	-
Dan Licata	+
Danny Krizanc	+
Saray Shai	-
Wai Kiu Chan	-

What is the label for Simone Biles?

Can you guess the label for my name? Yours?

How were the labels generated?

Let's play

Name	Label
Norman Danner	+
Karen Collins	-
Dan Licata	+
Danny Krizanc	+
Saray Shai	-
Wai Kiu Chan	-

What is the label for Simone Biles?

Can you guess the label for my name? Yours?

How were the labels generated?

```
if first or last name states with the substring "dan"
        label = +
else
        label = -
```

Questions to think about

How can you be certain that you got the right function?

- How did you arrive at it?

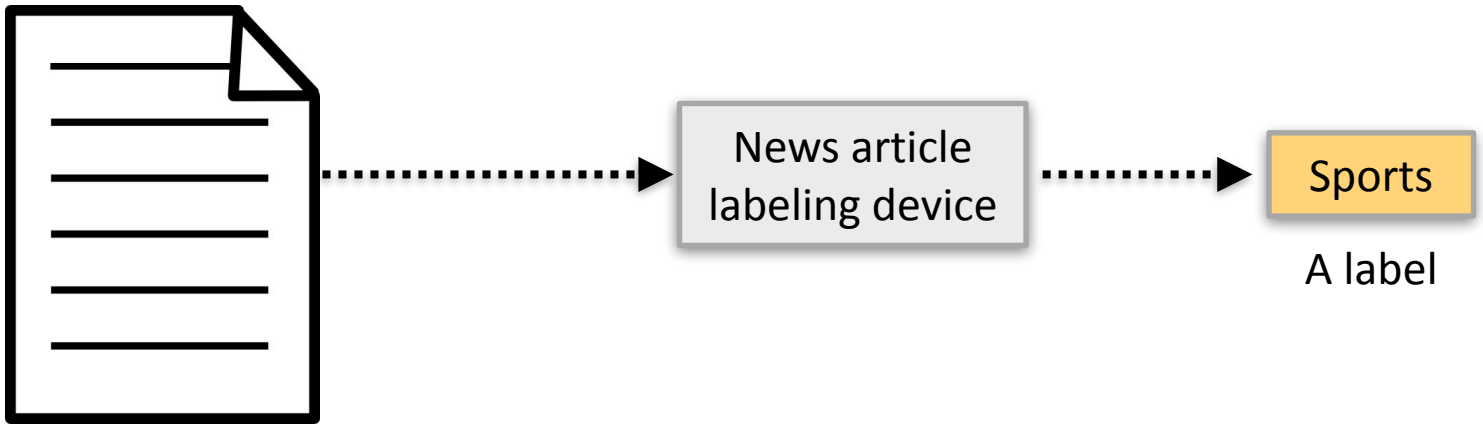
Learning issues

- Is this prediction or just modeling data? Is there a difference?
- How did you know that you should look at the letters?
- What background knowledge about letters did you use? How did you know that is relevant?
- What “learning algorithm” did you use?

