



10-301/10-601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

Machine Learning as Function Approximation

Matt Gormley

Lecture 2

Aug. 30, 2023

Reminders

- **Background Test**
 - Fri, Sep 1, in-class
- **Homework 1: Background**
 - **Out: Fri, Sep 1**
 - **Due: Wed, Sep 6 at 11:59pm**
 - Two parts:
 1. written part to Gradescope
 2. programming part to Gradescope
 - **unique policies for this assignment:**
 1. **unlimited submissions** for programming (i.e. keep submitting until you get 100%)
 2. we will grant (essentially) any and all extension requests

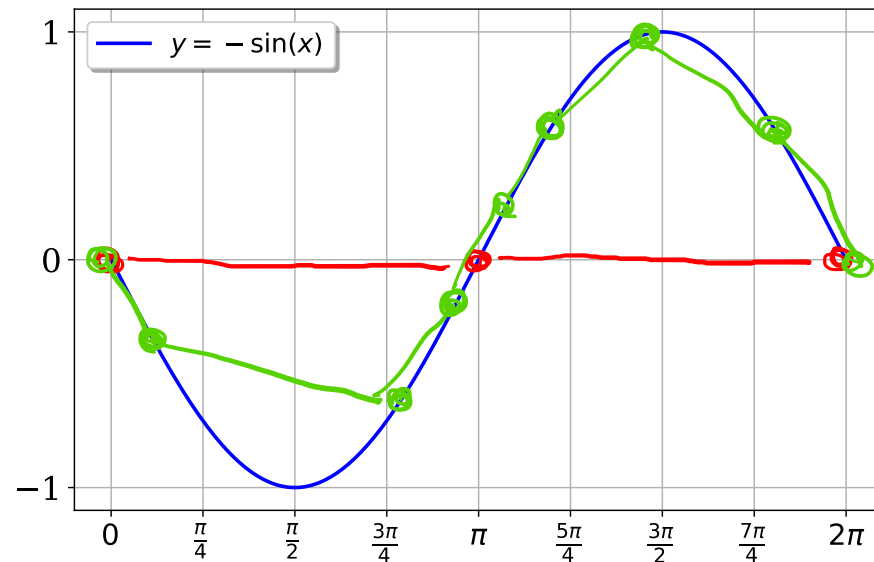
Big Ideas

1. How to formalize a learning problem
2. How to learn an expert system (i.e. Decision Tree)
3. Importance of inductive bias for generalization
4. Overfitting

FUNCTION APPROXIMATION

Function Approximation

Quiz: Implement a simple function which returns $-\sin(x)$.



- ① Taylor series approx.
- ② peice wise linear approx.
- ③ cheat
- ④ approx w/a cubic fn
- ⑤ infinite sum

A few constraints are imposed:

1. You can't call any other trigonometric functions
2. You *can* call an existing implementation of $\sin(x)$ a few times (e.g. 100) to test your solution
3. You only need to evaluate it for x in $[0, 2*\pi]$

SUPERVISED MACHINE LEARNING

Medical Diagnosis

- Setting:
 - Doctor must decide whether or not patient is sick
 - Looks at attributes of a patient to make a medical diagnosis
 - (Prescribes treatment if diagnosis is positive)
- Key problem area for Machine Learning
- Potential to reshape health care

Medical Diagnosis

Interview Transcript

Date: Jan. 15, 2023

Parties: Matt Gormley and Doctor S.

Topic: Medical decision making

Medical Diagnosis Dataset

As a (supervised) binary classification task

The diagram illustrates a supervised binary classification task. A table contains five rows of data. The first column is labeled 'i' and contains indices 1 through 5. The second column is labeled 'allergic?' and contains values '-' and '+'. The remaining four columns are labeled 'hives?', 'sneezing?', 'red eye?', and 'has cat?', and contain values 'Y' and 'N'. A blue bracket above the table groups the first column as 'labels' and the other four columns as 'features'. A yellow bracket on the left side groups the five rows as 'examples'.

	labels	features			
i	allergic?	hives?	sneezing?	red eye?	has cat?
1	-	Y	N	N	N
2	-	N	Y	N	N
3	+	Y	Y	N	N
4	-	Y	N	Y	Y
5	+	N	Y	Y	N

Medical Diagnosis Dataset

As a (supervised) binary classification task

	labels	features			
i	allergic?	hives?	sneezing?	red eye?	has cat?
1	-	Y	N	N	N
2	-	N	Y	N	N
3	+	Y	Y	N	N
4	-	Y	N	Y	Y
5	+	N	Y	Y	N

Medical Diagnosis Dataset

As a (supervised) binary classification task

		labels	features			
		allergic?	hives?	sneezing?	red eye?	has cat?
examples	i					
	1	-	Y	N	N	N
	2	-	N	Y	N	N
	3	+	Y	Y	N	N
	4	-	Y	N	Y	Y
	5	+	N	Y	Y	N

Medical Diagnosis Dataset

As a (supervised) classification task

		labels	features			
		allergy	hives?	sneezing?	red eye?	has cat?
examples	i					
	1	none	Y	N	N	N
	2	none	N	Y	N	N
	3	dust	Y	Y	N	N
	4	none	Y	N	Y	Y
5	mold	N	Y	Y	N	

