



# 10-601 Introduction to Machine Learning

Machine Learning Department  
School of Computer Science  
Carnegie Mellon University

## k-Nearest Neighbors

Matt Gormley  
Lecture 4  
Feb. 10, 2021

# Course Staff



Everett Knag



Amanda Coston



Catherine Cheng



Eric Liang



Daniel Min



Varun Natu



Evan Feder



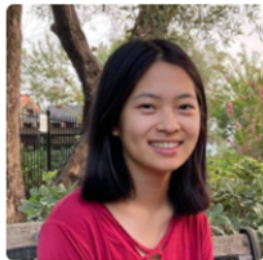
Alex Singh



Rebecca Yang



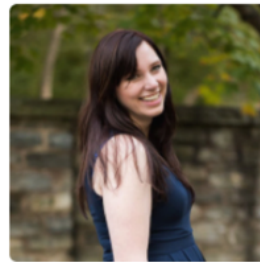
Young Kim



Vivian Cheng



Clay Yoo



Brynn Edmunds



Fatima Kizilkaya



Tom Mitchell



Matt Gormley

# Course Staff



Everett Knag



Amanda Coston



Catherine Cheng



Eric Liang



Daniel Min



Varun Natu



Evan Feder



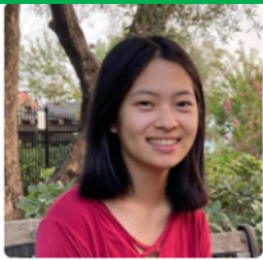
Alex Singh



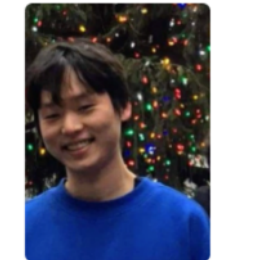
Rebecca Yang



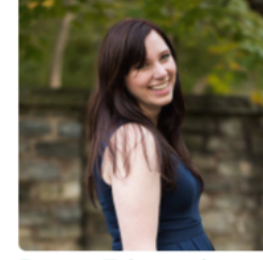
Young Kim



Vivian Cheng



Clay Yoo



Brynn Edmunds



Fatima Kizilkaya



Tom Mitchell



Matt Gormley

Team A

# Course Staff



Everett Knag



Amanda Coston



Catherine Cheng



Eric Liang



Daniel Min



Varun Natu



Evan Feder



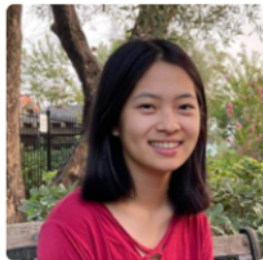
Alex Singh



Rebecca Yang



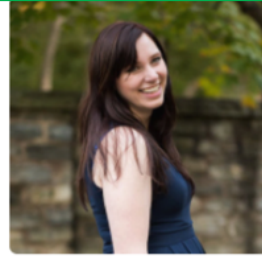
Young Kim



Vivian Cheng



Clay Yoo



Brynn Edmunds



Fatima Kizilkaya



Tom Mitchell



Matt Gormley

Team B

# Course Staff



Everett Knag



Amanda Coston



Catherine Cheng



Eric Liang



Daniel Min



Varun Natu



Evan Feder



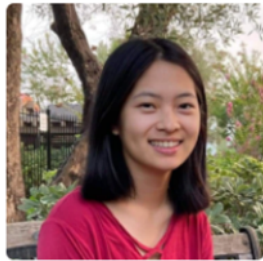
Alex Singh



Rebecca Yang



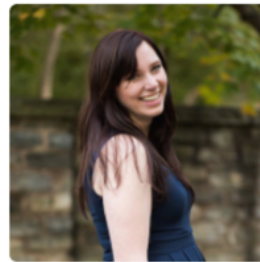
Young Kim



Vivian Cheng



Clay Yoo



Brynn Edmunds



Fatima Kizilkaya



Tom Mitchell



Matt Gormley

Team C

# Course Staff



Everett Knag



Amanda Coston



Catherine Cheng



Eric Liang



Daniel Min



Varun Natu



Evan Feder



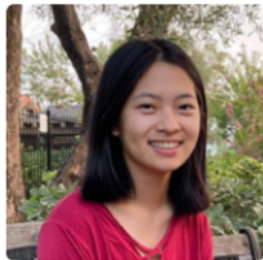
Alex Singh



Rebecca Yang



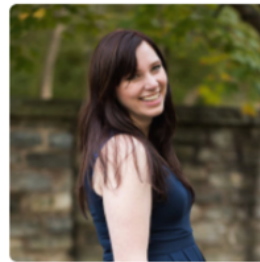
Young Kim



Vivian Cheng



Clay Yoo



Brynn Edmunds



Fatima Kizilkaya



Tom Mitchell



Matt Gormley

Team D

# Course Staff



Everett Knag



Amanda Coston



Catherine Cheng



Eric Liang



Daniel Min



Varun Natu



Evan Feder



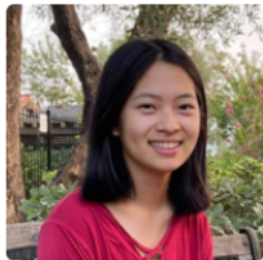
Alex Singh



