Machine Learning Department
School of Computer Science
Carnegie Mellon University

Machine Learning as Function Approximation

Matt Gormley & Henry Chai

Lecture 2

Jan. 15, 2025

Reminders

- **Background Test**
  - **Fri, Sep 1, in-class**
- **Homework 1: Background**
  - **Out: Mon, Jan 13**
  - **Due: Wed, Jan 22 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

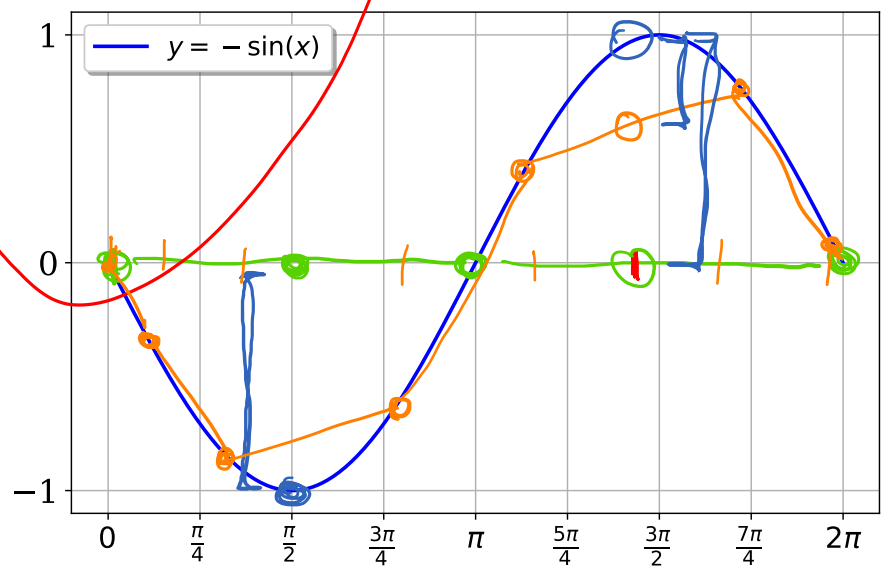
Big Ideas

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).



piecewise linear approximation

1. cheat and call sin(x) anyway
2. Taylor series approximation
3. geometric solutions

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

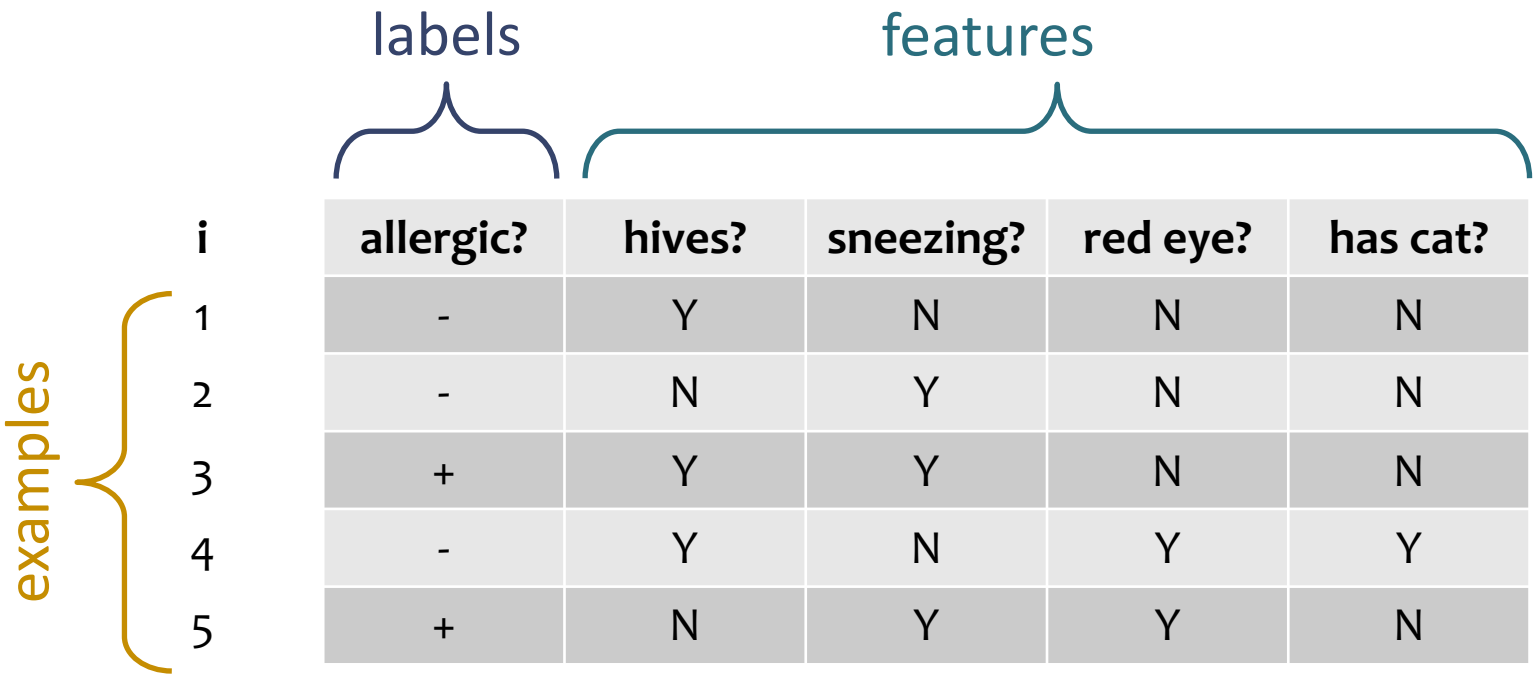
- Setting:
  - Doctor must decide whether or not patient is sick
  - Looks at attributes of a patient to make a medical diagnosis
  - (Prescribes treatment if diagnosis is positive)
- Key problem area for Machine Learning
- Potential to reshape health care