Medical Diagnosis Dataset

As a (supervised)
output

regression task

features

examples

	treatment	features			
i	cost	hives?	sneezing?	red eye?	has cat?
1	\$10	Y	N	N	N
2	\$25	N	Y	N	N
3	\$1000	Y	Y	N	N
4	\$25	Y	N	Y	Y
5	\$2000	N	Y	Y	N

Medical Diagnosis Dataset

As a (supervised) binary classification task

The diagram illustrates a supervised binary classification task using a dataset. The dataset is presented as a table with 5 rows of examples. The first column, labeled 'i', represents the example index. The second column, labeled 'allergic?', represents the label. The remaining four columns, labeled 'hives?', 'sneezing?', 'red eye?', and 'has cat?', represent the features. A blue bracket above the table groups the 'allergic?' column as 'labels' and the other four columns as 'features'. A yellow bracket on the left side groups the five rows as 'examples'.

i	allergic?	hives?	sneezing?	red eye?	has cat?
1	-	Y	N	N	N
2	-	N	Y	N	N
3	+	Y	Y	N	N
4	-	Y	N	Y	Y
5	+	N	Y	Y	N

Medical Diagnosis Dataset

Doctor diagnoses the patient as sick or not $y \in \{+, -\}$
based on attributes of the patient x_1, x_2, \dots, x_M

	y	x_1	x_2	x_3	x_4
i	allergic?	hives?	sneezing?	red eye?	has cat?
1	-	Y	N	N	N

Medical Diagnosis Dataset

Doctor diagnoses the patient as sick or not $y \in \{+, -\}$
based on attributes of the patient x_1, x_2, \dots, x_M

	y	x_1	x_2	x_3	x_4
i	allergic?	hives?	sneezing?	red eye?	has cat?
1	-	Y	N	N	N
2	-	N	Y	N	N
3	+	Y	Y	N	N
4	-	Y	N	Y	Y
5	+	N	Y	Y	N

Medical Diagnosis Dataset

Doctor diagnoses the patient as sick or not $y \in \{+, -\}$
based on attributes of the patient x_1, x_2, \dots, x_M

	y	x_1	x_2	x_3	x_4
i	allergic?	hives?	sneezing?	red eye?	has cat?
1	$y^{(1)} -$	$x_1^{(1)} Y$	$x_2^{(1)} N$	$x_3^{(1)} N$	$x_4^{(1)} N$
2	$y^{(2)} -$	$x_1^{(2)} N$	$x_2^{(2)} Y$	$x_3^{(2)} N$	$x_4^{(2)} N$
3	$y^{(3)} +$	$x_1^{(3)} Y$	$x_2^{(3)} Y$	$x_3^{(3)} N$	$x_4^{(3)} N$
4	$y^{(4)} -$	$x_1^{(4)} Y$	$x_2^{(4)} N$	$x_3^{(4)} Y$	$x_4^{(4)} Y$
5	$y^{(5)} +$	$x_1^{(5)} N$	$x_2^{(5)} Y$	$x_3^{(5)} Y$	$x_4^{(5)} N$

Medical Diagnosis Dataset

Doctor diagnoses the patient as sick or not $y \in \{+, -\}$
based on attributes of the patient x_1, x_2, \dots, x_M

	y	x_1	x_2	x_3	x_4	
i	allergic?	hives?	sneezing?	red eye?	has cat?	
1	$y^{(1)}$ -	$x_1^{(1)}$ Y	$x_2^{(1)}$ N	$x_3^{(1)}$ N	$x_4^{(1)}$ N	$x^{(1)}$
2	$y^{(2)}$ -	$x_1^{(2)}$ N	$x_2^{(2)}$ Y	$x_3^{(2)}$ N	$x_4^{(2)}$ N	$x^{(2)}$
3	$y^{(3)}$ +	$x_1^{(3)}$ Y	$x_2^{(3)}$ Y	$x_3^{(3)}$ N	$x_4^{(3)}$ N	$x^{(3)}$
4	$y^{(4)}$ -	$x_1^{(4)}$ Y	$x_2^{(4)}$ N	$x_3^{(4)}$ Y	$x_4^{(4)}$ Y	$x^{(4)}$
5	$y^{(5)}$ +	$x_1^{(5)}$ N	$x_2^{(5)}$ Y	$x_3^{(5)}$ Y	$x_4^{(5)}$ N	$x^{(5)}$

$N = 5$ training examples

$M = 4$ attributes

ML as Function Approximation

Whiteboard

– ML as Function Approximation

- Problem setting
- Input space
- Output space
- Unknown target function
- Hypothesis space
- Training examples
- Goal of Learning

ML as Function Approximation

Problem Setting

- Set of possible \mathcal{X} (all possible feature vectors)
- Set of possible \mathcal{Y} (all possible labels)
- Unknown target function $c^* : \mathcal{X} \rightarrow \mathcal{Y}$
- Set of candidate hypotheses
 $\mathcal{H} = \text{all possible hypotheses} = \{h \mid h : \mathcal{X} \rightarrow \mathcal{Y}\}$

Learner is given:

- Training examples $D_{\text{train}} = \{(\vec{x}^{(1)}, y^{(1)}), (\vec{x}^{(2)}, y^{(2)}), \dots, (\vec{x}^{(N)}, y^{(N)})\}$
of unknown target function $y^{(i)} = c^*(\vec{x}^{(i)}) \quad \forall i \in \{1, \dots, N\}$
- $N = \#$ of training examples, $M = \#$ features $= |\vec{x}^{(i)}|$