Rebecca Yang



Young Kim



Vivian Cheng



Clay Yoo



Brynn Edmunds



Fatima Kizilkaya



Tom Mitchell



Matt Gormley

EAs

# Course Staff



Everett Knag



Amanda Coston



Catherine Cheng



Eric Liang



Daniel Min



Varun Natu



Evan Feder



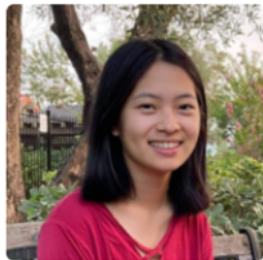
Alex Singh



Rebecca Yang



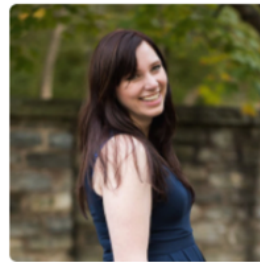
Young Kim



Vivian Cheng



Clay Yoo



Brynn Edmunds



Fatima Kizilkaya



Tom Mitchell



Matt Gormley



# Reminders

- **Homework 1: Background**
  - **Out: Wed, Feb 03 (2nd lecture)**
  - **Due: Wed, Feb 10 at 11:59pm**
  - **unique policy for this assignment: we will grant (essentially) any and all extension requests**
- **Homework 2: Decision Trees**
  - **Out: Wed, Feb. 10**
  - **Due: Mon, Feb. 22 at 11:59pm**

# First “Toxic Option” Poll

## Question:

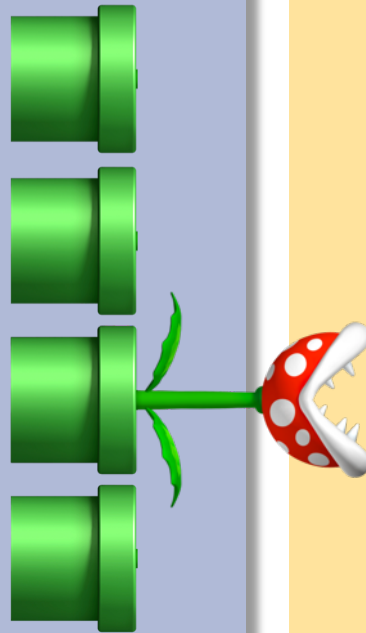
*How are you participating in class today?*

A. Laptop

B. Smart phone

C. Pay phone

D. Desktop



## Answer:

# **OVERFITTING (RECAP FOR DECISION TREES)**

# Decision Tree Generalization

## Question:

*Which of the following would generalize best to unseen examples?*

- A. Small tree with low training accuracy
- B. Large tree with low training accuracy
- C. Small tree with high training accuracy
- D. Large tree with high training accuracy

## Answer:



# DT: Remarks

ID3 = Decision Tree  
Learning with Mutual  
Information as the  
splitting criterion

**Question:** Which tree does ID3 find?

## **Definition:**

We say that the **inductive bias** of a machine learning algorithm is the principal by which it generalizes to unseen examples

## **Inductive Bias of ID3:**

Smallest tree that matches the data with high mutual information attributes near the top

## **Occam's Razor: (restated for ML)**

Prefer the simplest hypothesis that explains the data

# Overfitting and Underfitting

## Underfitting

- The model...
  - is too simple
  - is unable captures the trends in the data
  - exhibits too much bias
- *Example:* majority-vote classifier (i.e. depth-zero decision tree)
- *Example:* a toddler (that has **not** attended medical school) attempting to carry out medical diagnosis

