As you login

1. Rename yourself in Zoom to *pre*-pend your house number
   - e.g. “0 – Pat Virtue”

2. Open Piazza (getting ready for polls)

3. Download preview slides from course website

4. Grab something to write with/on ☺

Announcements

Assignments

- HW1 Feedback
- HW2
  - Due Mon, 9/21, 11:59 pm
  - Mostly programming
  - Written component in LaTeX
  - Don’t delay on this on. OH will be *super* crowded as the deadline gets closer

Feeling behind already?

- Ask for help now

Introduction to Machine Learning

Decision Trees and Nearest Neighbor

Instructor: Pat Virtue

Plan

Last time

- Decision trees
    - Recursive algorithm
    - Better splitting criteria (entropy, mutual information)

Today

- Decision trees
    - Continuous features
    - Overfitting
- Nearest neighbor methods

Next time

- More nearest neighbor and model selection

Building a Decision Tree

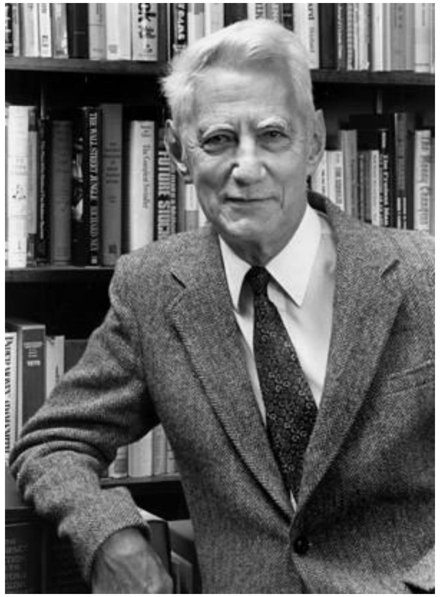
```
Function BuildTree(D,A)
    # D: dataset at current node, A: current set of attributes
    If empty(A) or all labels in D are the same
        # Leaf node
        class = most common class in D
    else
        # Internal node
        a ⇐ bestAttribute(D,A)
        LeftNode = BuildTree(D(a=1), A \ {a})
        RightNode = BuildTree(D(a=0), A \ {a})
    end
end
```

Entropy

- Quantifies the amount of uncertainty associated with a specific probability distribution
- The higher the entropy, the less confident we are in the outcome
- Definition

$$H(X) = \sum_x p(X = x) \log_2 \frac{1}{p(X = x)}$$

$$H(X) = -\sum_x p(X = x) \log_2 p(X = x)$$



Claude Shannon (1916 – 2001), most of the work was done in Bell labs

Mutual Information

Let X be a random variable with $X \in \mathcal{X}$.
Let Y be a random variable with $Y \in \mathcal{Y}$.

Entropy: $H(Y) = -\sum_{y \in \mathcal{Y}} P(Y = y) \log_2 P(Y = y)$

Specific Conditional Entropy: $H(Y \mid X = x) = -\sum_{y \in \mathcal{Y}} P(Y = y \mid X = x) \log_2 P(Y = y \mid X = x)$

Conditional Entropy: $H(Y \mid X) = \sum_{x \in \mathcal{X}} P(X = x) H(Y \mid X = x)$

Mutual Information: $I(Y; X) = H(Y) - H(Y|X)$

Slide credit: CMU MLD Matt Gormley

Mutual Information

Let X be a random variable with $X \in \mathcal{X}$.
Let Y be a random variable with $Y \in \mathcal{Y}$.

Entropy: $H(Y) = -\sum_{y \in \mathcal{Y}} P(Y=y) \log_2 P(Y=y)$

Specific Conditional Entropy: $H(Y \mid X=x) = -\sum_{y \in \mathcal{Y}} P(Y=y \mid X=x) \log_2 P(Y=y \mid X=x)$

Conditional Entropy: $H(Y \mid X) = \sum_{x \in \mathcal{X}} P(X=x) H(Y \mid X=x)$

Mutual Information: $I(Y;X) = H(Y) - H(Y|X)$

- For a decision tree, we can use **mutual information** of the output class *Y* and some attribute *X* on which to split **as a splitting criterion**
- Given a dataset *D* of training examples, we can estimate the required probabilities as…

$$P(Y=y) = N_{Y=y}/N$$
$$P(X=x) = N_{X=x}/N$$
$$P(Y=y|X=x) = N_{Y=y,X=x}/N_{X=x}$$

where $N_{Y=y}$ is the number of examples for which $Y=y$ and so on.

Slide credit: CMU MLD Matt Gormley

Mutual Information

Let X be a random variable with $X \in \mathcal{X}$.
Let Y be a random variable with $Y \in \mathcal{Y}$.