Learner produces:

- Hypothesis $h \in \mathcal{H}$ that "best approximates" c^*

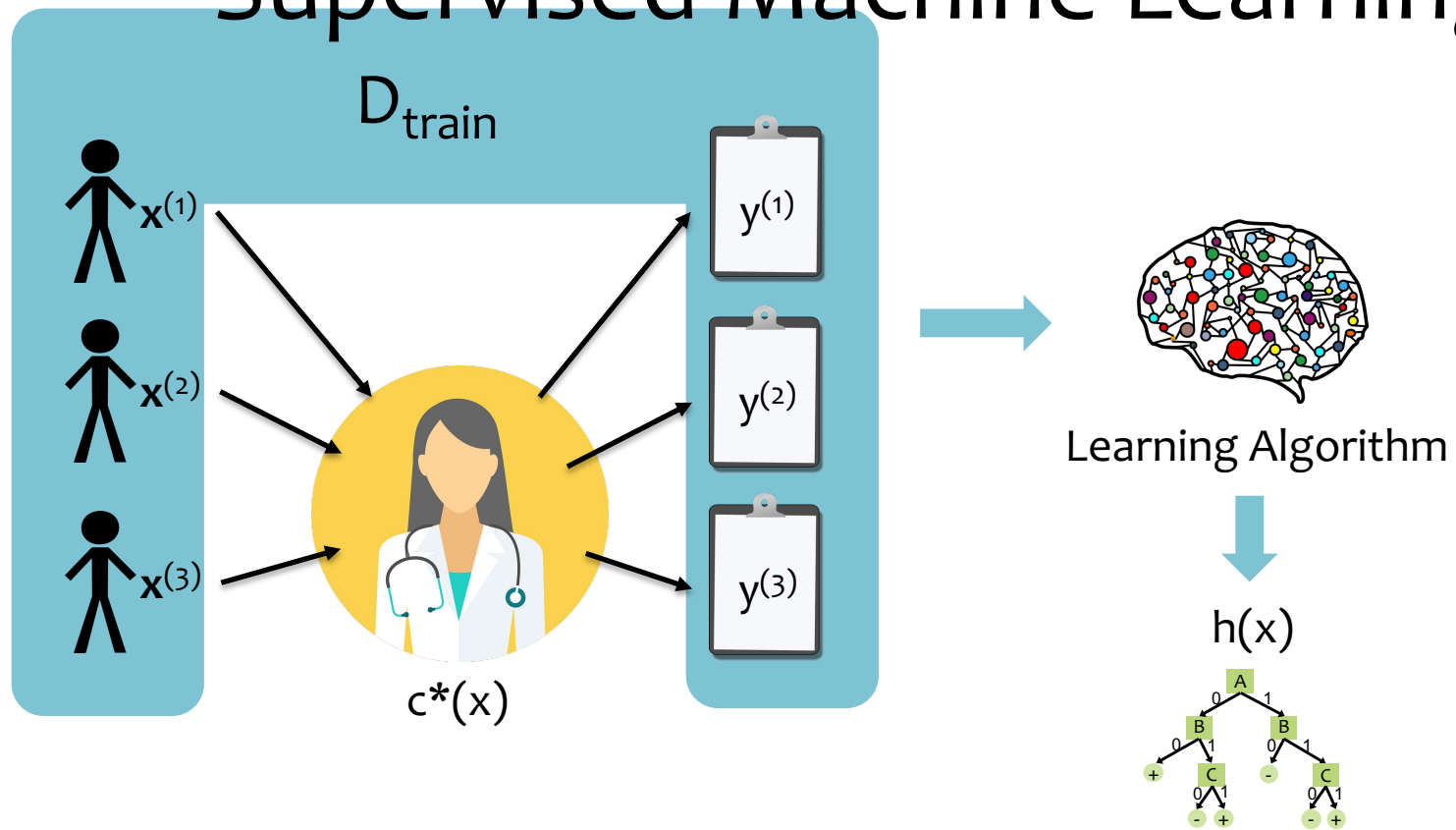
To Evaluate:

- Loss function $l : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ measures how "bad" predictions $\hat{y} = h(\vec{x})$ are compared to $c^*(\vec{x})$
- Another dataset $D_{\text{test}} = \{(\vec{x}^{(1)}, y^{(1)}), \dots, (\vec{x}^{(N')}, y^{(N')})\}$
- Evaluate the average loss of $h(\vec{x})$ on D_{test}

Aside: Function
Types

$$f(x_1, x_2, x_3) = (x_1 x_2)^2 + x_3$$
$$f : \mathbb{R}^3 \rightarrow \mathbb{R}$$