vumanfredi@wesleyan.edu

Instances and labels

Running example: automatically tag news articles



An instance of a news article that needs to be classified

Instances and labels

Running example: automatically tag news articles

Mapped by the learned function to

Business

Entertainment

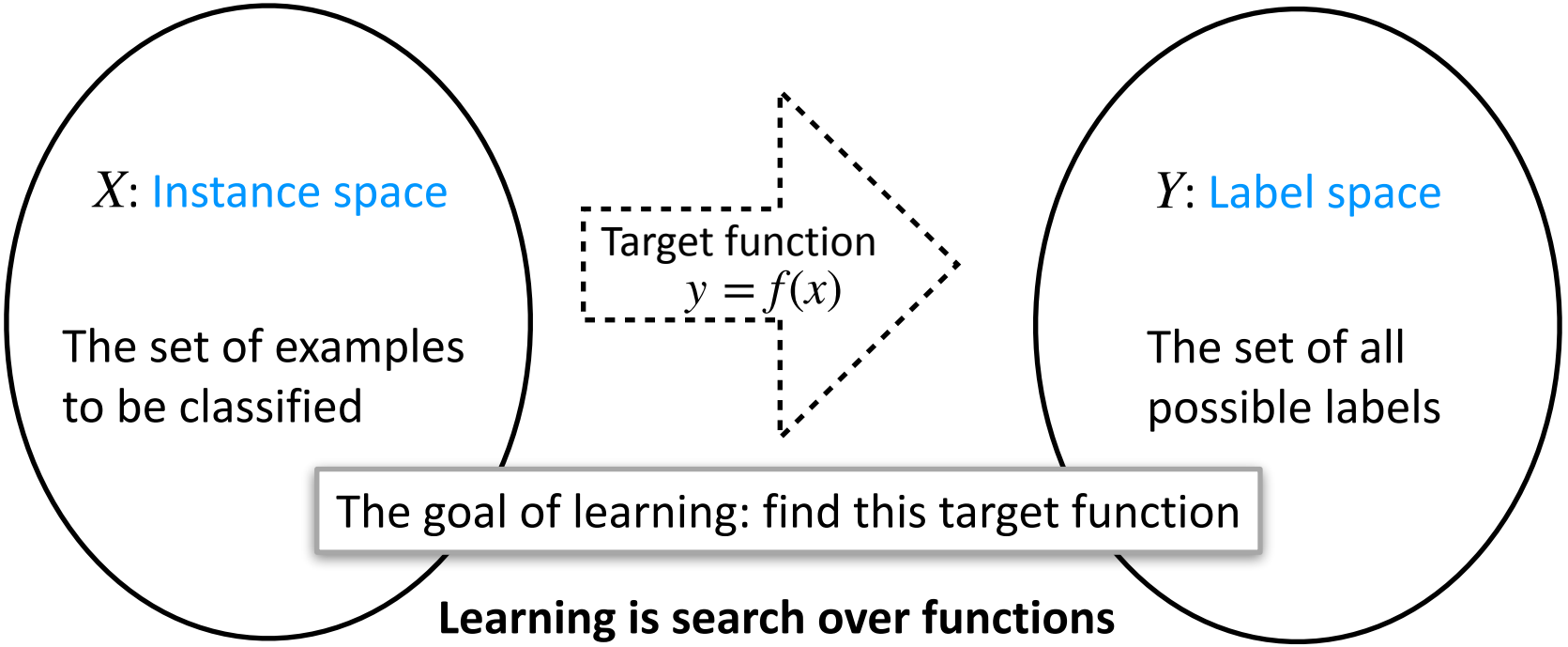
Politics

Sports

Instance space: all possible news articles

Label space: all possible labels

Instances and labels



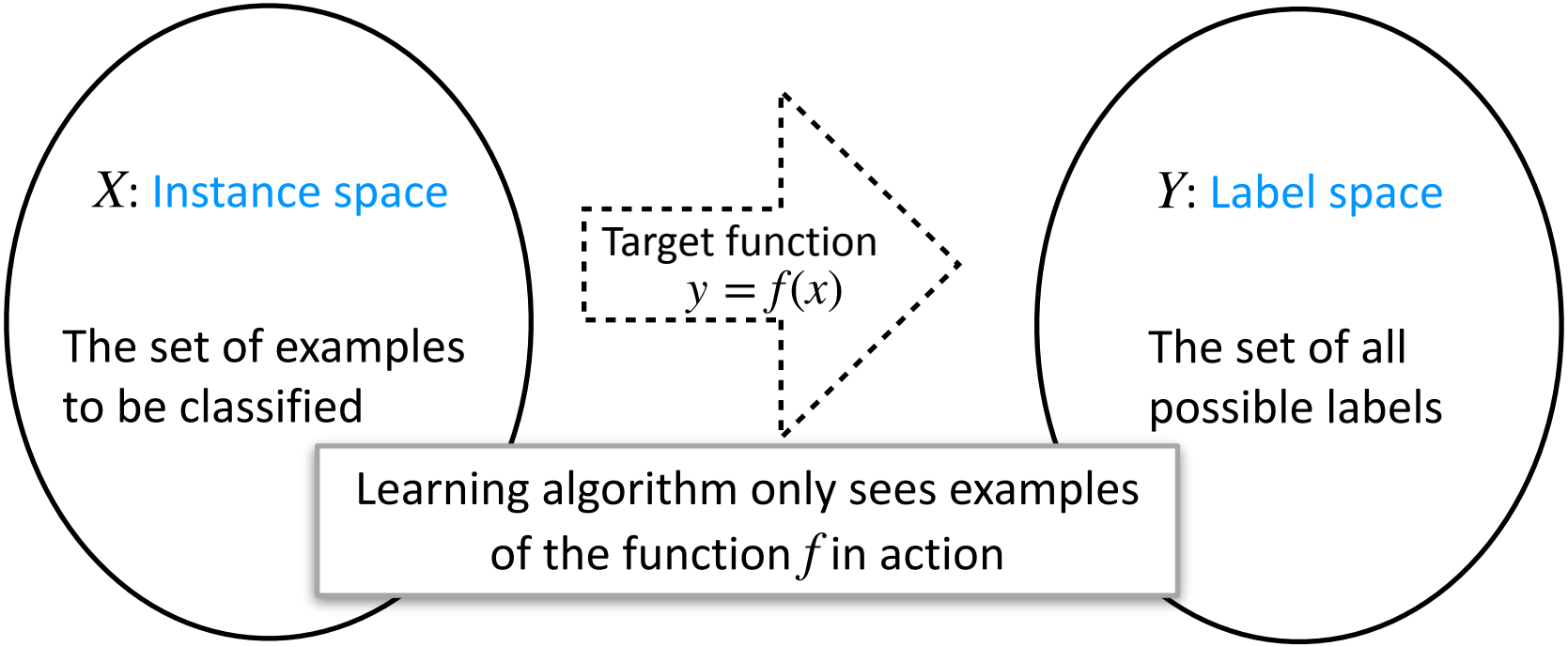
Learning is search over functions

E.g., the set of all possible names, documents, sentences, images, emails, …

E.g., {Spam, Not-Spam}, {+, -}, …

vumanfredi@wesleyan.edu

Supervised learning



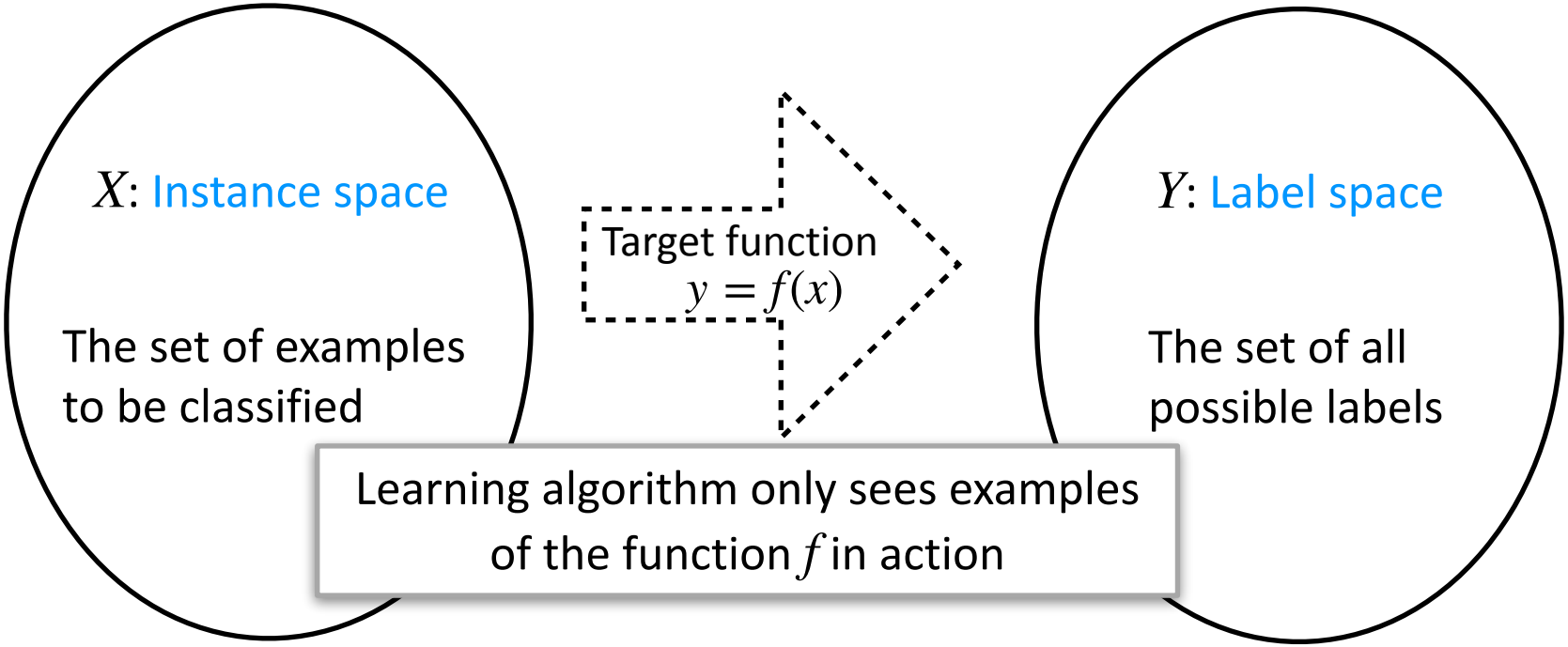
$x_1, f(x_1)$
$x_2, f(x_2)$
$x_3, f(x_3)$
$\vdots$
$x_n, f(x_n)$