Entropy: $H(Y) = -\sum_{y \in \mathcal{Y}} P(Y = y) \log_2 P(Y = y)$

Specific Conditional Entropy: $H(Y \mid X = x) = -\sum_{y \in \mathcal{Y}} P(Y = y \mid X = x) \log_2 P(Y = y \mid X = x)$

Conditional Entropy: $H(Y \mid X) = \sum_{x \in \mathcal{X}} P(X = x) H(Y \mid X = x)$

Mutual Information: $I(Y; X) = H(Y) - H(Y|X)$

- **Entropy** measures the **expected # of bits** to code one random draw from X.
- For a decision tree, we want to **reduce the entropy of the random variable we are trying to predict**!

Conditional entropy is the expected value of specific conditional entropy $E_{P(X=x)}[H(Y \mid X = x)]$

Informally, we say that **mutual information** is a measure of the following:
If we know X, how much does this reduce our uncertainty about Y?

Slide credit: CMU MLD Matt Gormley

Splitting with Mutual Information

Which attribute {A, B} would **mutual information** select for the next split?

1) A
2) B
3) A or B (tie)
4) I don't know

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

Decision Tree Learning Example

Entropy: $H(Y) = -\sum_{y \in \mathcal{Y}} P(Y = y) \log_2 P(Y = y)$

Specific Conditional Entropy: $H(Y \mid X = x) = -\sum_{y \in \mathcal{Y}} P(Y = y \mid X = x) \log_2 P(Y = y \mid X = x)$

Conditional Entropy: $H(Y \mid X) = \sum_{x \in \mathcal{X}} P(X = x) H(Y \mid X = x)$

Mutual Information: $I(Y; X) = H(Y) - H(Y|X)$

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

Slide credit: CMU MLD Matt Gormley

Decision Tree Learning Example

Entropy: $H(Y) = -\sum_{y \in \mathcal{Y}} P(Y = y) \log_2 P(Y = y)$

Specific Conditional Entropy: $H(Y \mid X = x) = -\sum_{y \in \mathcal{Y}} P(Y = y \mid X = x) \log_2 P(Y = y \mid X = x)$

Conditional Entropy: $H(Y \mid X) = \sum_{x \in \mathcal{X}} P(X = x) H(Y \mid X = x)$

Mutual Information: $I(Y; X) = H(Y) - H(Y|X)$

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

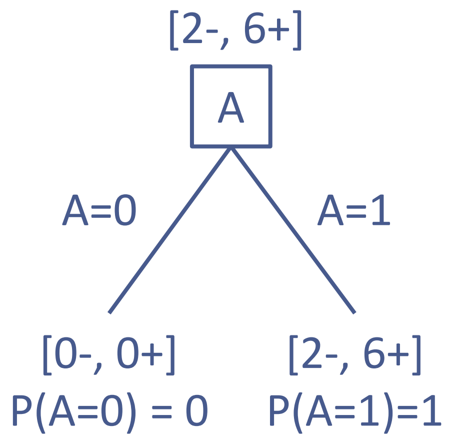
$$H(Y) = -\left[\frac{2}{8}\log_2\frac{2}{8} + \frac{6}{8}\log_2\frac{6}{8}\right]$$

$$H(Y \mid A = 0) = undefined$$

$$H(Y \mid A = 1) = -\left[\frac{2}{8}\log_2\frac{2}{8} + \frac{6}{8}\log_2\frac{6}{8}\right] = H(Y)$$

$$\begin{aligned} H(Y \mid A) &= P(A = 0)H(Y \mid A = 0) + P(A = 1)H(Y \mid A = 1) \\ &= 0 + H(Y \mid A = 1) \\ &= H(Y) \end{aligned}$$

$$I(Y; A) = H(Y) - H(Y \mid A) = 0$$



Slide credit: CMU MLD Matt Gormley

Decision Tree Learning Example

Entropy: $H(Y) = -\sum_{y \in \mathcal{Y}} P(Y=y) \log_2 P(Y=y)$

Specific Conditional Entropy: $H(Y \mid X=x) = -\sum_{y \in \mathcal{Y}} P(Y=y \mid X=x) \log_2 P(Y=y \mid X=x)$

Conditional Entropy: $H(Y \mid X) = \sum_{x \in \mathcal{X}} P(X=x) H(Y \mid X=x)$

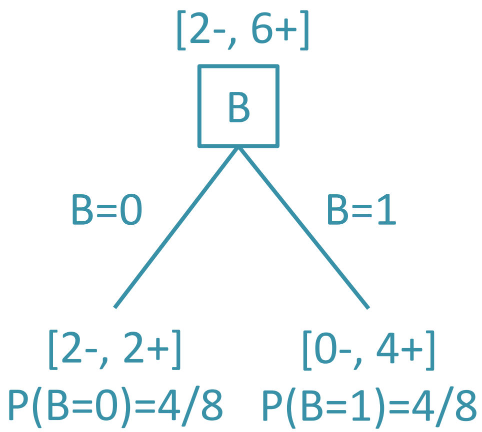
Mutual Information: $I(Y;X) = H(Y) - H(Y|X)$