Supervised Machine Learning



Medical Diagnosis Dataset

Doctor diagnoses the patient as sick or not $y \in \{+, -\}$ based on attributes of the patient x_1, x_2, \dots, x_M



predictions

	y	x ₁	x ₂	x ₃	x ₄	
i	allergic?	hives?	sneezing?	red eye?	has cat?	
1	y ⁽¹⁾ -	x ₁ ⁽¹⁾ Y	x ₂ ⁽¹⁾ N	x ₃ ⁽¹⁾ N	x ₄ ⁽¹⁾ N	x ⁽¹⁾
2	y ⁽²⁾ -	x ₁ ⁽²⁾ N	x ₂ ⁽²⁾ Y	x ₃ ⁽²⁾ N	x ₄ ⁽²⁾ N	x ⁽²⁾
3	y ⁽³⁾ +	x ₁ ⁽³⁾ Y	x ₂ ⁽³⁾ Y	x ₃ ⁽³⁾ N	x ₄ ⁽³⁾ N	x ⁽³⁾
4	y ⁽⁴⁾ -	x ₁ ⁽⁴⁾ Y	x ₂ ⁽⁴⁾ N	x ₃ ⁽⁴⁾ Y	x ₄ ⁽⁴⁾ Y	x ⁽⁴⁾
5	y ⁽⁵⁾ +	x ₁ ⁽⁵⁾ N	x ₂ ⁽⁵⁾ Y	x ₃ ⁽⁵⁾ Y	x ₄ ⁽⁵⁾ N	x ⁽⁵⁾

N = 5 training examples

M = 4 attributes

Example hypothesis function:

$$h(\mathbf{x}) = \begin{cases} + & \text{if sneezing} = Y \\ - & \text{otherwise} \end{cases}$$

Supervised Machine Learning

- **Problem Setting**

- Set of possible inputs, $\mathbf{x} \in \mathcal{X}$ (all possible patients)
- Set of possible outputs, $y \in \mathcal{Y}$ (all possible diagnoses)
- Exists an unknown target function, $c^* : \mathcal{X} \rightarrow \mathcal{Y}$
(the doctor's brain)
- Set, \mathcal{H} , of candidate hypothesis functions, $h : \mathcal{X} \rightarrow \mathcal{Y}$
(all possible decision trees)

- **Learner is given** N training examples

$$D = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(N)}, y^{(N)})\}$$

where $y^{(i)} = c^*(\mathbf{x}^{(i)})$

(history of patients and their diagnoses)

- **Learner produces** a hypothesis function, $\hat{y} = h(\mathbf{x})$, that best approximates unknown target function $y = c^*(\mathbf{x})$ on the training data

Supervised Machine Learning

- **Problem Setting**

- Set of possible inputs, $\mathbf{x} \in \mathcal{X}$ (all possible patients)
- Set of possible outputs, $y \in \mathcal{Y}$ (all possible diagnoses)
- Exists an unknown target function, $c^* : \mathcal{X} \rightarrow \mathcal{Y}$
(the doctor's brain)
- Set, \mathcal{H} , of candidate hypothesis
(all possible decision trees)

- **Learner is given** N training data
 $D = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(N)}, y^{(N)})\}$
where $y^{(i)} = c^*(\mathbf{x}^{(i)})$
(history of patients and the doctor's diagnosis)

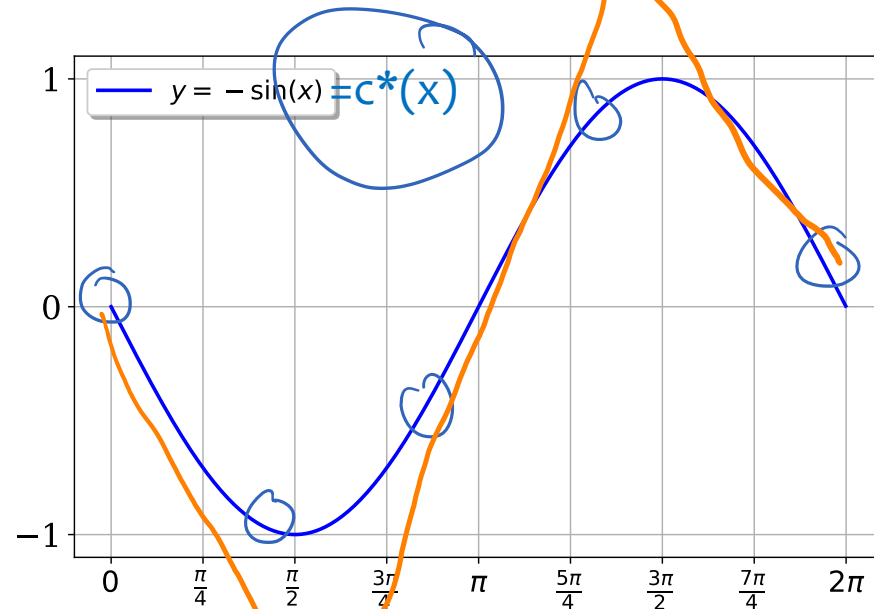
- **Learner produces** a hypothesis h that
approximates unknown target function c^* on training data

Two important settings we'll consider:

1. **Classification:** the possible outputs are **discrete**
2. **Regression:** the possible outputs are **real-valued**

Function Approximation

Quiz: Implement a simple function which returns $-\sin(x)$.



