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
Jan. 14, 2026

Reminders

- **Homework 1: Background**
  - **Out: Mon, Jan 12**
  - **Due: Wed, Jan 21 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. for Slot A submissions (human only – no AI use!), we will grant any reasonable extension requests (helps with late adds), but you must request one; and it must be in *before* we finish grading

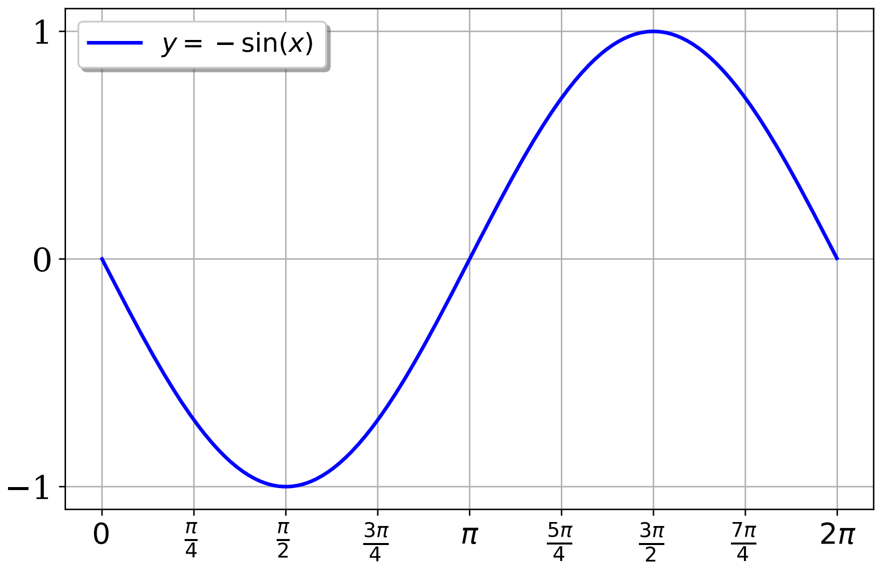
Big Ideas

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).



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

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

As a (supervised) binary classification task

labels: allergic? — features: hives?, sneezing?, red eye?, has cat? — examples: rows 1–5

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 (allergic?) · features (hives?, sneezing?, red eye?, has cat?) · examples (i = 1–5)

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

labels features

examples

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: treatment cost; features: hives?, sneezing?, red eye?, has cat?; examples: i = 1–5

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: allergic?; features: hives?, sneezing?, red eye?, has cat?; rows: 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 $x_1, x_1, \ldots, x_M$