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

$$H(Y) = -\left[\frac{2}{8}\log_2\frac{2}{8} + \frac{6}{8}\log_2\frac{6}{8}\right]$$

$$H(Y \mid B=0) = -\left[\frac{2}{4}\log_2\frac{2}{4} + \frac{2}{4}\log_2\frac{2}{4}\right]$$
$$H(Y \mid B=1) = -[0\log_2 0 + 1\log_2 1] = 0$$

$$H(Y \mid B) = P(B=0)H(Y \mid B=0) + P(B=1)H(Y \mid B=1)$$
$$= \frac{4}{8}H(Y \mid B=0) + \frac{4}{8} \cdot 0$$



$I(Y;B) = H(Y) - H(Y \mid B) > 0$

$I(Y;B)$ ends up being greater than $I(Y;A) = 0$, so we split on B

Slide credit: CMU MLD Matt Gormley

Mutual Information Notation

We use mutual information in the context of before and after a split, regardless of where that split is in the tree.

$$I(Y;X) = H(Y) - H(Y \mid X)$$

How to learn a decision tree

- Top-down induction [ID3]

Main loop:

1. $X \leftarrow$ the $\boxed{\text{“best”}}$ decision feature for next *node*
2. Assign X as decision feature for *node*
3. For each value of X, create new descendant of *node* (Discrete features)
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes (steps 1-5) after removing current feature

6. When all features exhausted, assign majority label to the leaf node

Slide credit: CMU MLD Aarti Singh

How to learn a decision tree

- Top-down induction [ID3, C4.5, C5, …]

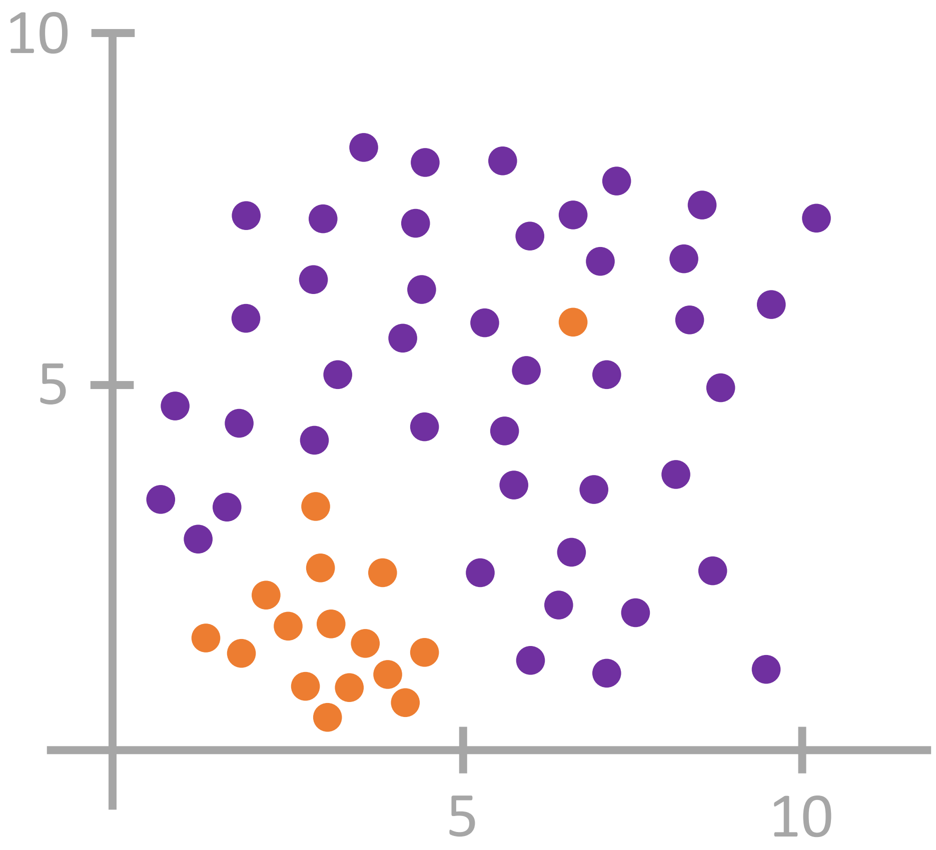
Main loop: C4.5

1. $X \leftarrow$ the "best" decision feature for next *node*
2. Assign X as decision feature for *node*
3. For "best" split of X, create new descendants of *node*
4. Sort training examples to leaf nodes
5. If training examples perfectly classified, Then STOP, Else iterate over new leaf nodes
6. Prune back tree to reduce overfitting
7. Assign majority label to the leaf node