Medical Diagnosis

Interview Transcript
Date: Jan.7, 2025
Parties: Matt Gormley and Doctor S.
Topic: Medical decision making

Medical Diagnosis Dataset

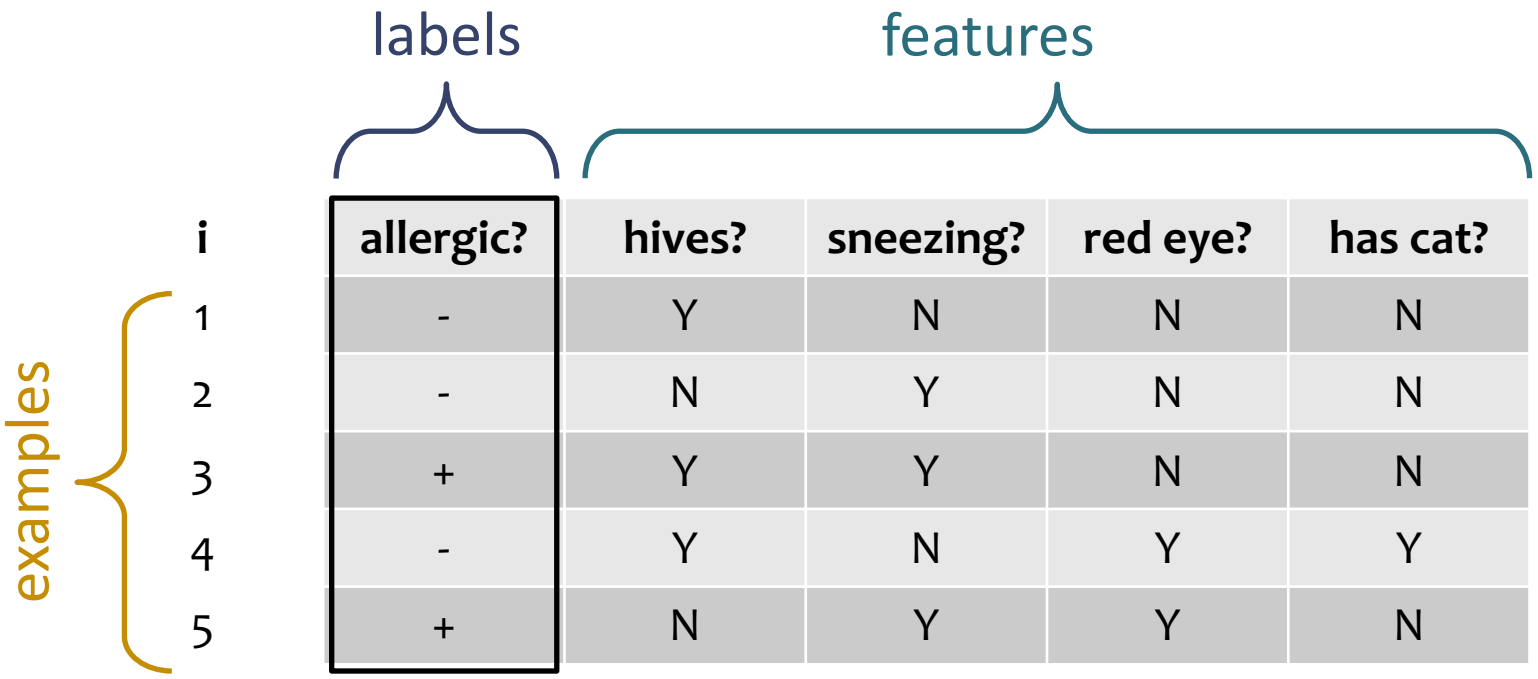
As a (supervised) binary classification task



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

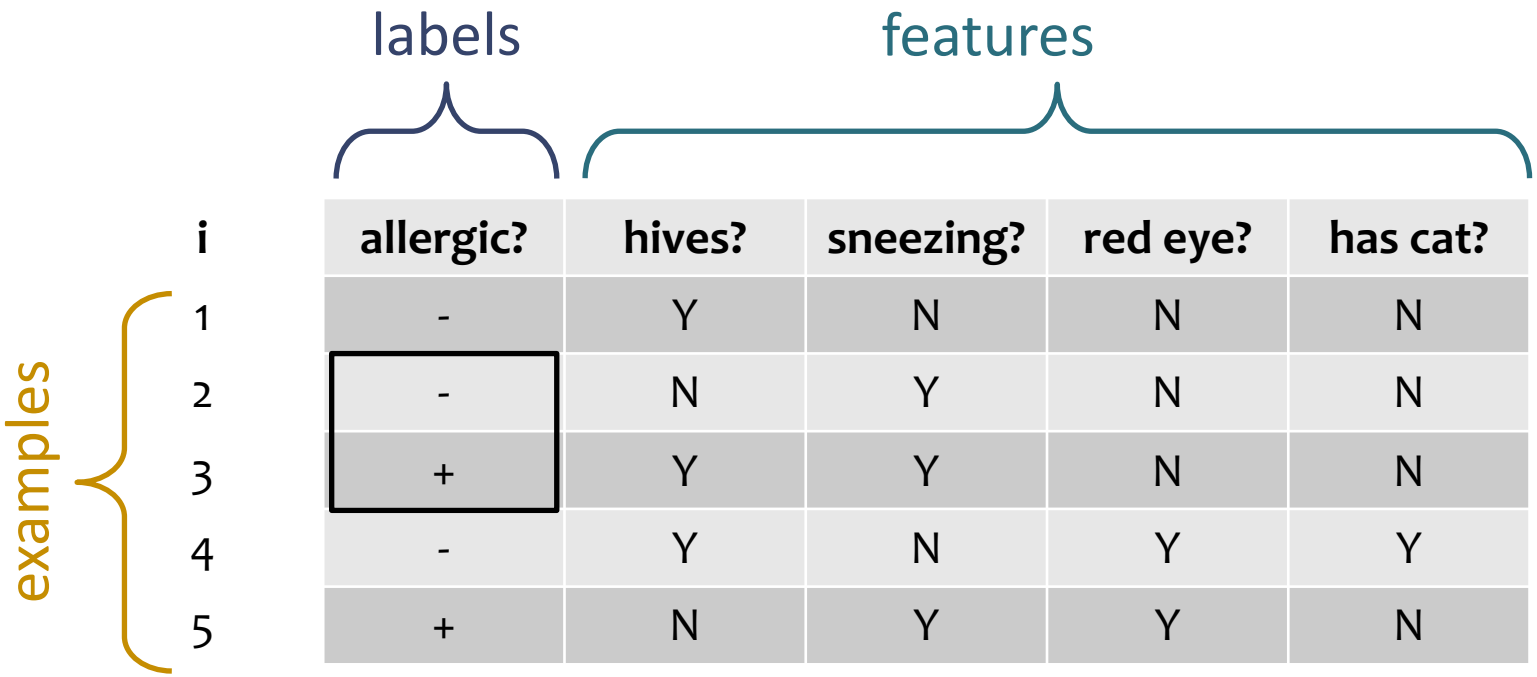
As a (supervised) binary classification task