A few constraints are imposed:

1. You can't call any other trigonometric functions
2. You *can* call an existing implementation of $\sin(x)$ a few times (e.g. 100) to test your solution
3. You only need to evaluate it for x in $[0, 2*\pi]$

Supervised Machine Learning

- **Problem Setting**

- Set of possible inputs, $\mathbf{x} \in \mathcal{X}$ (all values in $[0, 2\pi]$)
- Set of possible outputs, $y \in \mathcal{Y}$ (all values in $[-1, 1]$) or \mathbb{R}
- Exists an unknown target function, $c^* : \mathcal{X} \rightarrow \mathcal{Y}$
($c^*(x) = \sin(x)$)
- Set, \mathcal{H} , of candidate hypothesis functions, $h : \mathcal{X} \rightarrow \mathcal{Y}$
(all possible piecewise linear functions)

- **Learner is given** N training examples

$$D = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \dots, (\mathbf{x}^{(N)}, y^{(N)})\}$$

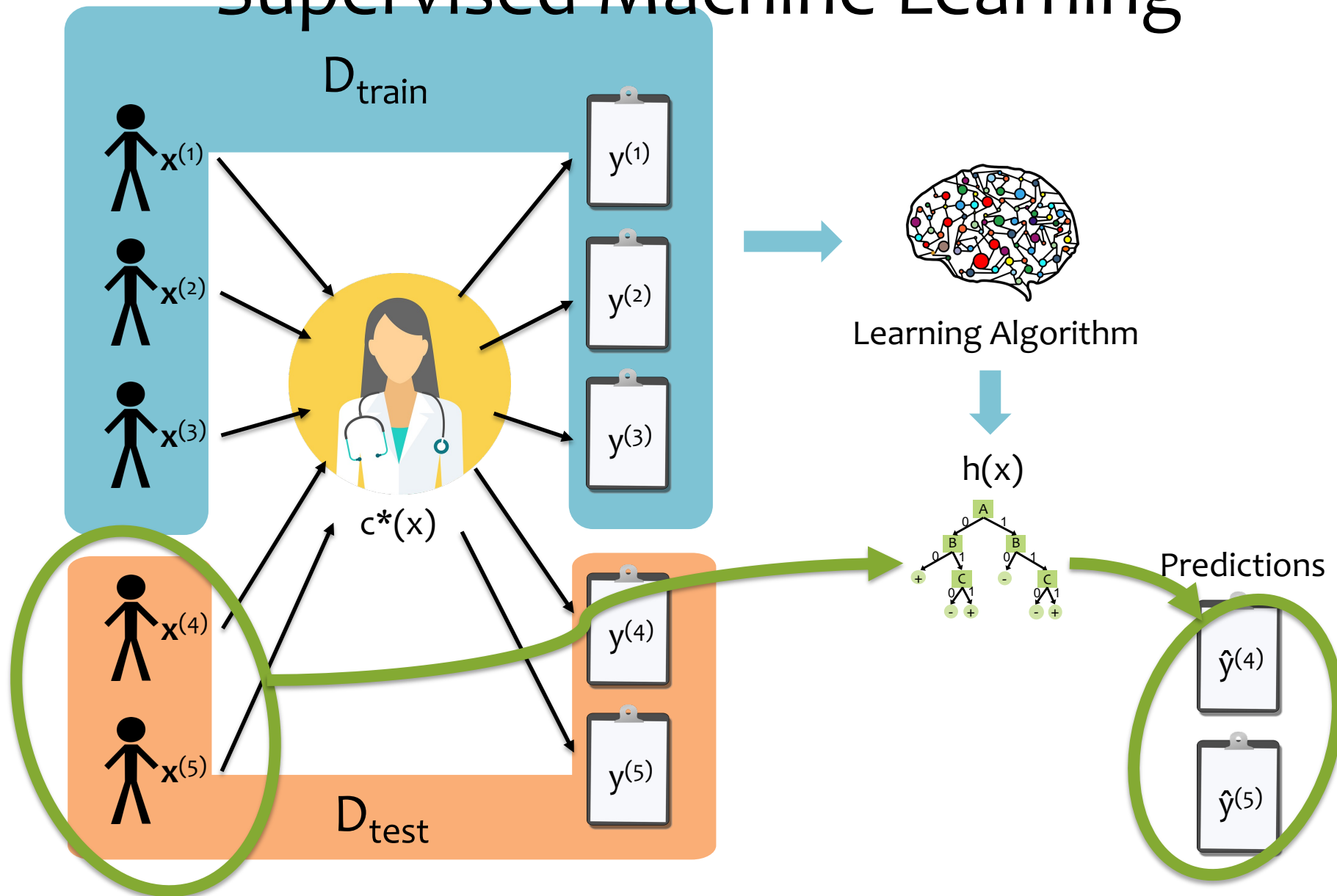
where $y^{(i)} = c^*(\mathbf{x}^{(i)})$

(true values of $\sin(x)$ for a few random x 's)

- **Learner produces** a hypothesis function, $\hat{y} = h(x)$, that best approximates unknown target function $y = c^*(x)$ on the training data