Slide credit: CMU MLD Aarti Singh

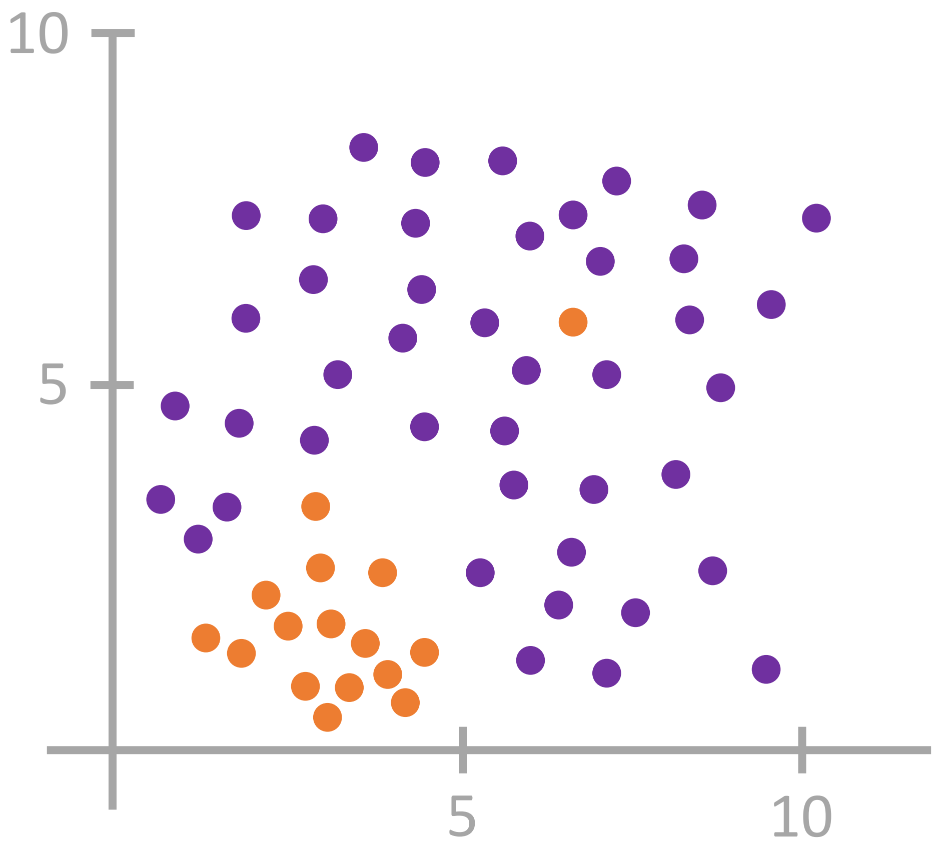
Continuous Features

Consider input features $x \in \mathbb{R}^2$



Generalization

Generalization: Ability to perform well on unseen data



Piazza Poll 1

Decision tree generalization

Which of the following 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

Piazza Poll 1

Decision tree generalization

Which of the following 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

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
- *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

Slide credit: CMU MLD Matt Gormley

Overfitting

- 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})