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

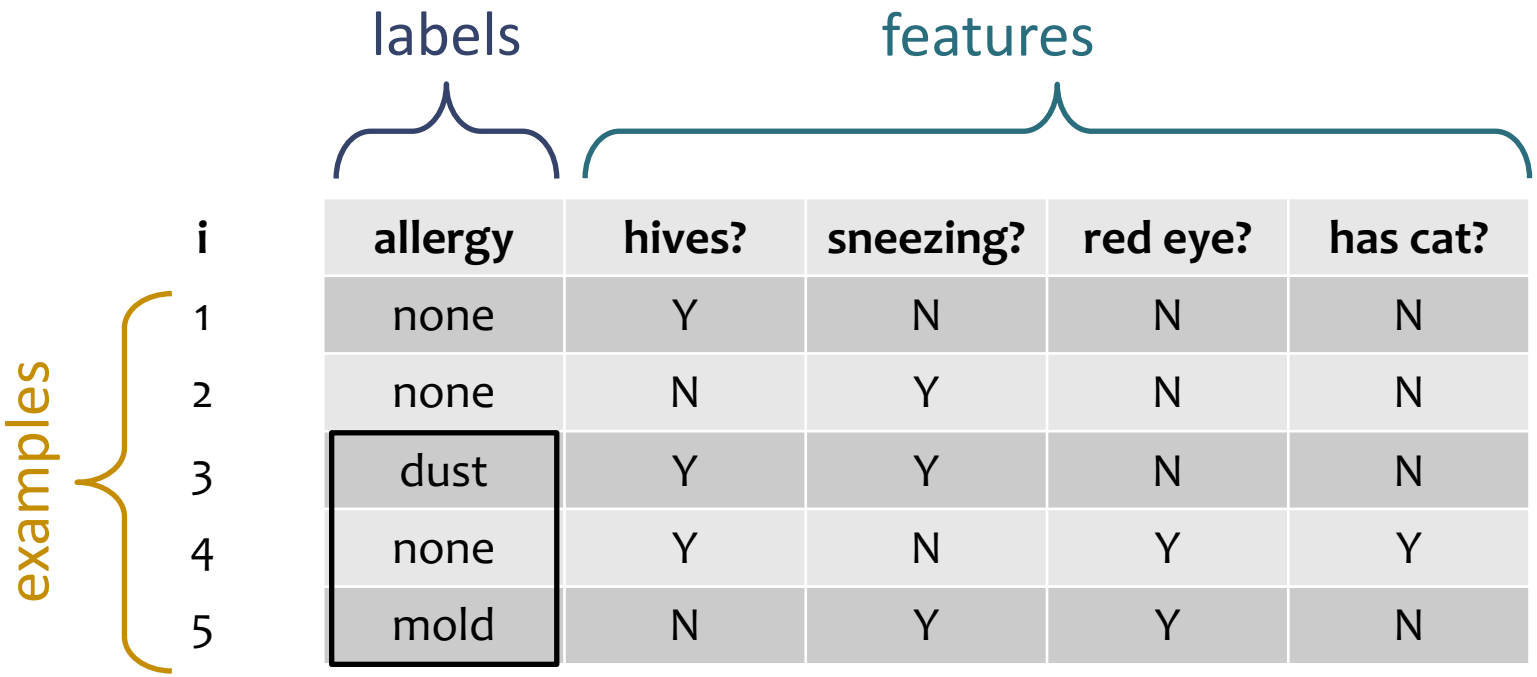
As a (supervised) binary classification task



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) classification task



i	allergy	hives?	sneezing?	red eye?	has cat?
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) regression task

output

features

examples

i	treatment 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

labels

features

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 $x_1, x_2, \ldots, x_M$

	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 $x_1, x_1, \ldots, x_M$

	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 $x_1, x_1, \ldots, x_M$

	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 $x_1, x_1, \ldots, x_M$

	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	$\mathbf{x}^{(1)}$
2	$y^{(2)}$ -	$x_1^{(2)}$ N	$x_2^{(2)}$ Y	$x_3^{(2)}$ N	$x_4^{(2)}$ N	$\mathbf{x}^{(2)}$
3	$y^{(3)}$ +	$x_1^{(3)}$ Y	$x_2^{(3)}$ Y	$x_3^{(3)}$ N	$x_4^{(3)}$ N	$\mathbf{x}^{(3)}$
4	$y^{(4)}$ -	$x_1^{(4)}$ Y	$x_2^{(4)}$ N	$x_3^{(4)}$ Y	$x_4^{(4)}$ Y	$\mathbf{x}^{(4)}$
5	$y^{(5)}$ +	$x_1^{(5)}$ N	$x_2^{(5)}$ Y	$x_3^{(5)}$ Y	$x_4^{(5)}$ N	$\mathbf{x}^{(5)}$