Labeled training data

→ Learning algorithm →

A learned function $g : X \to Y$

Supervised learning



$x_1, f(x_1)$
$x_2, f(x_2)$
$x_3, f(x_3)$
$\vdots$
$x_n, f(x_n)$

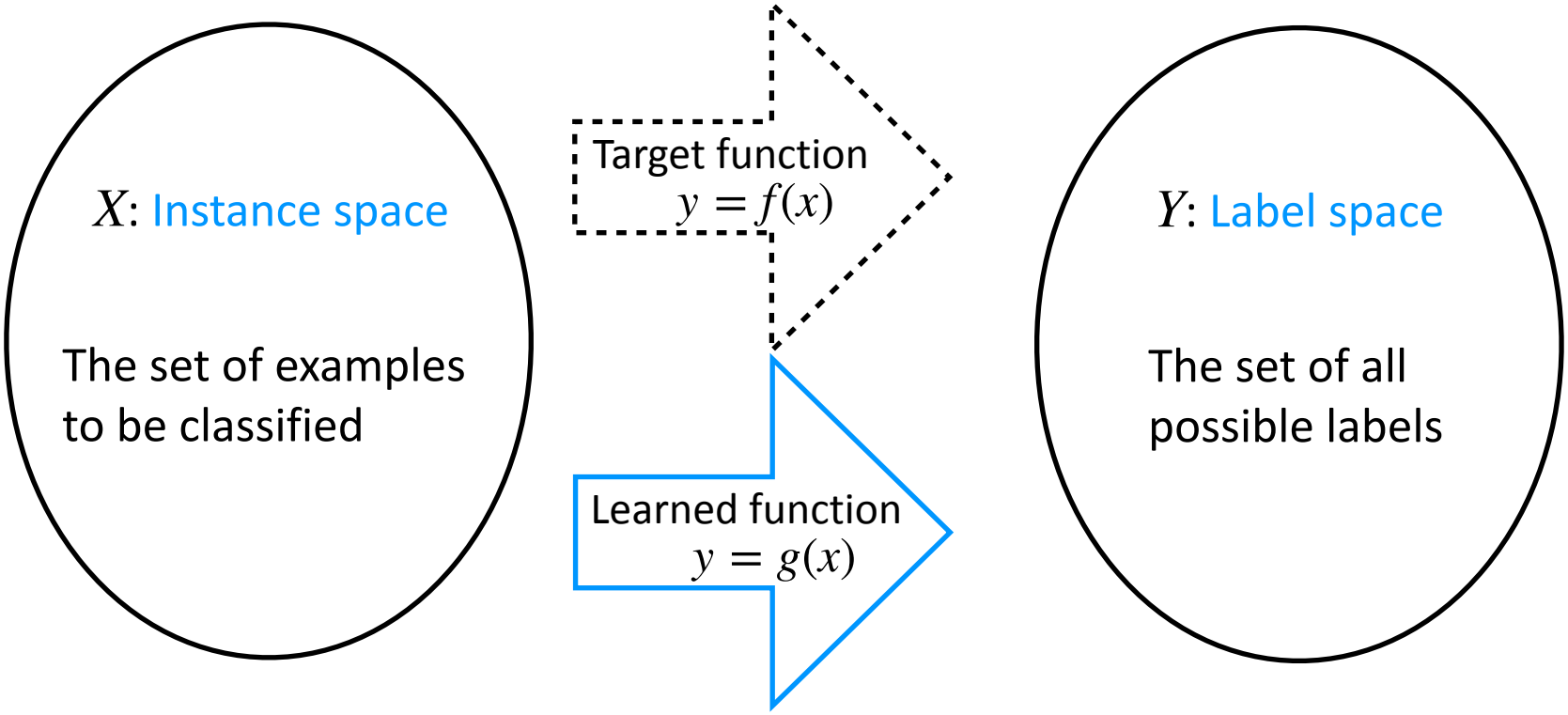
Labeled training data

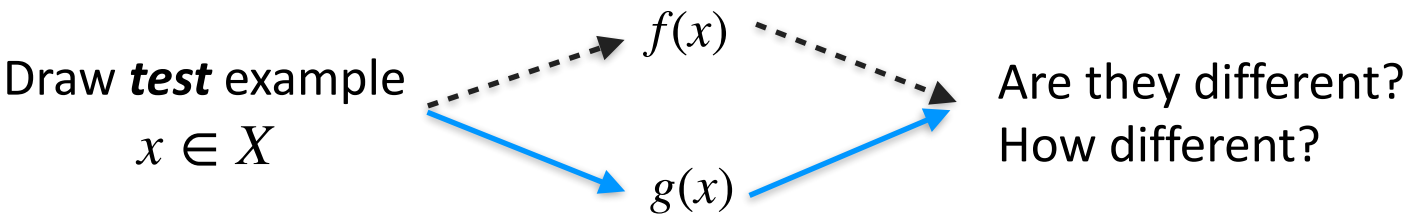
Learning algorithm

A learned function $g : X \to Y$

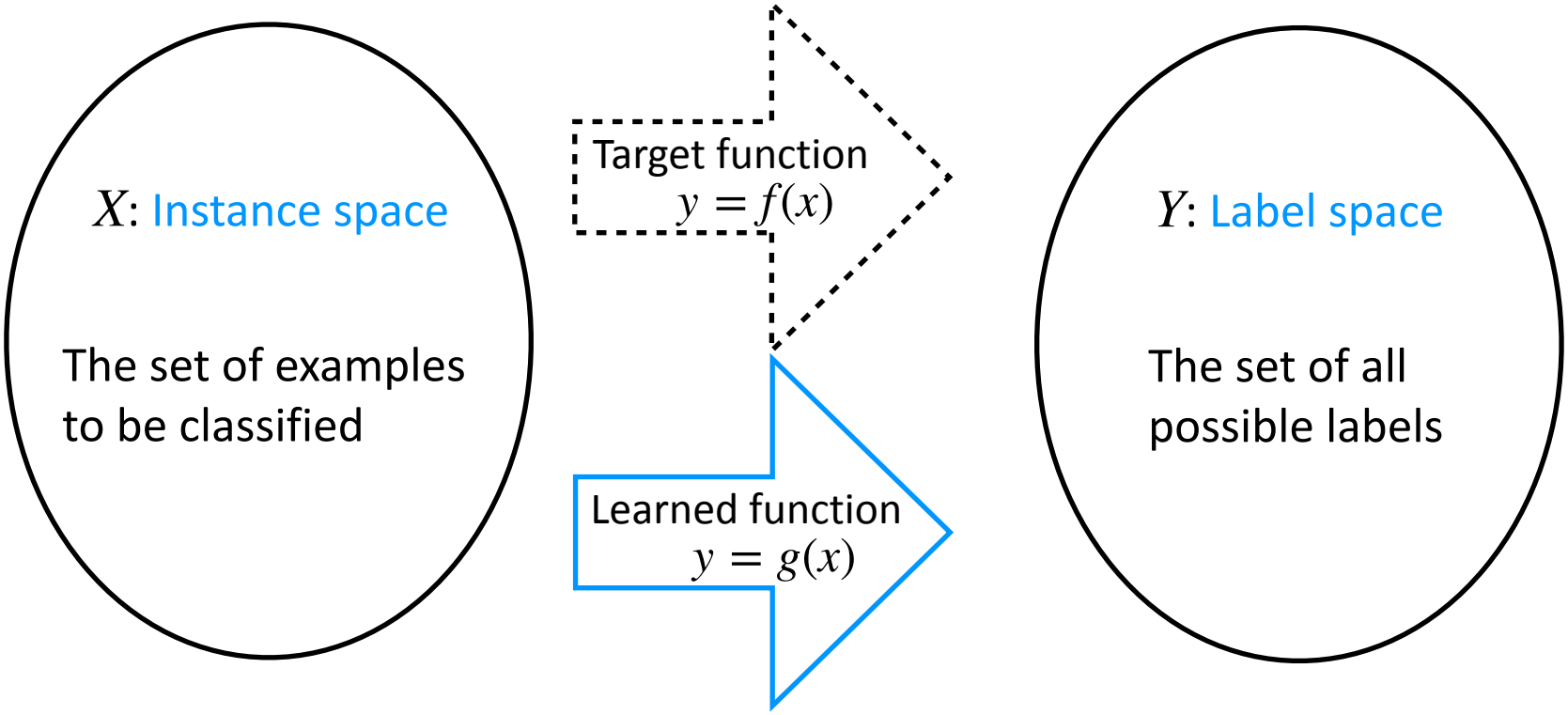
This is the training phase

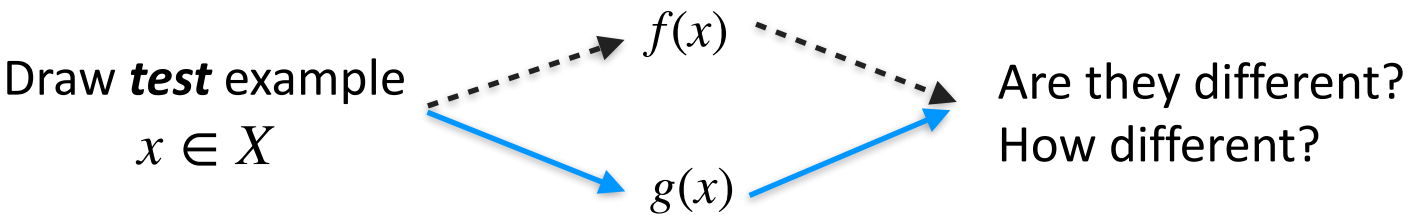
Supervised learning: evaluation





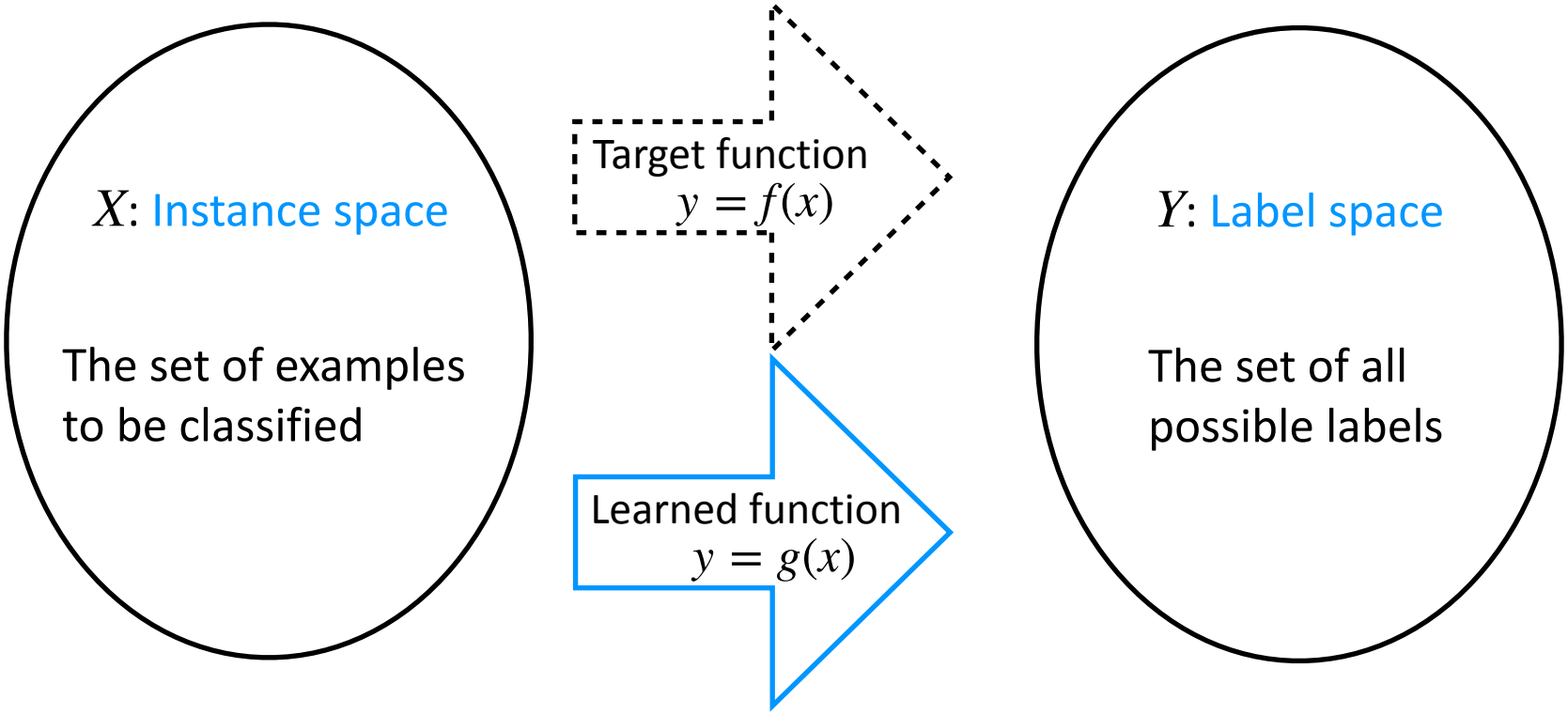
Supervised learning: evaluation

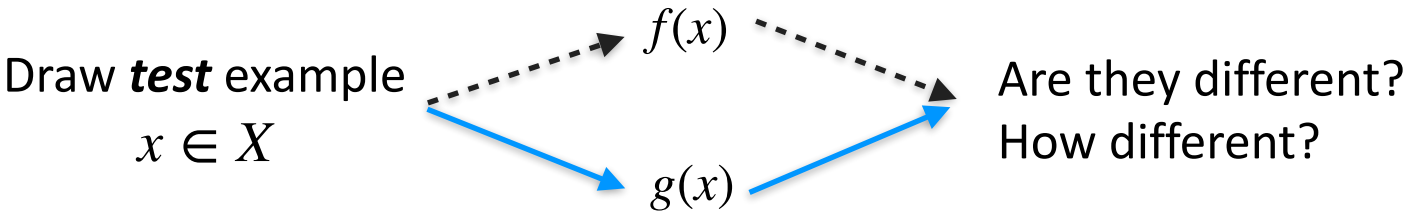




Apply model to many test examples and compare to the target's prediction
Aggregate these results to get a quality measure

Supervised learning: evaluation





Apply model to many ***test examples*** and compare to the target's prediction

Can we use these test examples during the training phase?

Supervised learning: general setting