Overfitting

- 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: $error_{true}(h)$

- We say h overfits the training data if…

$$error_{true}(h) > error(h, D_{train})$$

- Amount of overfitting =

$$error_{true}(h) - error(h, D_{train})$$

In practice, $error_{true}(h)$ is **unknown**

Overfitting in Decision Tree Learning

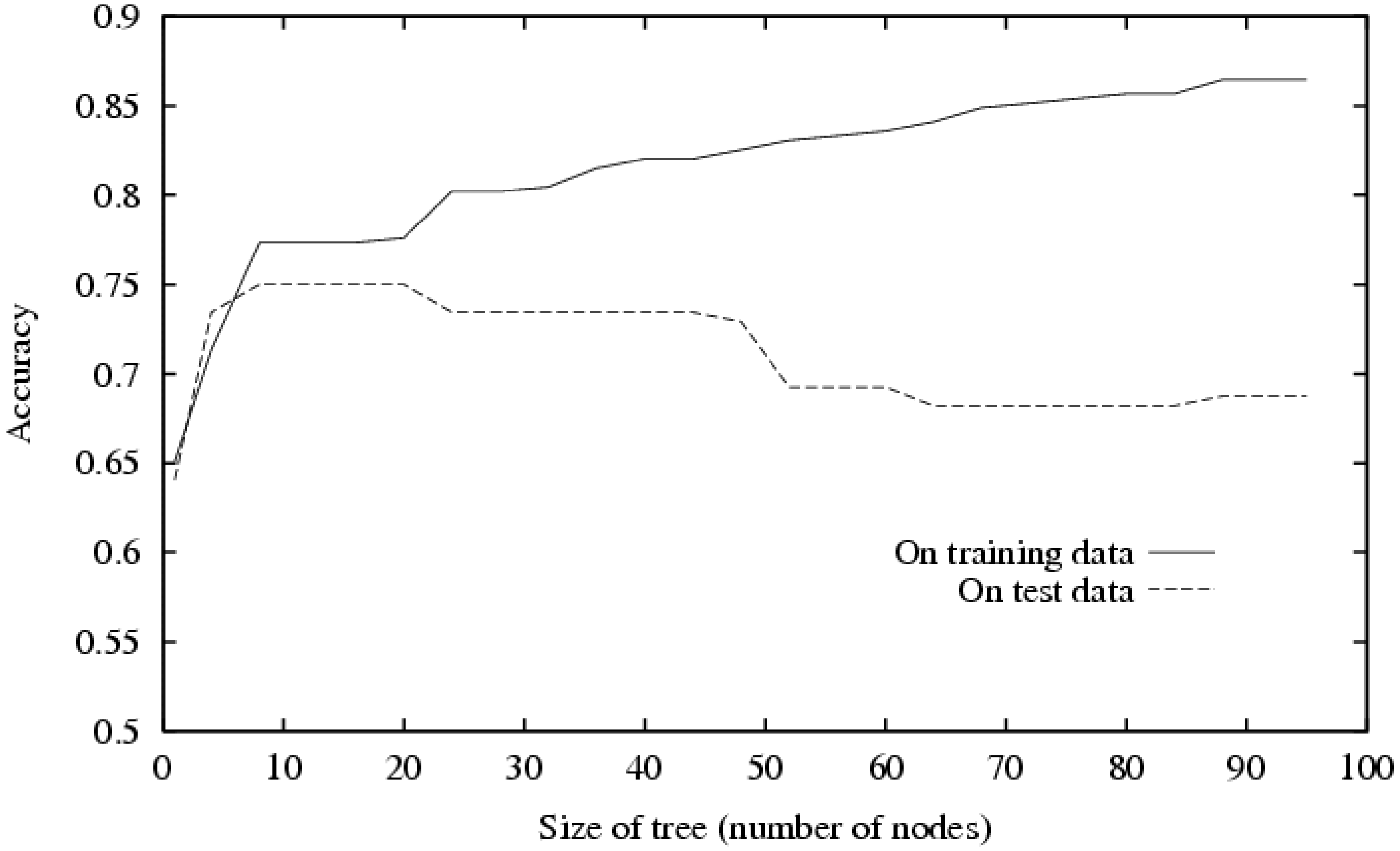


Figure from Tom Mitchell

How to Avoid Overfitting?

Many strategies for picking simpler trees:

- Fixed depth (e.g. ID3)
- Fixed number of leaves
- Mutual information threshold
- Grow entire tree then prune

Reduced-Error Pruning

Split data into *training* and *validation* set

Create tree that classifies *training* set correctly

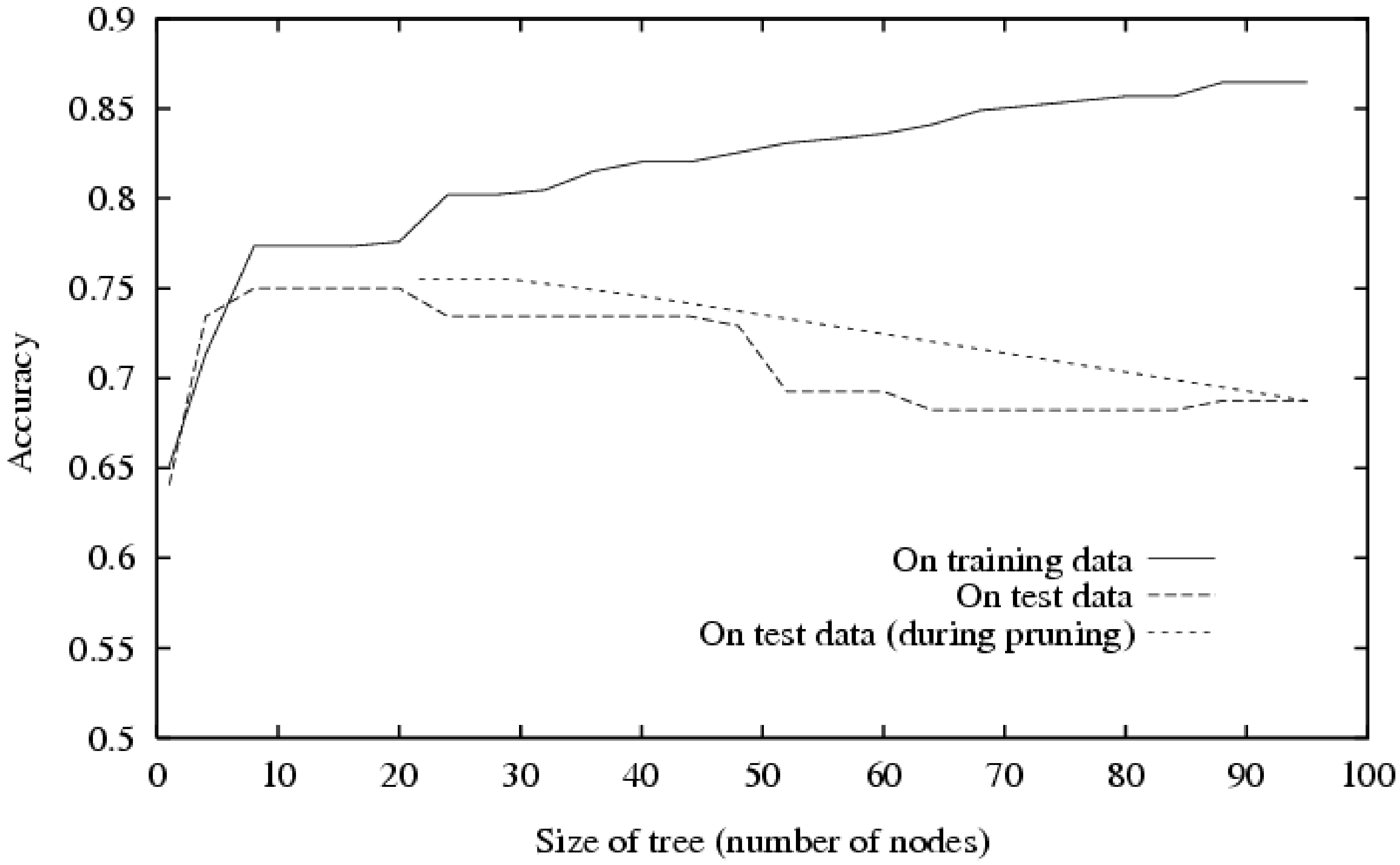
Do until further pruning is harmful:

1. Evaluate impact on *validation* set of pruning each possible node (plus those below it)
2. Greedily remove the one that most improves *validation* set accuracy

- produces smallest version of most accurate subtree
- What if data is limited?

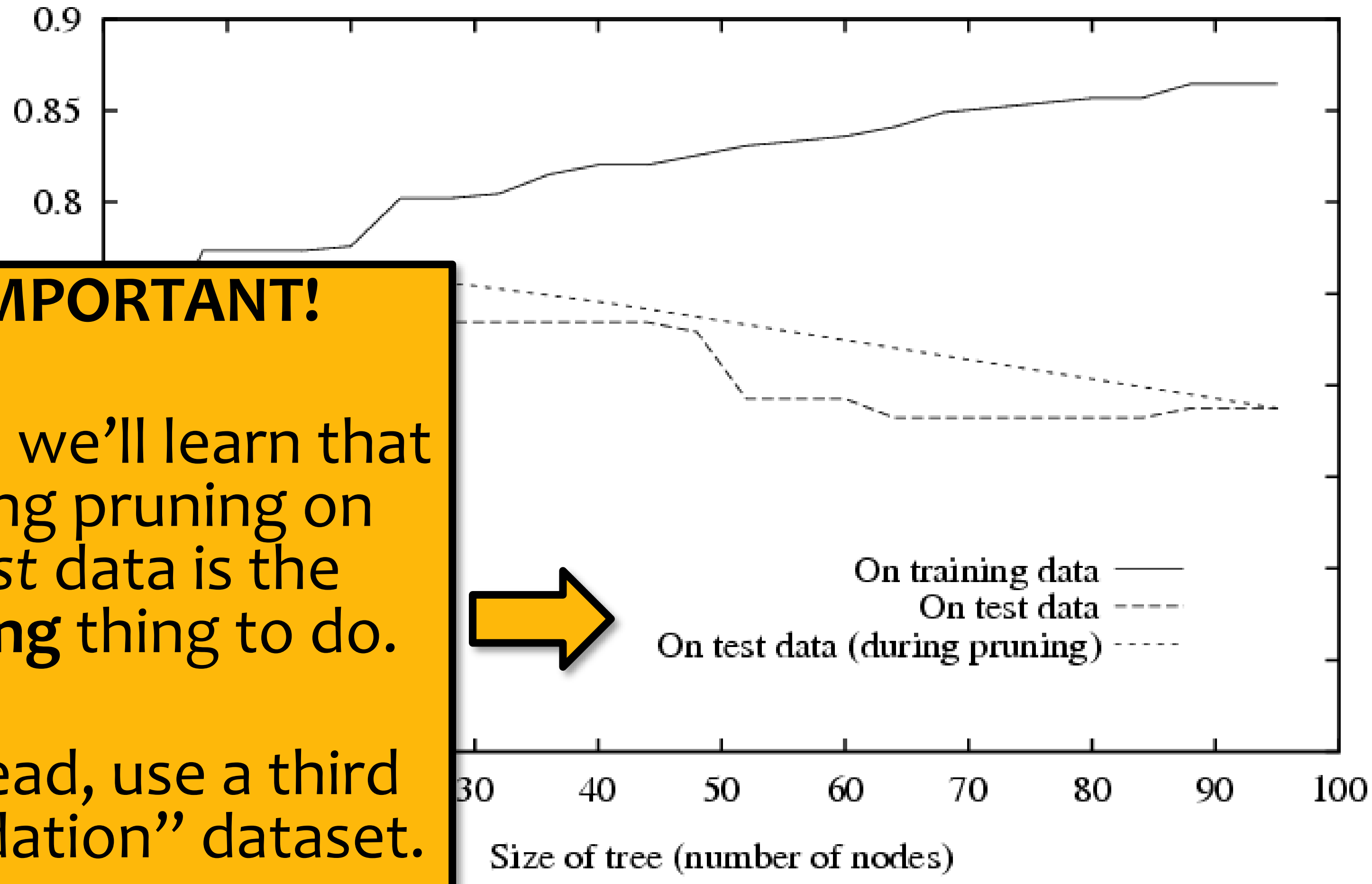
Slide from Tom Mitchell

Effect of Reduced-Error Pruning



Slide from Tom Mitchell

Effect of Reduced-Error Pruning



IMPORTANT!