N = 5 training examples

M = 4 attributes

ML as Function Approximation

Problem Setting:

- Set of possible inputs $\mathcal{X}$ (all possible feature vectors)
- Set of possible outputs $\mathcal{Y}$ (all possible labels)
- Unknown target function $c^* : \mathcal{X} \to \mathcal{Y}$
- Set of candidate hypotheses

  $\mathcal{H}$ = all possible hypothesis functions = $\{h \mid h : \mathcal{X} \to \mathcal{Y}\}$

Aside: Function Types

$f(x_1, x_2, x_3) = (x_1 x_2)^2 + x_3$

$f : \mathbb{R}^3 \to \mathbb{R}$

($\mathbb{R}^3$: input; $\mathbb{R}$: output)

Learner is given:

- Training examples $\mathcal{D} = \{(\vec{x}^{(1)}, y^{(1)}), (\vec{x}^{(2)}, y^{(2)}), \ldots, (\vec{x}^{(N)}, y^{(N)})\}$

  of unknown target function $y^{(i)} = c^*(\vec{x}^{(i)}), \forall i \in \{1, \ldots, N\}$
- N = # of training examples M = # of features = $|\vec{x}^{(i)}|$

Learner produces:

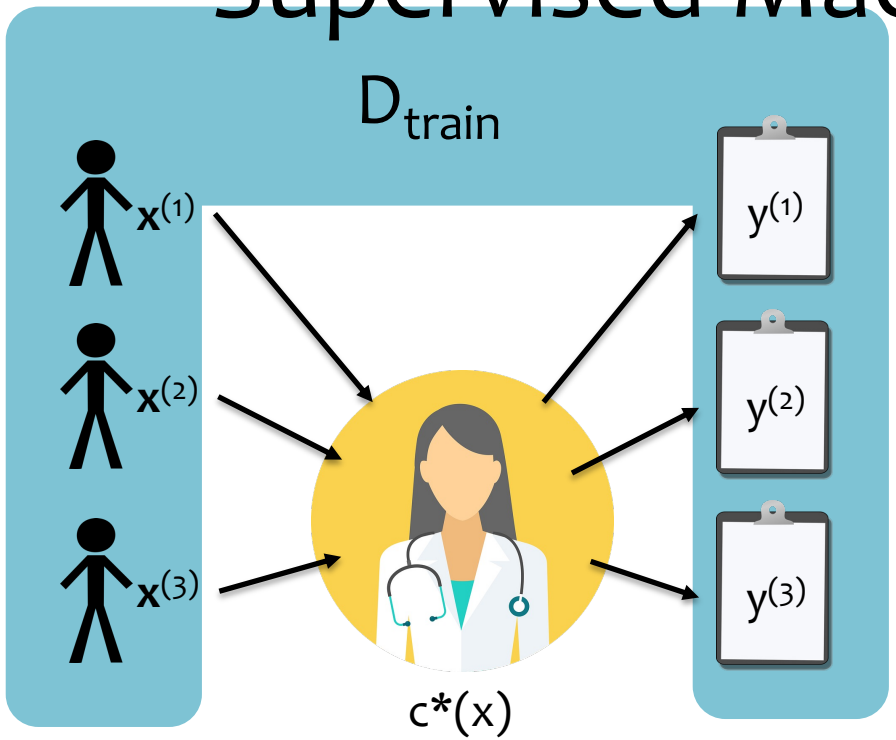
- Hypothesis $h \in \mathcal{H}$ that "best approximates" c^* according to $\mathcal{D}$

To Evaluate:

- Loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ measures how "bad" predictions $\hat{y} = h(\vec{x})$ are compared to true values $y^* = c^*(\vec{x})$

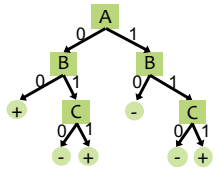
  (larger for worse pred. smaller for better pred.)
- Another dataset $\mathcal{D}_{test} = \{(\vec{x}^{(1)}, y^{(1)}), \ldots, (\vec{x}^{(N')}, y^{(N')})\}$
- Evaluate the average loss of $h(\vec{x})$ on $\mathcal{D}_{test}$

Supervised Machine Learning



Learning Algorithm

h(x)



Medical Diagnosis Dataset

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

i	y	x_1 allergic?	x_2 hives?	x_3 sneezing?	x_4 red eye?	x_5 has cat?	
1	$y^{(1)}$ -	$x_1^{(1)}$ Y	$x_2^{(1)}$ N	$x_3^{(1)}$ N	$x_4^{(1)}$ N		$\mathbf{x}^{(1)}$
2	$y^{(2)}$ -	$x_1^{(2)}$ N	$x_2^{(2)}$ Y	$x_3^{(2)}$ N	$x_4^{(2)}$ N		$\mathbf{x}^{(2)}$
3	$y^{(3)}$ +	$x_1^{(3)}$ Y	$x_2^{(3)}$ Y	$x_3^{(3)}$ N	$x_4^{(3)}$ N		$\mathbf{x}^{(3)}$
4	$y^{(4)}$ -	$x_1^{(4)}$ Y	$x_2^{(4)}$ N	$x_3^{(4)}$ Y	$x_4^{(4)}$ Y		$\mathbf{x}^{(4)}$
5	$y^{(5)}$ +	$x_1^{(5)}$ N	$x_2^{(5)}$ Y	$x_3^{(5)}$ Y	$x_4^{(5)}$ N		$\mathbf{x}^{(5)}$

c^*

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
  $D = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(N)}, y^{(N)})\}$
  where $y^{(i)} = c^*(\mathbf{x}^{(i)})$
  (history of patients and their diagnoses)
- **Learner produces** a hypothesis function, $\hat{y} = h(x)$, that best approximates unknown target function $y = c^*(x)$ on the training data