Given: training examples that are pairs of the form $(x, f(x))$

Typically the input x is represented as feature vectors

- E.g.,: $x \in \{0,1\}^d$ or $x \in \Re^d$ (d-dimensional vectors)
- A deterministic mapping from instances in your problem (e.g., news articles) to features

The function f is unknown

For a training example $(x, f(x))$, the value of $f(x)$ is called its label

Supervised learning: general setting

Given: training examples that are pairs of the form $(x, f(x))$

The goal of learning: use the training examples to find a good approximation for f

The label determines the kind of problem we have

- **Binary classification**: label space = $\{-1, 1\}$
- **Multiclass classification**: label space = $\{1, 2, 3, \ldots, K\}$
- **Regression**: label space $= \mathfrak{R}$

Examples of binary classification

The label space consists of two elements

Spam filtering

- Is an email spam or not?

Recommendation systems

- Given user's movie preferences, will she like a new movie?

Anomaly detection

- Is a smartphone app malicious?
- Is a Twitter user a bot?

Authorship identification

- Were these two documents written by the same person?

Time series prediction

- Will the future value of a stock increase or decrease with respect to its value?

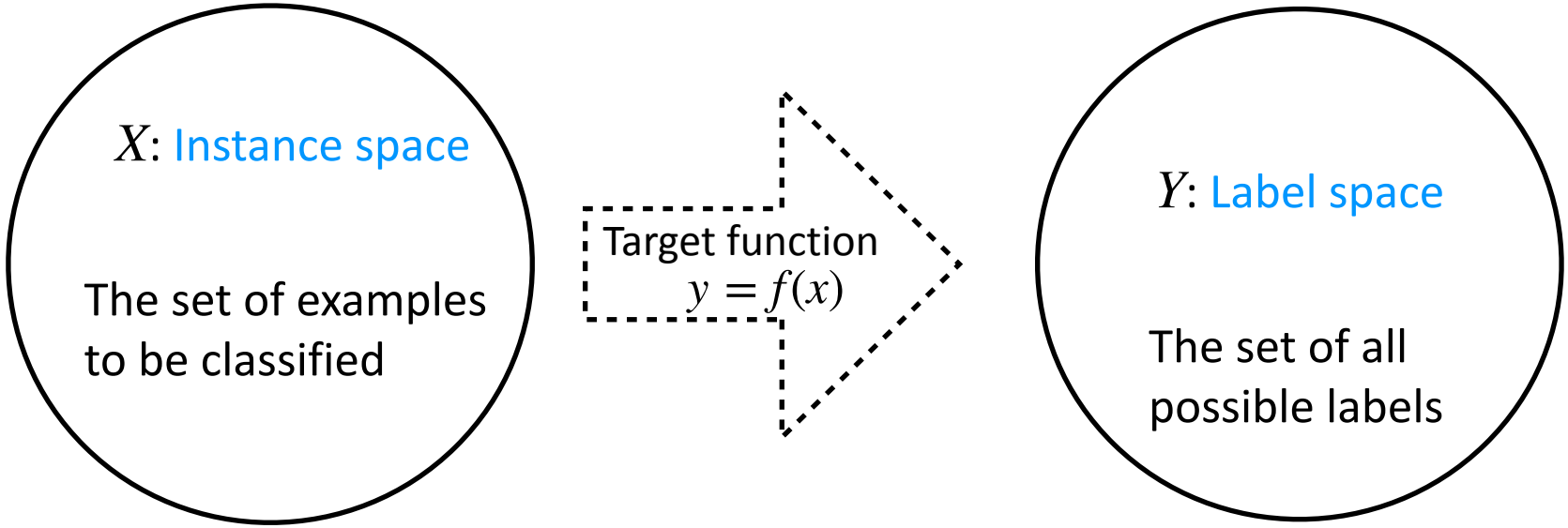
Using supervised learning

1. What is our instance space?
   - What are the inputs to the problem? What are the features?
2. What is our label space?
   - What kind of learning task are we dealing with?
3. What is our hypothesis space?
   - What functions should the learning algorithm search over?
4. What is our learning algorithm?
   - How do we learn the model from the labeled data?
5. What is our loss function or evaluation metric?
   - How do we measure success? What drives learning?