## Overfitting

- The model...
  - is too complex
  - is fitting the noise in the data
  - or fitting random statistical fluctuations inherent in the “sample” of training data
  - does not have enough bias
- *Example:* our “memorizer” algorithm responding to an “orange shirt” attribute
- *Example:* medical student who simply memorizes patient case studies, but does not understand how to apply knowledge to new patients

# Overfitting

- Consider a hypothesis  $h$  its...
  - ... error rate over all training data:  $\text{error}(h, D_{\text{train}})$
  - ... error rate over all test data:  $\text{error}(h, D_{\text{test}})$

# Overfitting

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

- We say  $h$  overfits the training data if...

$$\text{error}_{\text{true}}(h) > \text{error}(h, D_{\text{train}})$$

- Amount of overfitting =

$$\text{error}_{\text{true}}(h) - \text{error}(h, D_{\text{train}})$$



In practice,  
 $\text{error}_{\text{true}}(h)$  is  
unknown



# How to Avoid Overfitting?

## For Decision Trees...

1. Do not grow tree beyond some **maximum depth**
2. Do not split if splitting criterion (e.g. mutual information) is **below some threshold**
3. Stop growing when the split is **not statistically significant**
4. Grow the entire tree, then **prune**

# DT Learning Objectives

*You should be able to...*

1. Implement Decision Tree training and prediction
2. Use effective splitting criteria for Decision Trees and be able to define entropy, conditional entropy, and mutual information / information gain
3. Explain the difference between memorization and generalization [CIML]
4. Describe the inductive bias of a decision tree
5. Formalize a learning problem by identifying the input space, output space, hypothesis space, and target function
6. Explain the difference between true error and training error
7. Judge whether a decision tree is "underfitting" or "overfitting"
8. Implement a pruning or early stopping method to combat overfitting in Decision Tree learning

# **K-NEAREST NEIGHBORS**

# Fisher Iris Dataset

Fisher (1936) used 150 measurements of flowers from 3 different species: Iris setosa (0), Iris virginica (1), Iris versicolor (2) collected by Anderson (1936)

Species	Sepal Length	Sepal Width	Petal Length	Petal Width
0	4.3	3.0	1.1	0.1
0	4.9	3.6	1.4	0.1
0	5.3	3.7	1.5	0.2
1	4.9	2.4	3.3	1.0
1	5.7	2.8	4.1	1.3
1	6.3	3.3	4.7	1.6
1	6.7	3.0	5.0	1.7