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

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 $x_1, x_1, \ldots, x_M$

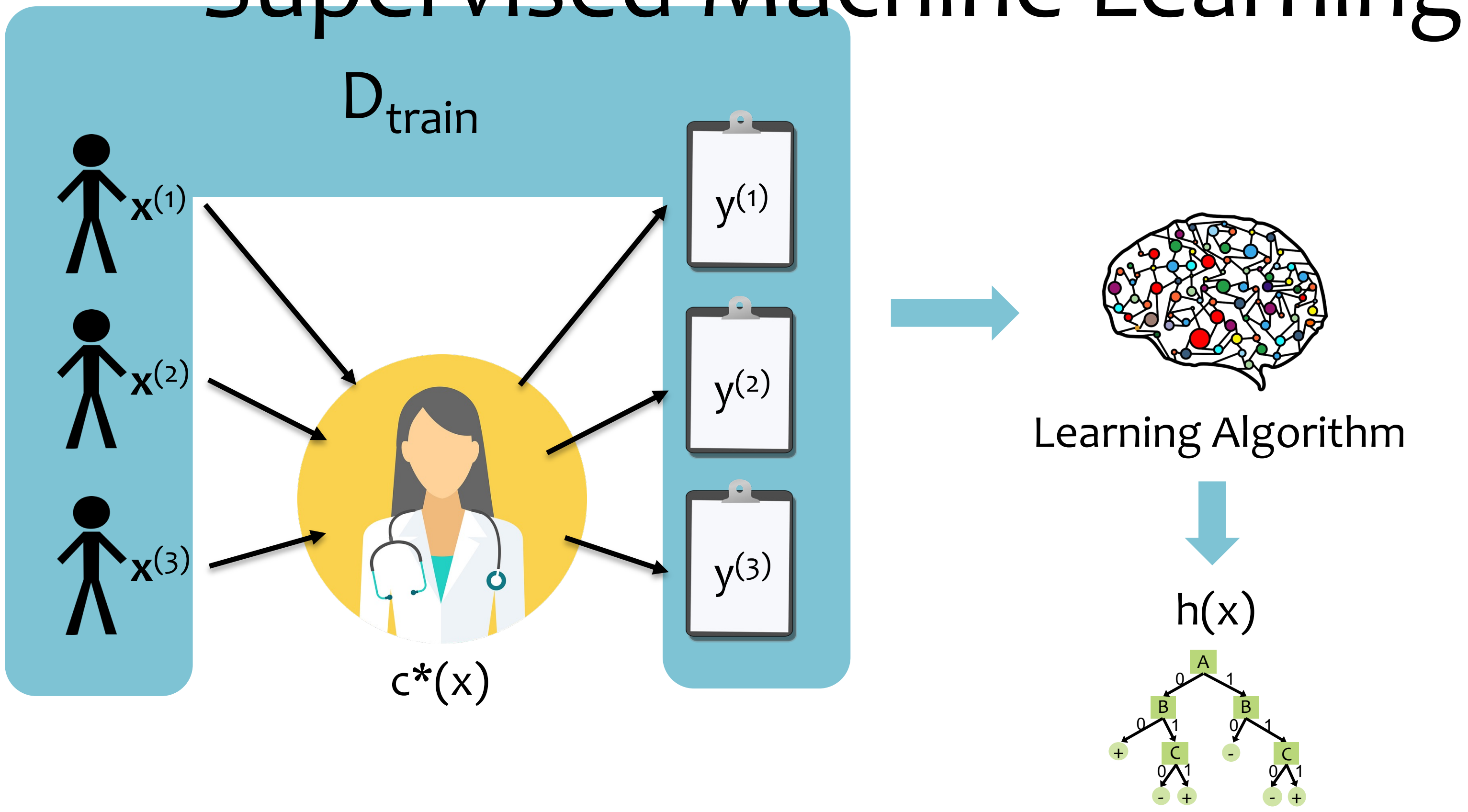
	y	x_1	x_2	x_3	x_4	
i	**allergic?**	**hives?**	**sneezing?**	**red eye?**	**has cat?**	
1	$y^{(1)}$ -	$x_1^{(1)}$ Y	$x_2^{(1)}$ N	$x_3^{(1)}$ N	$x_4^{(1)}$ N	$\mathbf{x}^{(1)}$
2	$y^{(2)}$ -	$x_1^{(2)}$ N	$x_2^{(2)}$ Y	$x_3^{(2)}$ N	$x_4^{(2)}$ N	$\mathbf{x}^{(2)}$
3	$y^{(3)}$ +	$x_1^{(3)}$ Y	$x_2^{(3)}$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