Supervised Learning

WHAT IS OUR INSTANCE SPACE?

The instance space X



Designing an appropriate feature representation of the instance space is crucial

Instances $x \in X$ are defined by features/attributes

Features could be Boolean: e.g., does the email contain the word “free”?

Features could be real-valued: e.g., what is the height of the person?

Features could be hand-crafted or themselves learned

Instances as feature vectors

Feature functions, also known as feature extractors

- Often deterministic, but could also be learned
- Convert the examples to a collection of attributes (typically thought of as high-dimensional vectors)

Important part of the design of a learning based solution

The instance space X

Features are supposed to capture all of the information needed for a learned system to make its prediction

- Think of them as the sensory inputs for the learned system

Not all information about the instances is necessary or relevant

- Bad features could even confuse a learner

What might be good features for the name label game?

Instances as feature vectors

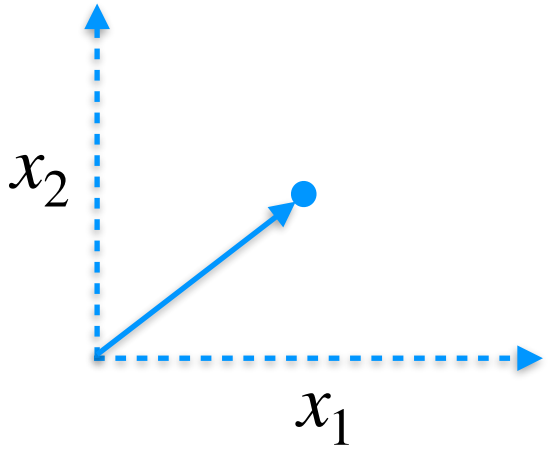
Feature functions convert inputs to vectors

The input space X is a d-dimensional vector space (e.g., $\Re^d$ or $\{0,1\}^d$)

- Each dimension is one feature, we have d features in all

Each $\mathbf{x} \in X$ is a feature vector

- Each $\mathbf{x} = [x_1, x_2, \ldots, x_d]$ is a point in the vector space with d dimensions (hence the boldface $\mathbf{x}$)



Feature functions produce feature vectors

When designing feature functions, think of them as templates

- Feature: "The second letter of the name"
  - Norman a → [0 0 0 0 … 1 …]
  - Karen a → [1 0 0 0 … 0 …]
  - Dan a → [1 0 0 0 … 0 …]
  - Danny a → [1 0 0 0 … 0 …]
  - Saray a → [1 0 0 0 … 0 …]
  - Wai a → [1 0 0 0 … 0 …]

What is the dimensionality of these feature vectors?

26 (one dimension per letter)

Such vectors where exactly one dimension is 1 and all others are 0 are called one-hot vectors

This is the one-hot representation of the feature "The second letter of the name"

Feature functions produce feature vectors

When designing feature functions, think of them as templates

- Feature: "The second letter of the name"
  - Norman o → [0 0 0 0 … 1 …]
  - Karen a → [1 0 0 0 … 0 …]
  - Dan a → [1 0 0 0 … 0 …]
  - Danny a → [1 0 0 0 … 0 …]
  - Saray a → [1 0 0 0 … 0 …]
  - Wai a → [1 0 0 0 … 0 …]
- Feature: "The length of the first name"
  - Norman → 6
  - Karen → 5
- "The second letter of the name, The length of the first name, The length of the last name"
  - Norman Danner → [0 0 0 0 … 1 … 6 6]
  - Karen Collins → [1 0 0 0 … 0 … 5 7]

Features can be accumulated by concatenating the vectors

Good features are essential

Features determine how well a task can be learned

- E.g., a bad feature for our game: "Is there a day of the week that begins with the last letter of the first name?
- *Why would we think that this is a bad feature?*

Much effort goes into designing (or learning) features

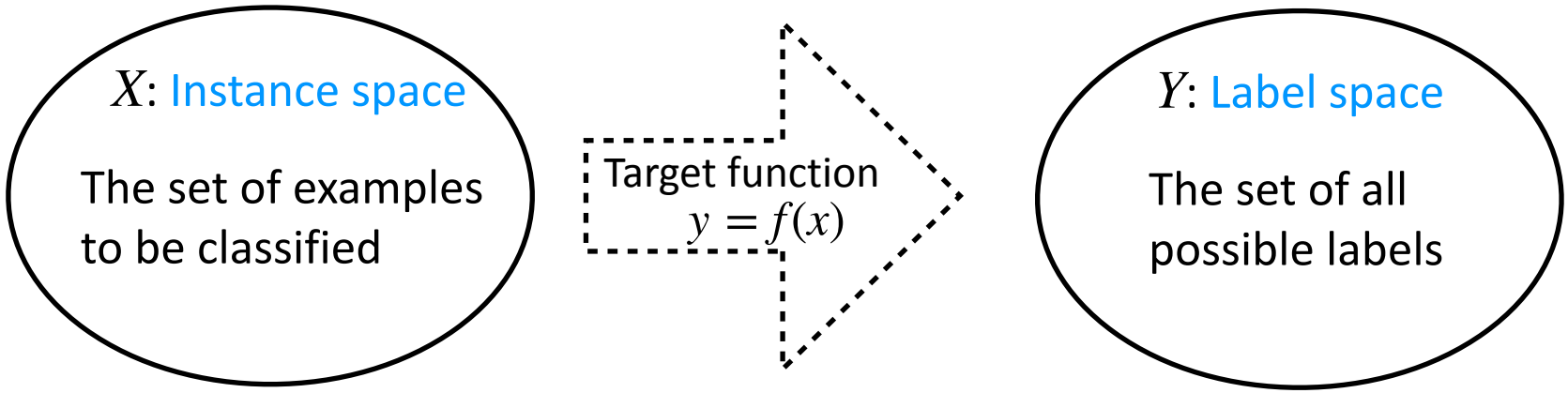
Will touch upon general principles for designing good features

- But feature definition largely domain specific
- Comes with experience

Supervised Learning

WHAT IS OUR LABEL SPACE?