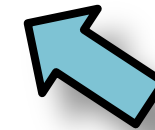
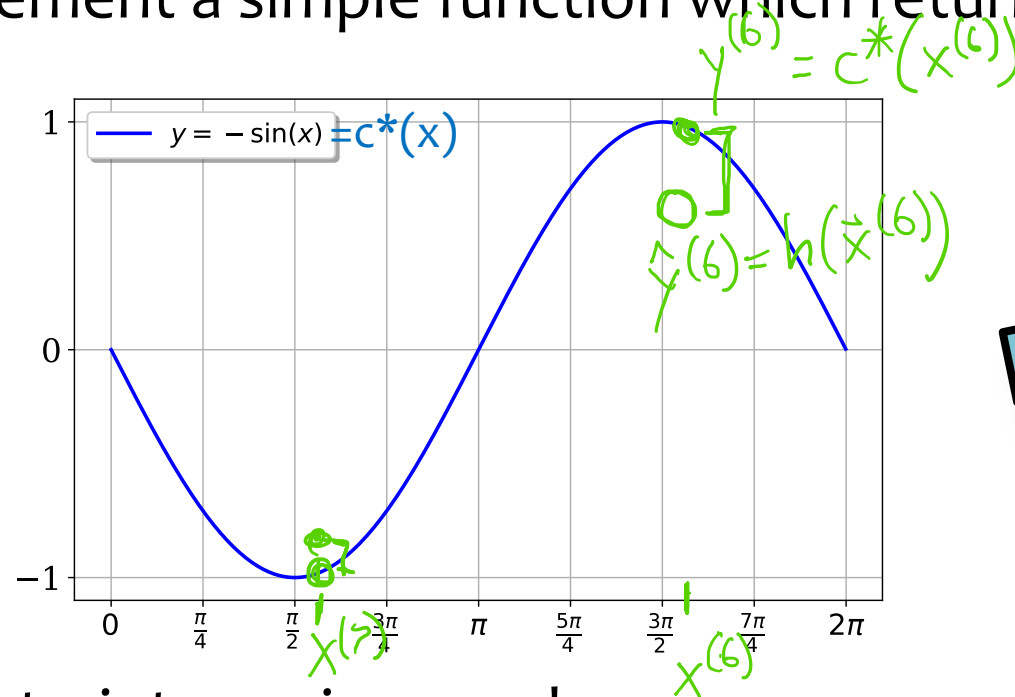
EVALUATION OF MACHINE LEARNING ALGORITHM

Supervised Machine Learning



Function Approximation

Quiz: Implement a simple function which returns $-\sin(x)$.



How well does $h(x)$ approximate $c^*(x)$?

A few constraints are imposed:

1. You can't call any other trigonometric functions
2. You *can* call an existing implementation of $\sin(x)$ a few times (e.g. 100) to test your solution
3. You only need to evaluate it for x in $[0, 2*\pi]$

Evaluation of ML Algorithms

- **Definition: loss function, $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$**

- Defines how “bad” predictions, $\hat{y} = h(\mathbf{x})$, are compared to the true labels, $y = c^*(\mathbf{x})$

- Common choices

1. Squared loss (for regression): $\ell(y, \hat{y}) = (y - \hat{y})^2$

2. Binary or 0-1 loss (for classification):

$$\ell(y, \hat{y}) = \mathbb{1}(y \neq \hat{y}) = \begin{cases} 1 & \text{if incorrect} \\ 0 & \text{otherwise} \end{cases}$$

Absolute loss
 $\ell(y, \hat{y}) = |y - \hat{y}|$

- **Error rate:**

$$\text{err}(h, \mathcal{D}) = \frac{1}{N} \sum_{n=1}^N \mathbb{1}(y^{(n)} \neq \hat{y}^{(n)})$$

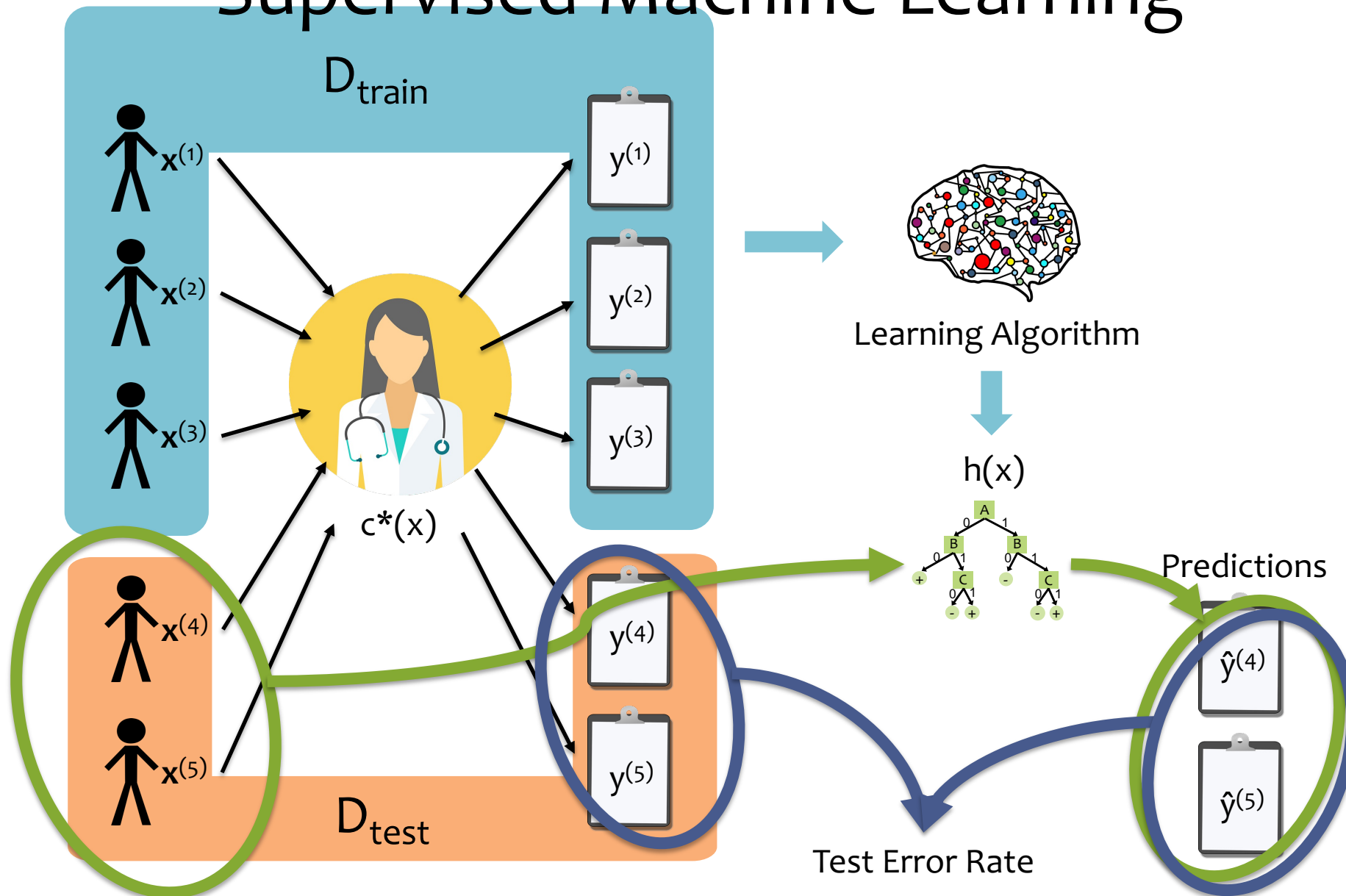
- Q: How do we evaluate a machine learning algorithm?