Supervised Machine Learning

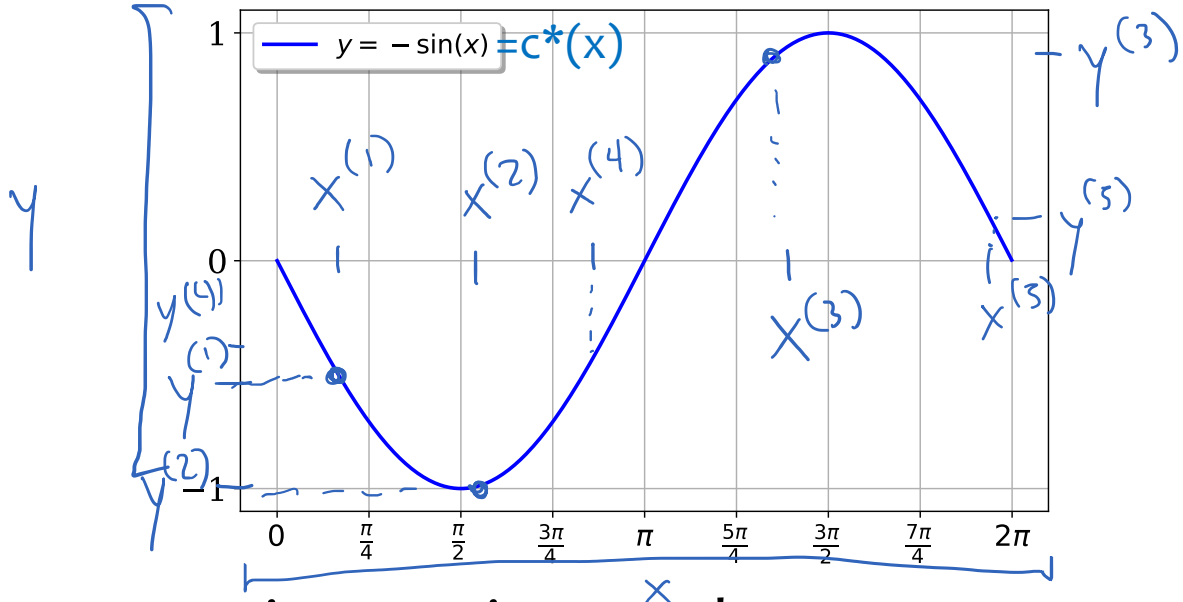
- **Problem Setting**
  - Set of possible inputs, $\mathbf{x} \in \mathcal{X}$ (all possible patients)
  - Set of possible outputs, $y \in \mathcal{Y}$ (all possible diagnoses)
  - Exists an unknown target function, $c^* : \mathcal{X} \rightarrow \mathcal{Y}$ (the doctor's brain)
  - Set, $\mathcal{H}$, of candidate hypoth (all possible decision trees)
- **Learner is given** N training D = $\{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots,$ where $y^{(i)} = c^*(\mathbf{x}^{(i)})$ (history of patients and the
- **Learner produces** a hypoth approximates unknown tar 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:

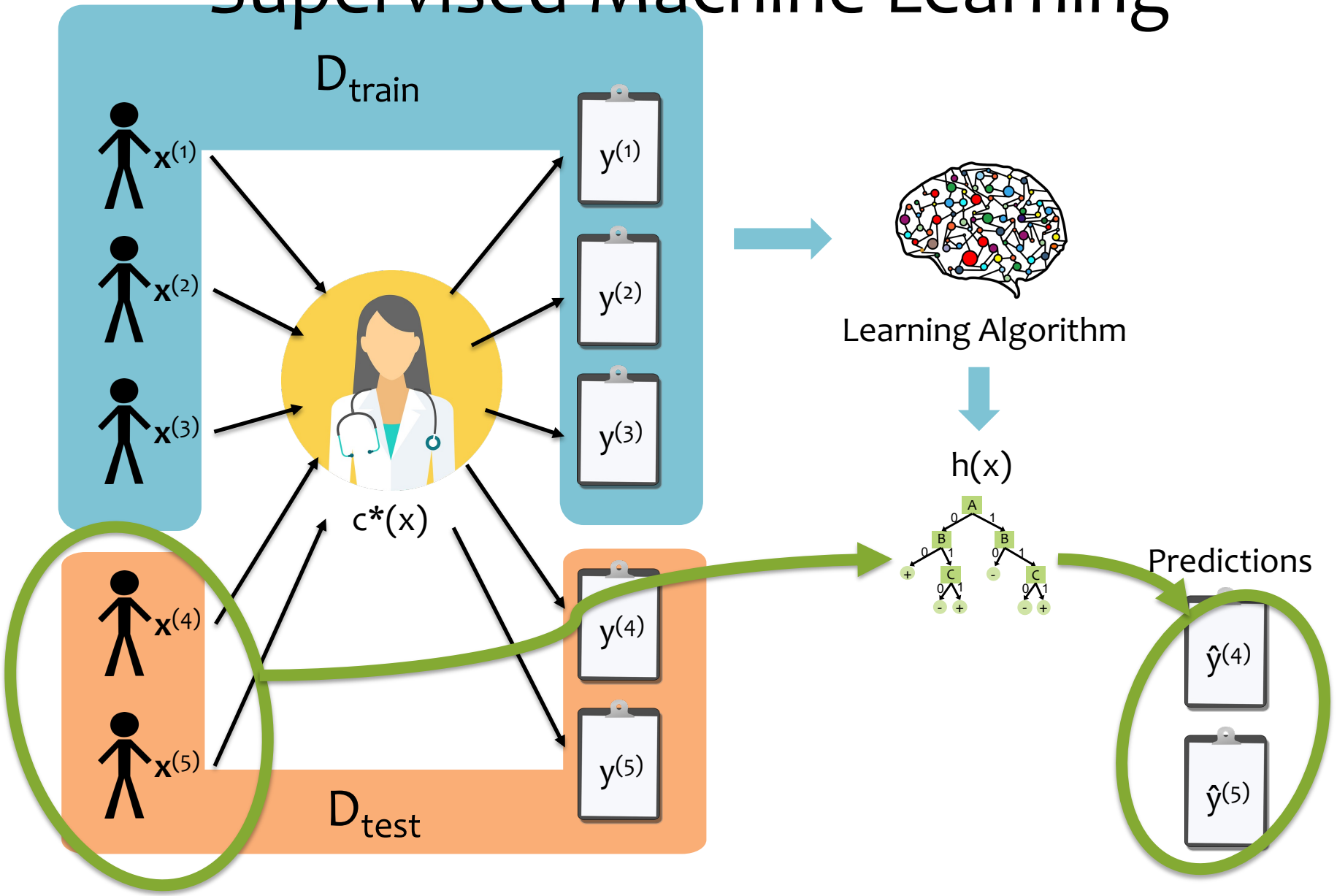
1. You can't call any other trigonometric functions
2. You *can* call an existing implementation of sin(x) a few times (e.g. 100) to test your solution
3. You only need to evaluate it for x in [0, 2*pi]