The label space Y



The label space depends on the nature of the problem

Classification: the outputs are categorical

- Binary: 2 possible labels
- Multiclass: K possible labels
- Structured: labels are structured objects (sequences of labels, parse trees, …)

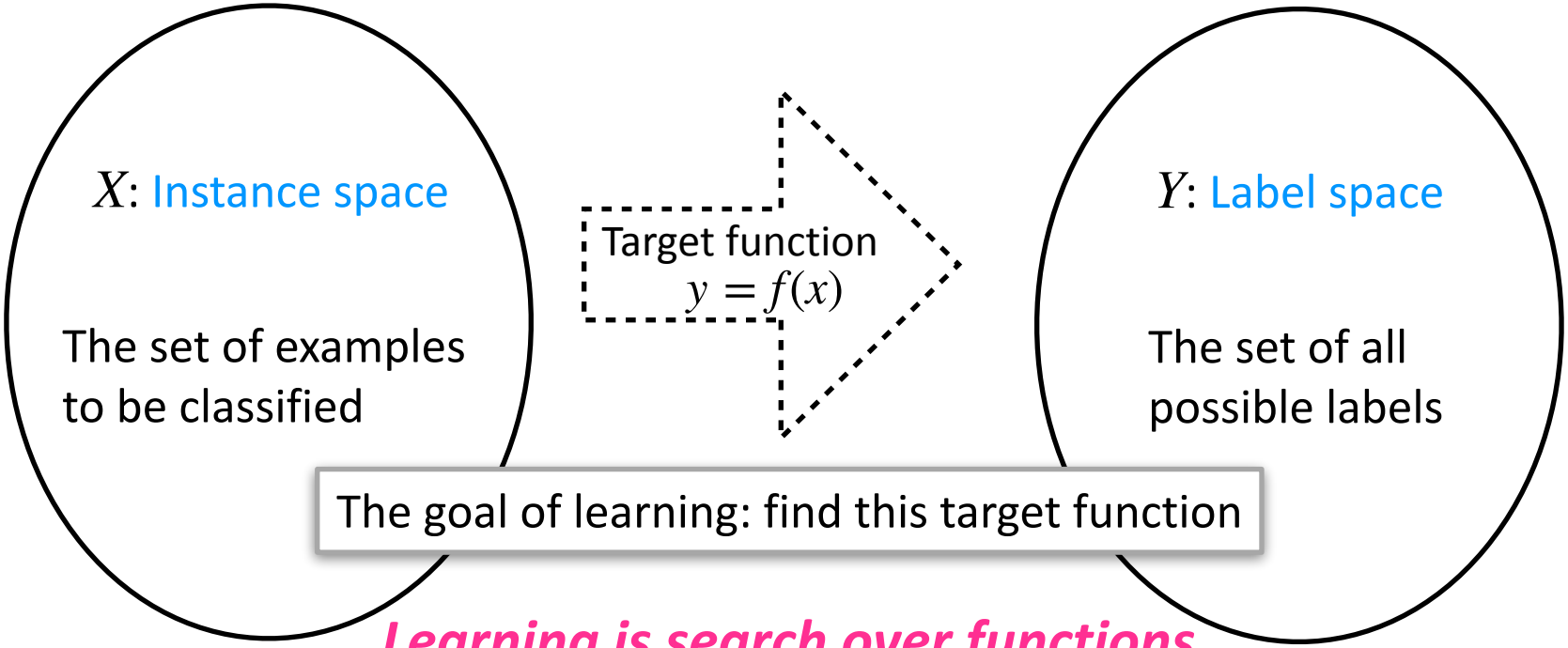
Regression: the outputs are numerical/ordinal

- Regression: labels are continuous-valued. Learn a linear/polynomial function $f(\mathbf{x})$
- Ranking: labels are ordinal. Learn an ordering $f(\mathbf{x}_1) > f(\mathbf{x}_2)$ over input

Supervised Learning

WHAT IS OUR HYPOTHESIS SPACE?

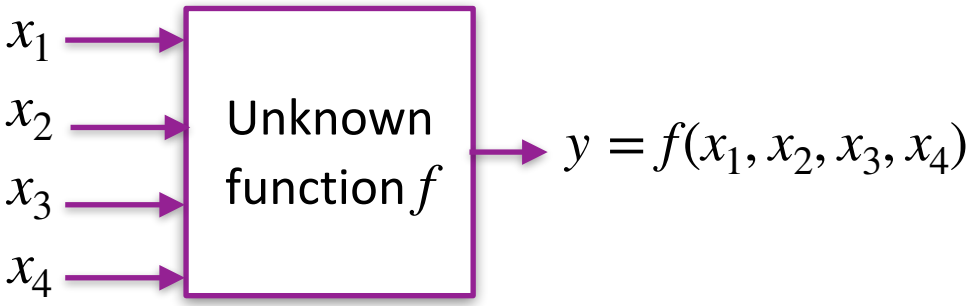
3. The hypothesis space



Learning is search over functions

Example of search over functions

0=False, 1=True



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

Can you learn this function? What is it?

The fundamental problem: machine learning is ill-posed

vumanfredi@wesleyan.edu

Is learning possible at all?

Complete ignorance: There are $2^{16} = 65536$ possible Boolean functions over 4 input features

- Why? There are 16 possible outputs. Each way to fill these 16 slots is a different function, giving 2^{16} functions.

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

vumanfredi@wesleyan.edu

Is learning possible at all?

Complete ignorance: There are $2^{16} = 65536$ possible Boolean functions over 4 input features

- Why? There are 16 possible outputs. Each way to fill these 16 slots is a different function, giving 2^{16} functions.

We have seen only 7 outputs, leaving 2^9 possibilities for f

How could we possibly know the rest without seeing every label?

- Think of an adversary filling in the labels every time you make a guess at the function

Is learning possible?

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	?

vumanfredi@wesleyan.edu

Solution: restrict the search space

The "When in doubt, make an assumption" school of thought!