Later, we'll learn that doing pruning on *test* data is the **wrong** thing to do.

Instead, use a third "validation" dataset.

Slide from Tom Mitchell

Inductive Bias

ID3 = Decision Tree Learning with Mutual Information and choosing attributes without replacement

Question: How does an ID3 tree generalize?

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

Slide credit: CMU MLD Matt Gormley

Decision Trees (DTs) in the Wild

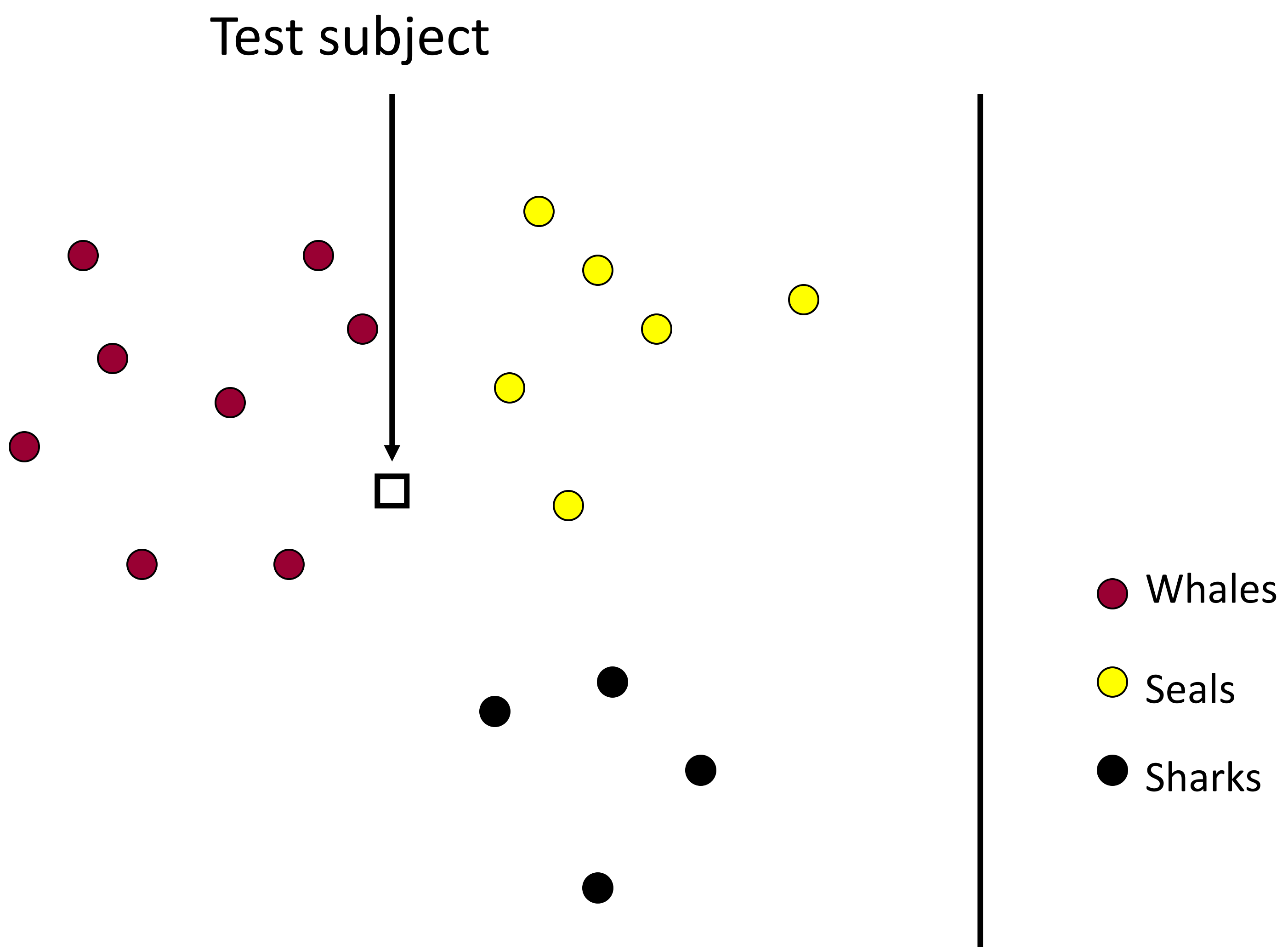
- DTs are one of the most popular classification methods for practical applications
  - Reason #1: The learned representation is **easy to explain** a non-ML person
  - Reason #2: They are **efficient** in both computation and memory
- DTs can be applied to a wide variety of problems including **classification, regression, density estimation**, etc.
- **Applications of DTs** include…
  - medicine, molecular biology, text classification, manufacturing, astronomy, agriculture, and many others
- **Decision Forests** learn many DTs from random subsets of features; the result is a very powerful example of an **ensemble method** (discussed later in the course)