A: Check its error rate on a separate test dataset, $\mathcal{D}_{\text{test}}$

Aside: Indicator Fns.

$$\mathbb{1}(\text{proposition}) = \begin{cases} 1 & \text{if prop. is true} \\ 0 & \text{if prop. is false} \end{cases}$$

Supervised Machine Learning




Error Rate

- Consider a hypothesis h its...

... error rate over all training data: $\text{error}(h, D_{\text{train}})$

... error rate over all test data: $\text{error}(h, D_{\text{test}})$ ←

... true error over all data: $\text{error}_{\text{true}}(h)$ ↗



This is the quantity we care most about!
But, in practice, $\text{error}_{\text{true}}(h)$ is **unknown**.

Majority Vote Classifier Example

$h(A, B) = +$ always

Dataset:

Output Y, Attributes A and B

	Y	A	B
+	-	1	0
+	-	1	0
+	+	1	0
+	+	1	0
+	+	1	1
+	+	1	1
+	+	1	1
+	+	1	1

$\frac{2}{8}$

X
X

In-Class Exercise

What is the **training error** (i.e. *error rate on the training data*) of the **majority vote classifier** on this dataset?

Choose one of:

$\{0/8, 1/8, 2/8, \dots, 8/8\}$

LEARNING ALGORITHMS FOR SUPERVISED CLASSIFICATION

Algorithms for Classification

Algorithm 1 majority vote: predict the most common label in the training dataset

	y	x ₁	x ₂	x ₃	x ₄
predictions	allergic?	hives?	sneezing?	red eye?	has cat?
-	-	Y	N	N	N
-	-	N	Y	N	N
-	+	Y	Y	N	N
-	-	Y	N	Y	Y
-	+	N	Y	Y	N

Algorithms for Classification

Algorithm 2 memorizer: if a set of features exists in the training dataset, predict its corresponding label; otherwise, predict a random label

	y	x ₁	x ₂	x ₃	x ₄
predictions	allergic?	hives?	sneezing?	red eye?	has cat?
<u>-</u>	-	Y	N	N	N
<u>-</u>	-	N	Y	N	N
<u>+</u>	+	Y	Y	N	N
<u>-</u>	-	Y	N	Y	Y
<u>+</u>	+	N	Y	Y	N

The memorizer always gets zero training error!

Algorithms for Classification

Question:

If we have 100 features, how many patients does the memorizer need to see to ensure zero test error?

Answer:

$$|X| = 2^{100}$$

Algorithm 1: Majority Vote

Pseudocode

def train(D):

store $v = \text{majority-vote}(D)$

= the class $y \in \mathcal{Y}$ that appears most often in D

def $h(\vec{x})$:

return v

def predict(D_{test}):

for $(\vec{x}^{(i)}, y^{(i)}) \in D_{\text{test}}$:

$\hat{y}^{(i)} = h(\vec{x}^{(i)})$

} reuse this for
any classifier today

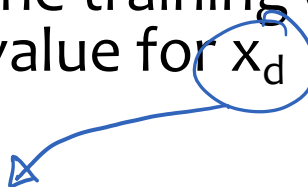
Algorithm 2: Memorizer

Pseudocode

```
def train(D):  
    store dataset D  
  
def h( $\vec{x}$ ):  
    if  $\exists \vec{x}^{(i)} \in D$  s.t.  $\vec{x} = \vec{x}^{(i)}$ :  
        return  $y^{(i)}$   
    else:  
        return random  $y \in \mathcal{Y}$ 
```