# Fisher Iris Dataset

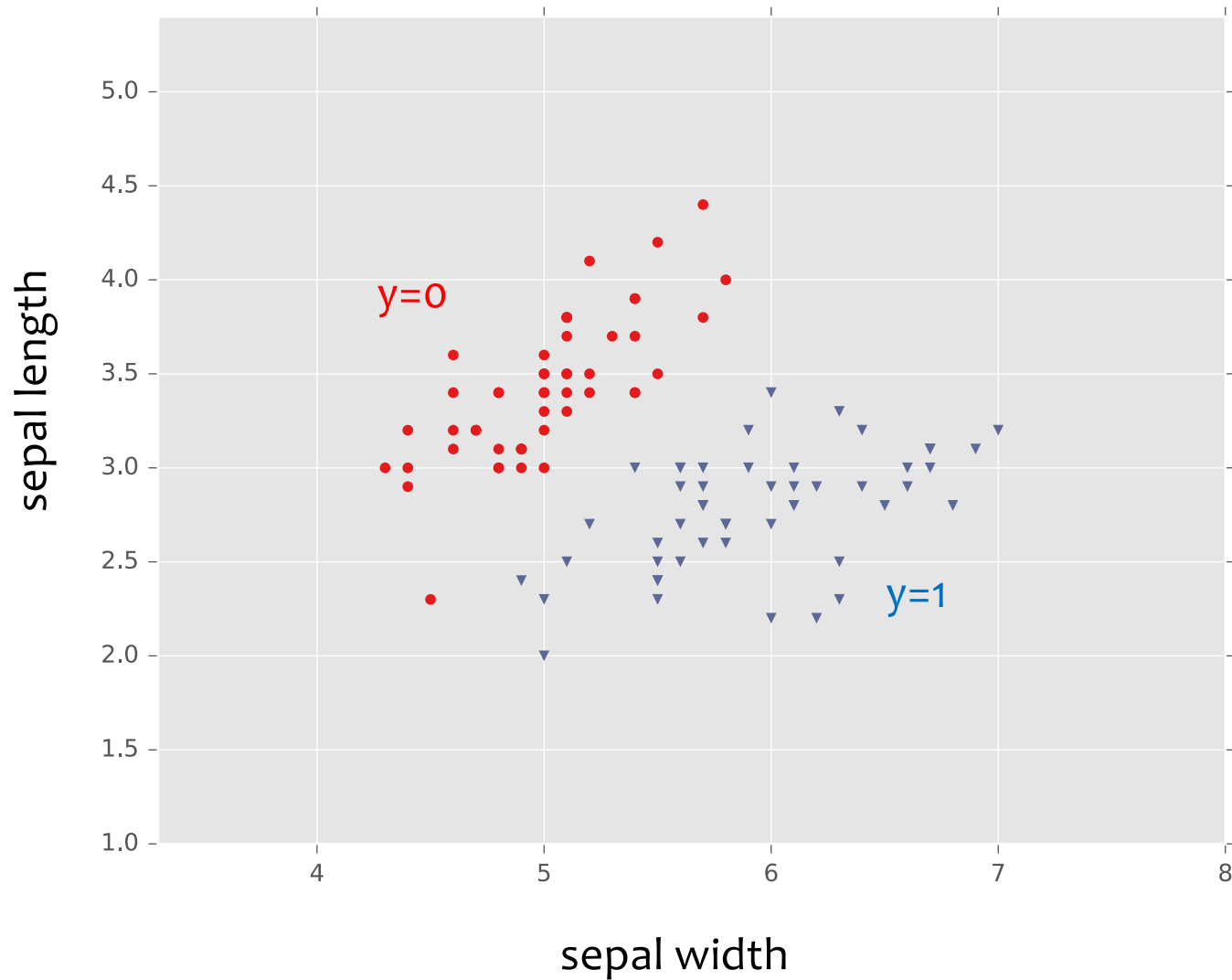
Fisher (1936) used 150 measurements of flowers from 3 different species: Iris setosa (0), Iris virginica (1), Iris versicolor (2) collected by Anderson (1936)

Species	Sepal Length	Sepal Width
0	4.3	3.0
0	4.9	3.6
0	5.3	3.7
1	4.9	2.4
1	5.7	2.8
1	6.3	3.3
1	6.7	3.0

Deleted two of the four features, so that input space is 2D



# Fisher Iris Dataset





# Classification

## *Whiteboard:*

- Binary classification
- 2D examples
- Decision rules / hypotheses



# k-Nearest Neighbors

*Whiteboard:*

- Nearest Neighbor classifier
- KNN for binary classification

# KNN: Remarks

## Distance Functions:

- KNN requires a **distance function**

$$g : \mathbb{R}^M \times \mathbb{R}^M \rightarrow \mathbb{R}$$

- The most common choice is **Euclidean distance**

$$g(\mathbf{u}, \mathbf{v}) = \sqrt{\sum_{m=1}^M (u_m - v_m)^2}$$

- But other choices are just fine (e.g. **Manhattan distance**)

$$g(\mathbf{u}, \mathbf{v}) = \sum_{m=1}^M |u_m - v_m|$$

# KNN: Remarks

## In-Class Exercises

1. How can we handle ties for even values of  $k$ ?
2. What is the inductive bias of KNN?

## Answer(s) Here:

# KNN: Remarks

## In-Class Exercises

1. How can we handle ties for even values of  $k$ ?
2. What is the inductive bias of KNN?

## Answer(s) Here:

1)

- Consider another point
- Remove farthest of  $k$  points
- Weight votes by distance
- Consider another distance metric