Supervised Machine Learning



Medical Diagnosis Dataset

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

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

N = 5 training examples

M = 4 attributes

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

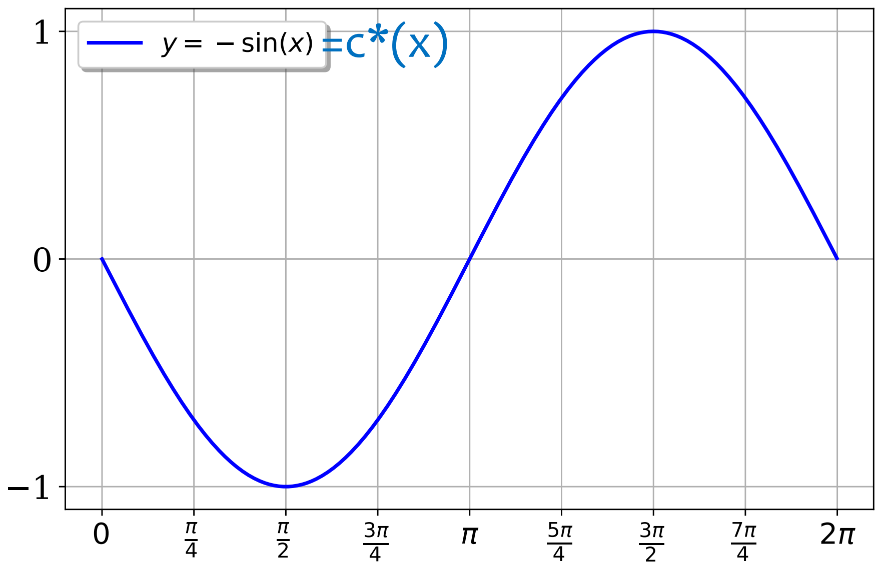
- **Problem Setting**
  - Set of possible inputs, $\mathbf{x} \in \mathcal{X}$ (all possible patients)
  - Set of possible outputs, $y \in \mathcal{Y}$ (all possible diagnoses)
  - Exists an unknown tar[...] function, $c^* : \mathcal{X} \rightarrow \mathcal{Y}$ (the doctor's brain)
  - Set, $\mathcal{H}$, 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:

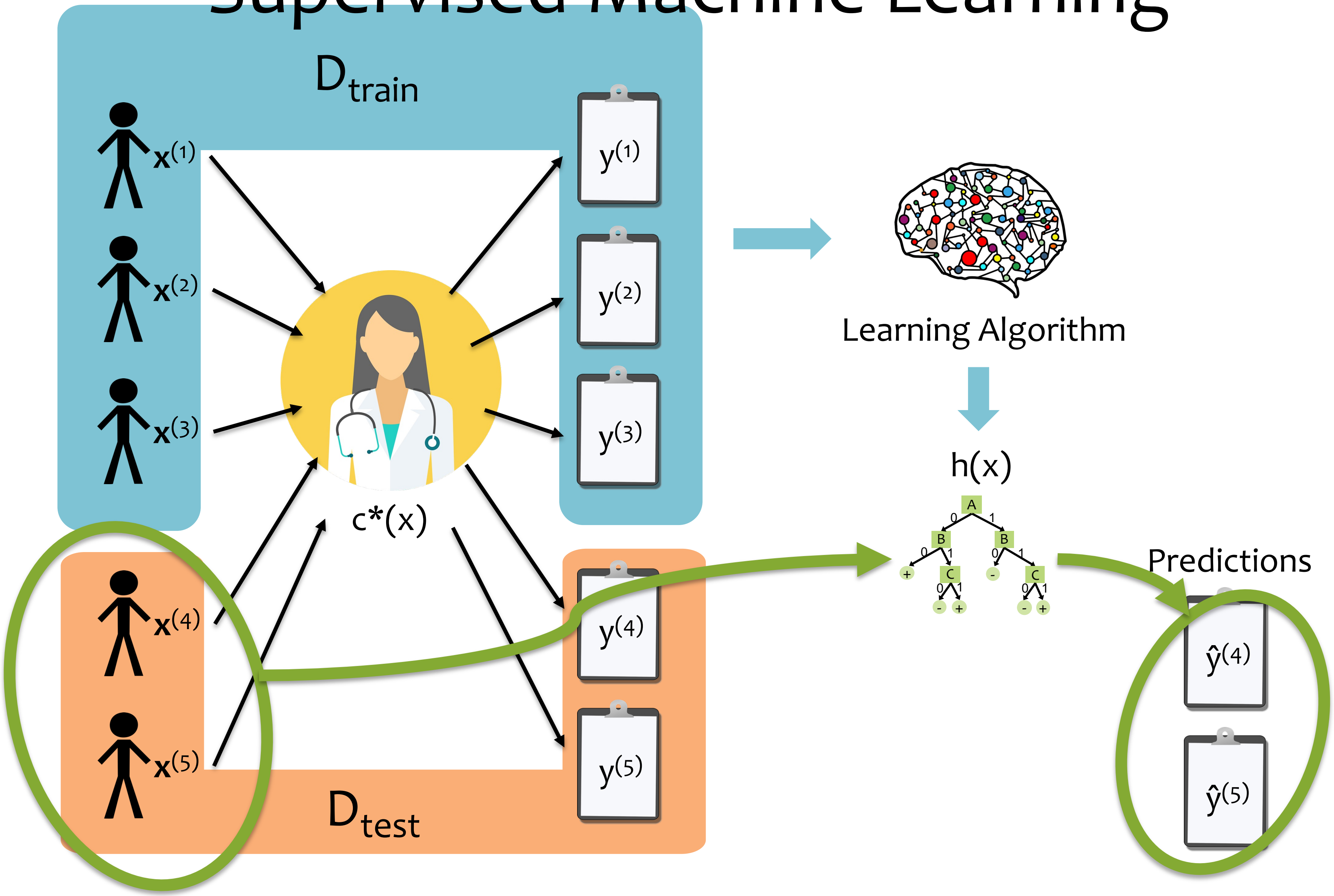
1. You can't call any other trigonometric functions
2. You *can* call an existing implementation of sin(x) a few times (e.g. 100) to test your solution
3. You only need to evaluate it for x in [0, 2*pi]