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

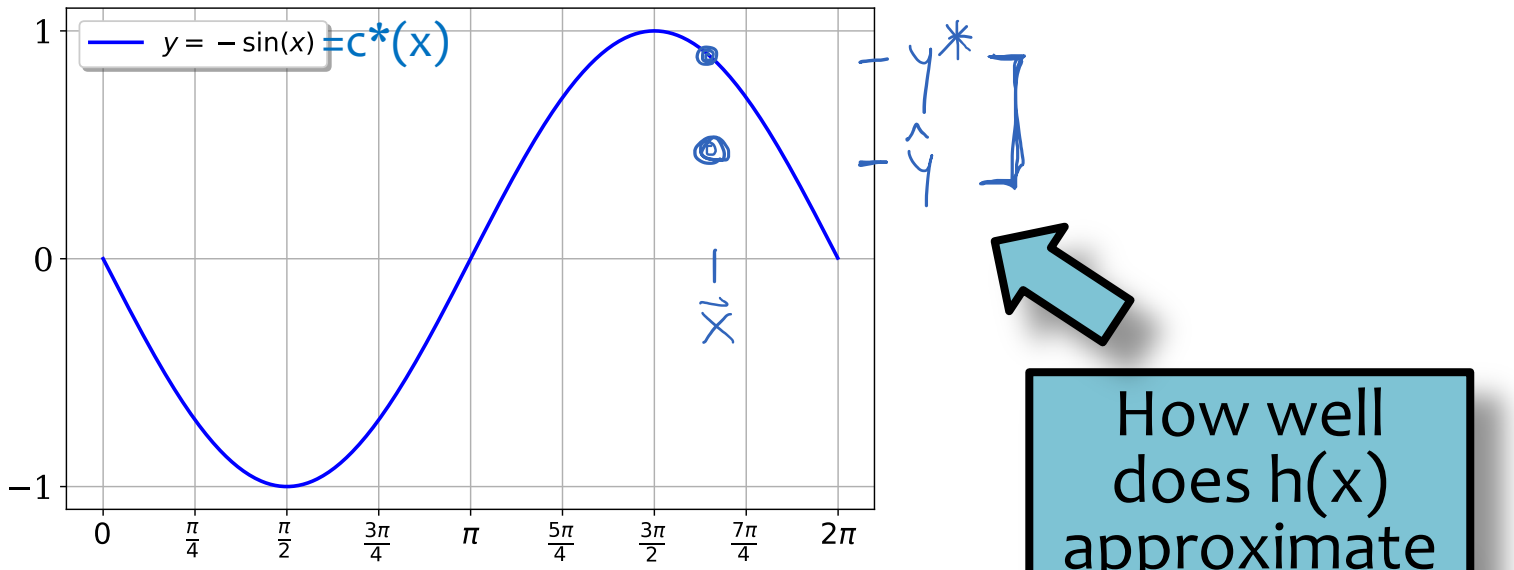
EVALUATION OF MACHINE LEARNING ALGORITHM

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*, $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$
  - Defines how "bad" predictions, $\hat{y} = h(x)$, are compared to the true labels, $y = c^*(x)$
  - Common choices:
    1. Squared loss (for regression): $\ell(y, \hat{y}) = (y - \hat{y})^2$
    2. Binary or 0-1 loss (for classification): $\ell(y, \hat{y}) = \mathbb{1}(y \neq \hat{y})$

true predicted

absolute loss

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

$= \begin{cases} 1 & \text{if } y \neq \hat{y} \\ 0 & \text{otherwise} \end{cases}$

Evaluation of ML Algorithms

- **Definition:** *loss function*, $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$
  - Defines how "bad" predictions, $\hat{y} = h(x)$, are compared to the true labels, $y = c^*(x)$
  - Common choices:
    1. Squared loss (for regression): $\ell(y, \hat{y}) = (y - \hat{y})^2$
    2. Binary or 0-1 loss (for classification): $\ell(y, \hat{y}) = \mathbb{1}(y \neq \hat{y})$