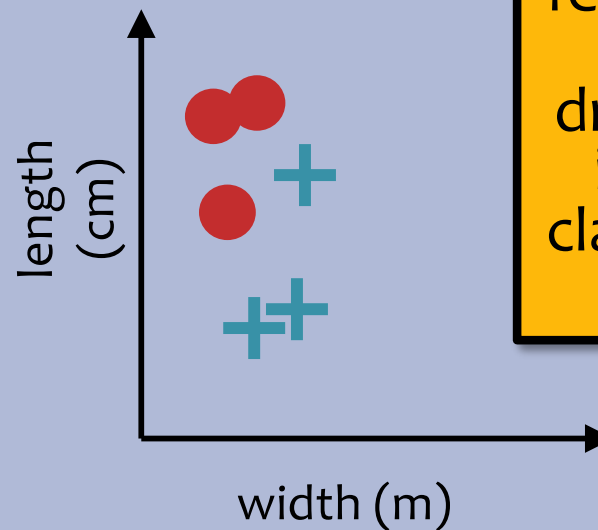
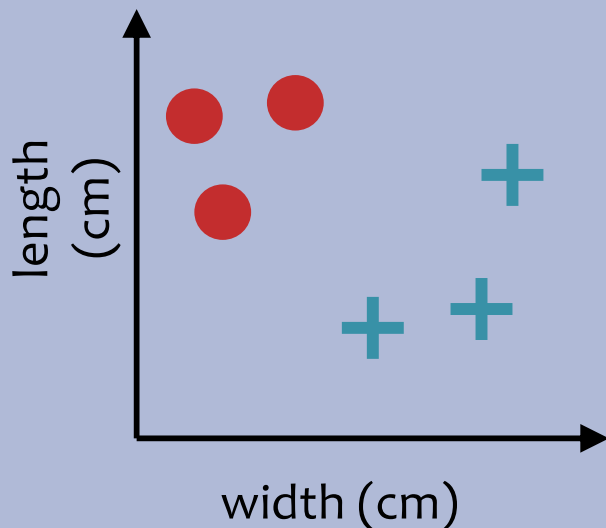
2)

# KNN: Remarks

## Inductive Bias:

1. Similar points should have similar labels
2. All dimensions are created equally!

Example: two features for KNN



**big problem:**  
feature scale  
could  
dramatically  
influence  
classification  
results

# KNN: Remarks

## Computational Efficiency:

- Suppose we have  $N$  training examples, and each one has  $M$  features
- Computational complexity for the special case where  $k=1$ :

Task	Naive	k-d Tree
Train	$O(1)$	$\sim O(M N \log N)$
Predict (one test example)	$O(MN)$	$\sim O(2^M \log N)$ on average



**Problem:** Very fast for small  $M$ , but very slow for large  $M$

**In practice:** use stochastic approximations (very fast, and empirically often as good)

# KNN: Remarks


## Theoretical Guarantees:

### Cover & Hart (1967)

Let  $h(x)$  be a Nearest Neighbor ( $k=1$ ) binary classifier. As the number of training examples  $N$  goes to infinity...

$$\text{error}_{\text{true}}(h) < 2 \times \text{Bayes Error Rate}$$

“In this sense, it may be said that half the classification information in an infinite sample set is contained in the nearest neighbor.”



**very informally,**  
Bayes Error Rate can be thought of as:  
*‘the best you could possibly do’*

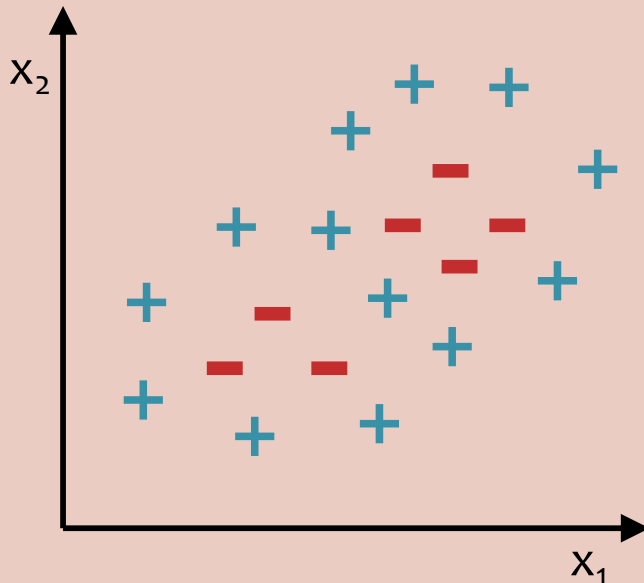
# Decision Boundary Example

**Dataset:** Outputs  $\{+, -\}$ ; Features  $x_1$  and  $x_2$

## In-Class Exercise

Question:

- Can a **k-Nearest Neighbor classifier with  $k=1$**  achieve **zero training error** on this dataset?
- If **'Yes'**, draw the learned decision boundary. If **'No'**, why not?



Question:

- Can a **Decision Tree classifier** achieve **zero training error** on this dataset?
- If **'Yes'**, draw the learned decision boundary. If **'No'**, why not?