Algorithms for Classification

Algorithm 3 decision stump: based on a single feature, x_d , predict the most common label in the training dataset among all data points that have the same value for x_d



	y	x_1	x_2	x_3	x_4
	allergic?	hives?	sneezing?	red eye?	has cat?
-	-	Y	N	N	N
+	-	N	Y	N	N
+	+	Y	Y	N	N
-	-	Y	N	Y	Y
+	+	N	Y	Y	N

predictions

X

Nonzero training error, but perhaps still better than the memorizer

Example decision stump:
$$h(\mathbf{x}) = \begin{cases} + & \text{if sneezing} = Y \\ - & \text{otherwise} \end{cases}$$

Algorithm 3: Decision Stump

Pseudocode

def train(D):

① pick an attribute, m

② divide dataset D on x_m

$$D^{(0)} = \{(\vec{x}^{(i)}, y^{(i)}) \in D \mid x_m = 0\}$$

$$D^{(1)} = \{(\vec{x}^{(i)}, y^{(i)}) \in D \mid x_m = 1\}$$

③ two votes

$$v^{(0)} = \text{majority_vote}(D^{(0)})$$

$$v^{(1)} = \text{majority_vote}(D^{(1)})$$

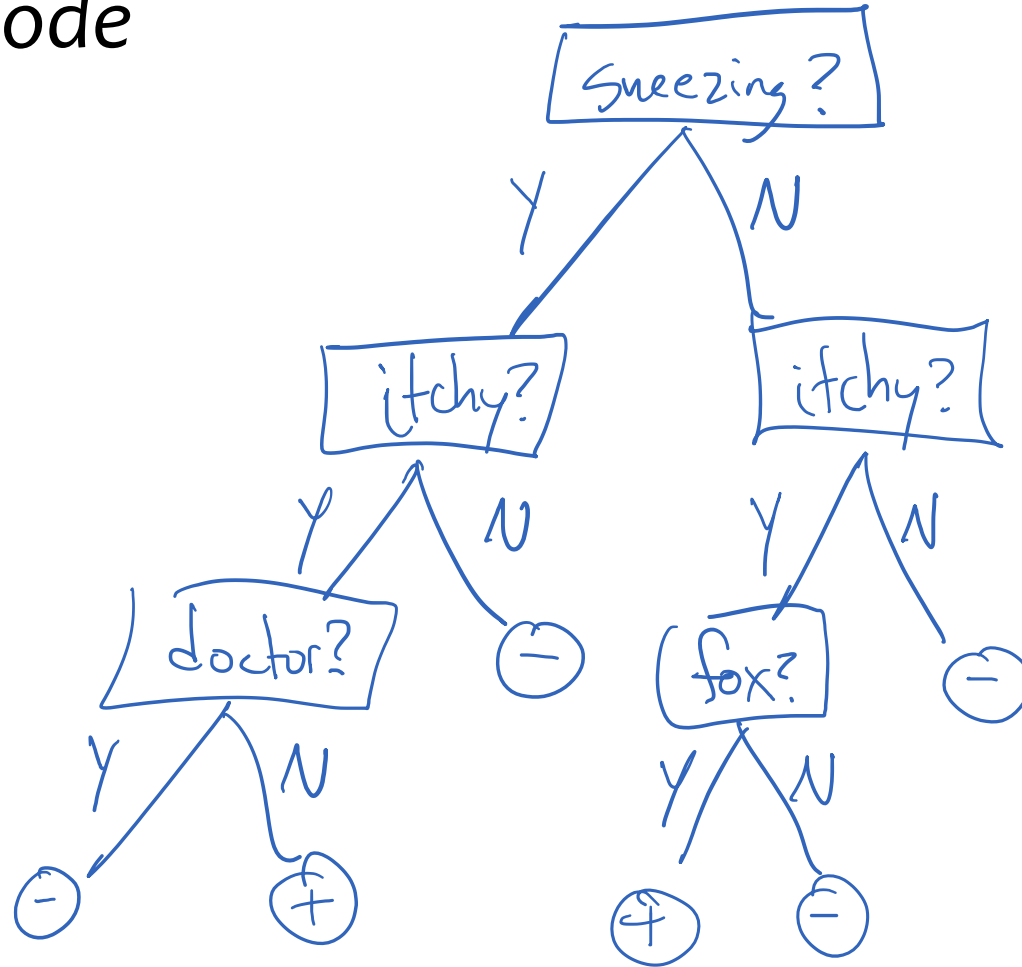
def $h(\vec{x})$:

if $x_m = 0$: return $v^{(0)}$

if $x_m = 1$: return $v^{(1)}$

Algorithm 4: Decision Tree (preview)

Pseudocode



Tree to Predict C-Section Risk

Learned from medical records of 1000 women (Sims et al., 2000)

Negative examples are C-sections

```
[833+,167-] .83+ .17-
{ Fetal_Presentation = 1: [822+,116-] .88+ .12-
  | Previous_Csection = 0: [767+,81-] .90+ .10-
  | | Primiparous = 0: [399+,13-] .97+ .03-
  | | Primiparous = 1: [368+,68-] .84+ .16-
  | | | Fetal_Distress = 0: [334+,47-] .88+ .12-
  | | | | Birth_Weight < 3349: [201+,10.6-] .95+ .05-
  | | | | Birth_Weight >= 3349: [133+,36.4-] .78+ .22-
  | | | Fetal_Distress = 1: [34+,21-] .62+ .38-
  | Previous_Csection = 1: [55+,35-] .61+ .39-
  { Fetal_Presentation = 2: [3+,29-] .11+ .89-
  { Fetal_Presentation = 3: [8+,22-] .27+ .73-
```