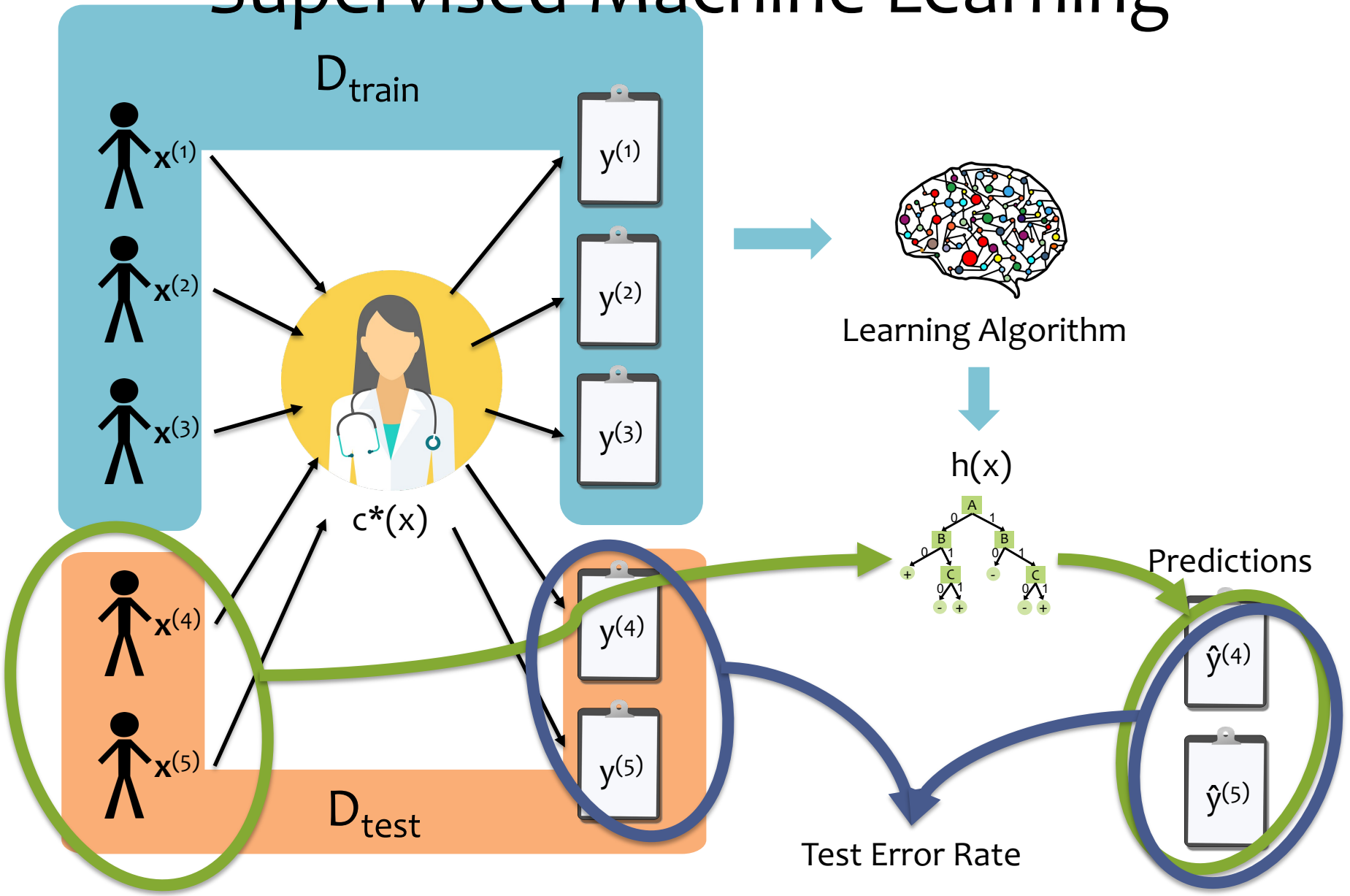
- **Definition:** the *error rate* of a hypothesis h on a dataset $\mathcal{D}$ is the average 0-1 loss:

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

- **Definition:** the *mean squared error* is the average squared loss (more on this later)

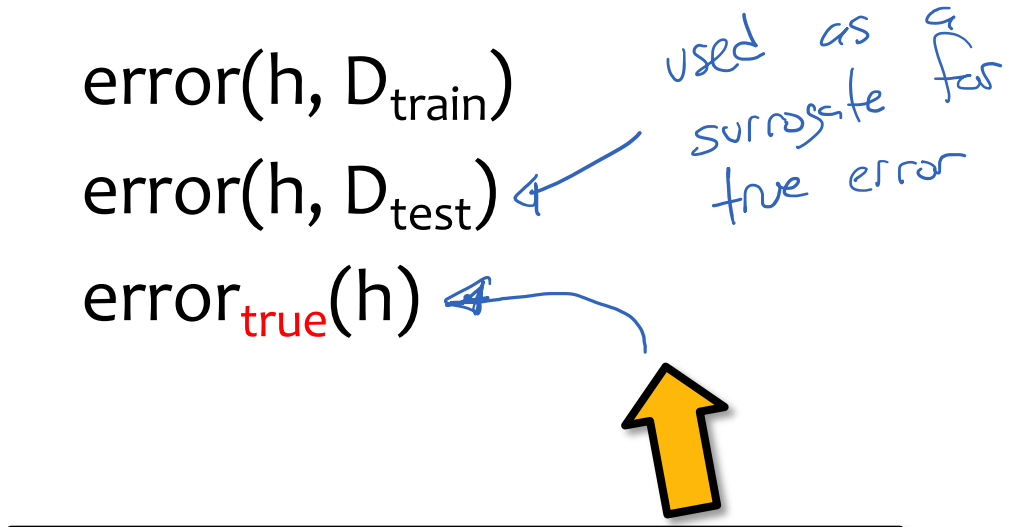
- **Q:** How do we evaluate a machine learning algorithm?
  A: Check its average loss on a separate test dataset, $\mathcal{D}_{\text{test}}$.