Hypothesis space

- set of possible functions we consider

We were looking at the space of all Boolean functions …

Instead, choose a hypotheses space that is ***not*** all possible functions

- Only **simple conjunctions**: with 4 variables, there are only 16 conjunctions without negations
- **m-of-n rules**: pick a set of n variables. At least m of them must be true
- **Linear functions**
- **Deep neural networks**
- …

Example hypothesis space #1

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

Simple conjunctions: there are only 16 simple conjunctive rules of the form

$$g(x) = x_i \wedge x_j \wedge x_k \cdots$$

Rule	Rule
Always False	$x_2 \wedge x_3$
x_1	$x_2 \wedge x_4$
x_2	$x_3 \wedge x_4$
x_3	$x_1 \wedge x_2 \wedge x_3$
x_4	$x_1 \wedge x_2 \wedge x_4$
$x_1 \wedge x_2$	$x_1 \wedge x_3 \wedge x_4$
$x_1 \wedge x_3$	$x_2 \wedge x_3 \wedge x_4$
$x_1 \wedge x_4$	$x_1 \wedge x_2 \wedge x_3 \wedge x_4$

Exercise: how many simple conjunctions are possible when there are n inputs instead of 4?

Example hypothesis space #1

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

Simple conjunctions: there are only 16 simple conjunctive rules of the form

$$g(x) = x_i \wedge x_j \wedge x_k \cdots$$

Is there a consistent hypothesis in this space?

Rule	Counter-example	Rule	Counter-example
Always False	1001	$x_2 \wedge x_3$	0011
x_1	1100	$x_2 \wedge x_4$	0011
x_2	0100	$x_3 \wedge x_4$	1001
x_3	0110	$x_1 \wedge x_2 \wedge x_3$	0011
x_4	0101	$x_1 \wedge x_2 \wedge x_4$	0011
$x_1 \wedge x_2$	1100	$x_1 \wedge x_3 \wedge x_4$	0011
$x_1 \wedge x_3$	0011	$x_2 \wedge x_3 \wedge x_4$	0011
$x_1 \wedge x_4$	0011	$x_1 \wedge x_2 \wedge x_3 \wedge x_4$	0011

Example hypothesis space #1

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

Simple conjunctions: there are only 16 simple conjunctive rules of the form

$$g(x) = x_i \wedge x_j \wedge x_k \cdots$$

Is there a consistent hypothesis in this space?

Rule Counter-example Rule Counter-example

No simple conjunction explains the data!

(Confirm each counterexample by going through the list)

Our hypothesis space is too small and the true function we are looking or is not in it.

Solution: restrict the search space

The "When in doubt, make an assumption" school of thought!

Hypothesis space

- set of possible functions we consider

We were looking at the space of all Boolean functions. Instead, choose a hypotheses space that is ***not*** all possible functions

How do we pick hypothesis space? Use some prior knowledge or guess

What if the hypothesis space is so small that nothing in it agrees with the data? Need a hypothesis space that is flexible enough

Example hypothesis space #2

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

m-of-n rules: pick a subset with n variables. The label y is 1 if at least m of the variables are 1

Example: if at least 2 of $\{x_1, x_2, x_3, x_4\}$ are 1, then the output is 1. Otherwise the output is 0

Is there a consistent hypothesis in this space?

Exercise: How many m-of-n rules are there for 4 variables?

m-of-n rules for 4 variables

Index	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

m-of-n rules: There are 32 possible rules of the form "$y = 1$ if and only if at least m of the following n variables are 1"

Notation: 2 variables from the set on the left.

Variables	1-of	2-of	3-of	4-of	Variables	1-of	2-of	3-of	4-of
$\{x_1\}$	3	–	–	–	$\{x_2, x_4\}$	2	3	–	–
$\{x_2\}$	2	–	–	–	$\{x_2, x_4\}$	2	3	–	–
$\{x_3\}$	1	–	–	–	$\{x_3, x_4\}$	4	4	–	–
$\{x_4\}$	7	–	–	–	$\{x_1, x_2, x_3\}$	1	3	3	–
$\{x_1, x_2\}$	2	3	–	–	$\{x_1, x_2, x_4\}$	2	3	3	–
$\{x_1, x_3\}$	1	3	–	–	$\{x_1, x_3, x_4\}$	1	*	3	–
$\{x_1, x_4\}$	6	3	–	–	$\{x_2, x_3, x_4\}$	1	5	3	–
					$\{x_1, x_2, x_3, x_4\}$	1	5	3	3

Value: index of the counterexample

m-of-n rules for 4 variables

m-of-n rules: There are 32 possible rules of the form "$y = 1$ if and only if at least m of the following n variables are 1"

Index	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

Variables	1-of	2-of	3-of	4-of	Variables	1-of	2-of	3-of	4-of
$\{x_1\}$	3	–	–	–	$\{x_2, x_4\}$	2	3	–	–
$\{x_2\}$	2	–	–	–	$\{x_2, x_4\}$	2	3	–	–
$\{x_3\}$	1	–	–	–	$\{x_3, x_4\}$	4	4	–	–
$\{x_4\}$	7	–	–	–	$\{x_1, x_2, x_3\}$	1	3	3	–
$\{x_1, x_2\}$	2	3	–	–	$\{x_1, x_2, x_4\}$	2	3	3	–
$\{x_1, x_3\}$	1	3	–	–	$\{x_1, x_3, x_4\}$	1	*	3	–
$\{x_1, x_4\}$	6	3	–	–	$\{x_2, x_3, x_4\}$	1	5	3	–
					$\{x_1, x_2, x_3, x_4\}$	1	5	3	3

Found a consistent hypothesis!
(In practice find, e.g., with neural net)

General strategies for Machine Learning

Pick expressive hypothesis spaces

- Decision trees, neural networks, m-of-n functions, linear functions, …

Develop algorithms for finding a hypothesis in our hypothesis space, that fits the data well (or well enough)

Hope that the hypothesis generalizes

Perspectives on learning

Learning is the removal of our ***remaining*** uncertainty over a hypothesis space

- If we ***knew*** that the unknown function is a simple conjunction, then we could use the training data to infer which one it is.

Learning requires guessing a ***good, small*** hypothesis class

- We can start with a very small class and enlarge it until it contains an hypothesis that fits the data.
- And we could be wrong. We could find a consistent hypothesis and still be incorrect when given a new example!

Using supervised learning

1. What is our instance space?
   - What are the inputs to the problem? What are the features?
2. What is our label space?
   - What kind of learning task are we dealing with?
3. What is our hypothesis space?
   - What functions should the learning algorithm search over?
4. What is our learning algorithm?
   - How do we learn the model from the labeled data?
5. What is our loss function or evaluation metric?
   - How do we measure success? What drives learning?

Much of the rest of the semester