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

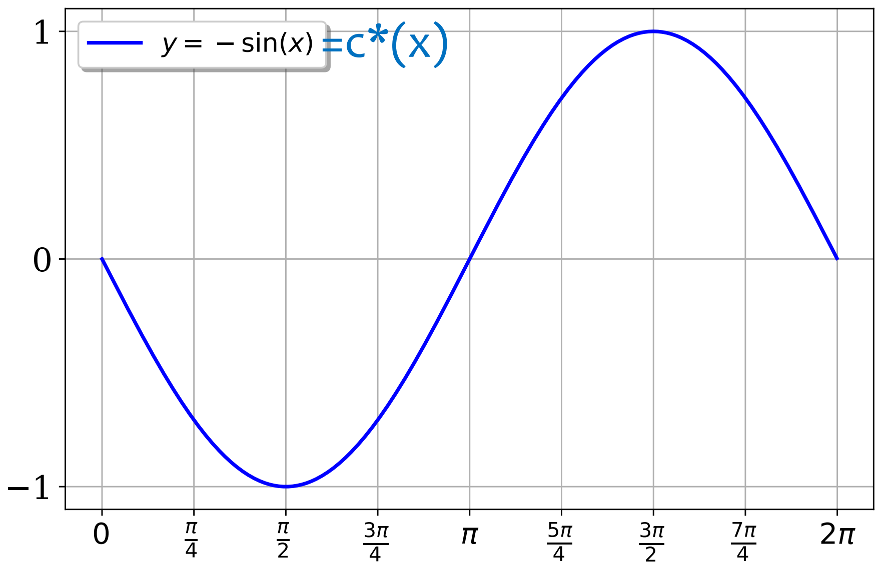
EVALUATION OF MACHINE LEARNING ALGORITHM

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*, $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$
  - 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): $\ell(y, \hat{y}) = \mathbb{1}(y \neq \hat{y}) = \begin{cases} 1, & \text{if } y \neq \hat{y} \\ 0, & \text{otherwise} \end{cases}$