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: $\text{error}_{\text{true}}(h)$



Majority Vote Classifier Example

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

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, …, 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?**
-	-	Y	N	N	N
-	-	N	Y	N	N
+	+	Y	Y	N	N
-	-	Y	N	Y	Y
+	+	N	Y	Y	N

The memorizer always gets zero training error!

Algorithms for Classification

Assume $y^{(i)} = c^*(x^{(i)})$

Question:

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

Answer:

2^{100}

$\vec{X} = [X_1, X_2, \cdots, X_{100}]$, $X_M \in \{Y, N\}$

$|\mathcal{Y}| = 2^{100}$

Algorithm 1: Majority Vote

Pseudocode

```
def train(D):
    store v = majority_vote(D)
            = the class y ∈ 𝒴 that appears most often in D

def h(x⃗):
    return v

def predict(D_test):
    for (x⃗^(i), y^(i)) ∈ D_test:
        ŷ^(i) = h(x⃗^(i))
```

reuse for any classifier

Algorithm 2: Memorizer

Pseudocode

def train(D):
    store D

def $h(\vec{x})$:
    if $\exists \vec{x}^{(i)} \in D$ s.t. $\vec{x}^{(i)} = \vec{x}$
        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
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

1/5

Nonzero training error, but perhaps still better than the memorizer

Example decision stump:

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

sneezy = N

Algorithm 3: Decision Stump

Assume $x_m \in \{0, 1\}$

Pseudocode

def train(D):

1. pick an attribute, m
2. divide dataset on x_m

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

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

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 $v^{(1)}$

Algorithms for Classification

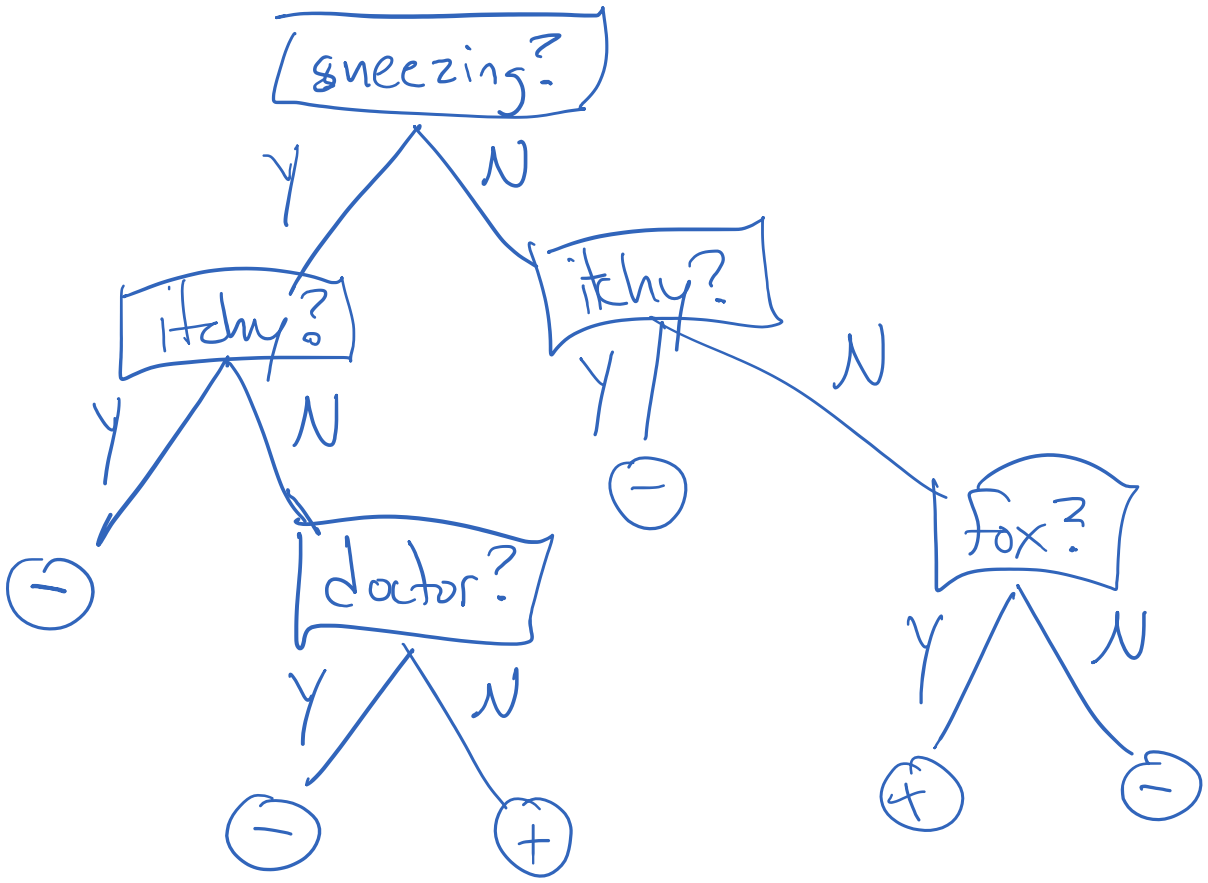
Algorithm 3 **decision stump**: based on a single feature, x_d, predict the most common label in the training dataset among all data points that have the same value for x_d

Questions:

1. How do we pick which feature to split on?
2. Why stop at one feature?

Algorithm 4: Decision Tree (preview)

Example



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+ .
| | | | Birth_Weight >= 3349: [133+,36.4-] .78+
| | | 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-
```

Figure from Tom Mitchell