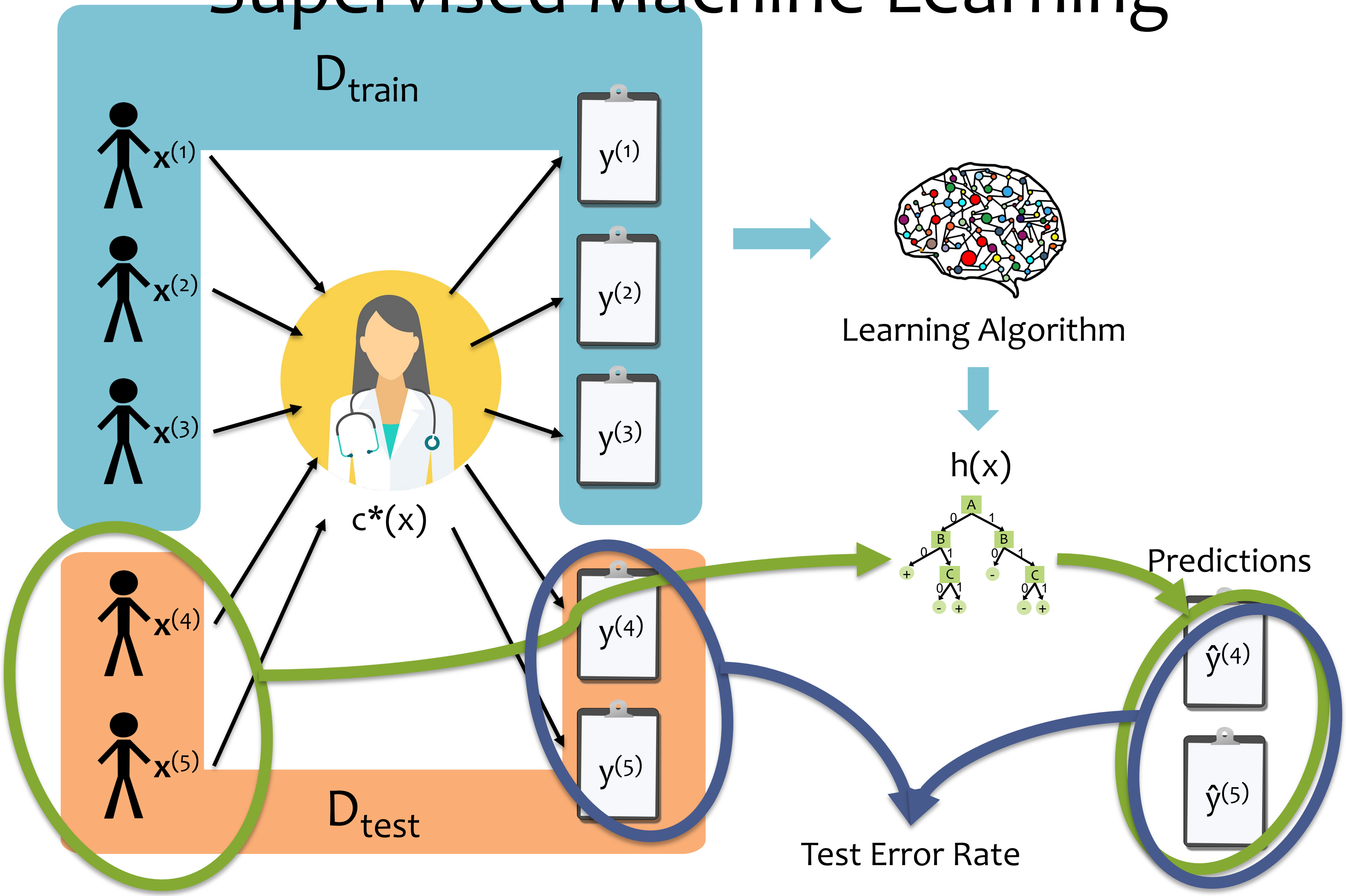
Evaluation of ML Algorithms

- **Definition:** *loss function*, $\ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$
  - 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): $\ell(y, \hat{y}) = \mathbb{1}(y \neq \hat{y}) = \begin{cases} 1, & \text{if } y \neq \hat{y} \\ 0, & \text{otherwise} \end{cases}$
- **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: $error(h, D_{train})$

  …error rate over all test data: $error(h, D_{test})$

  …true error over all data: $error_{true}(h)$

So we'll use $error(h, D_{test})$ as a surrogate for $error_{true}(h)$ in practice

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

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}

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

Could this dataset have come from our "problem setting" defined earlier?

Why or why not?

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

Question:

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

Answer:

Algorithm 1: Majority Vote

Pseudocode

Algorithm 2: Memorizer

Pseudocode

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

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

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