DT Learning Objectives

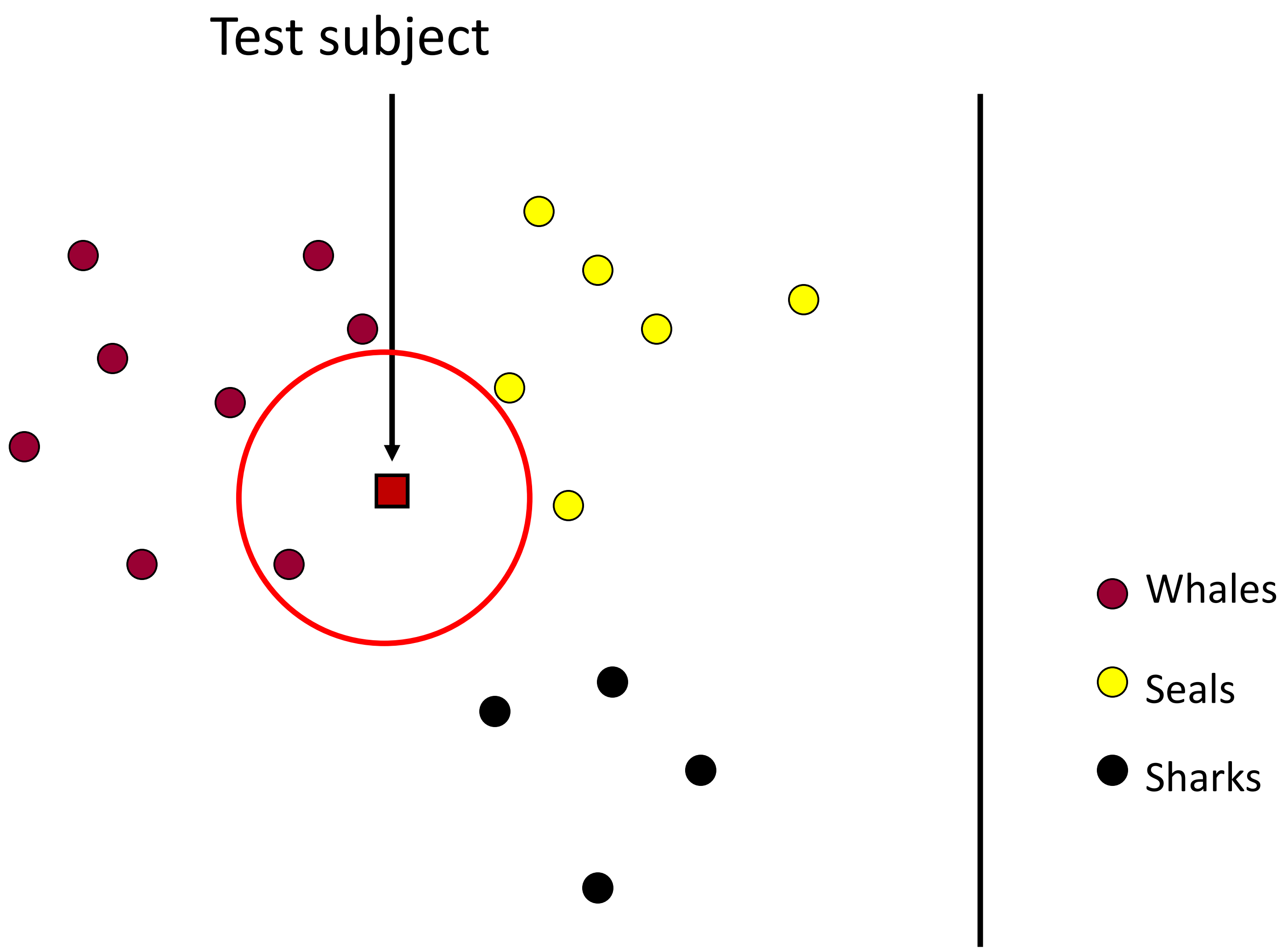
You should be able to…

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
3. Explain the difference between memorization and generalization
4. Describe the inductive bias of a decision tree
5. Formalize a learning problem by identifying the input space, output space, hypothesis space
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

Nearest Neighbor Classifier



Nearest Neighbor Classifier



Nearest Neighbor Classification

Given a training dataset $\mathcal{D} = \{y^{(n)}, \boldsymbol{x}^{(n)}\}_{n=1}^{N}$, $y \in \{1, \dots, C\}$, $\boldsymbol{x} \in \mathbb{R}^M$

and a test input $\boldsymbol{x}_{test}$, predict the class label, $\hat{y}_{test}$:

1) Find the closest point in the training data to $\boldsymbol{x}_{test}$

$$n = \operatorname*{argmin}_{n} d(\boldsymbol{x}_{test}, \boldsymbol{x}^{(n)})$$

2) Return the class label of that closest point

$$\hat{y}_{test} = y^{(n)}$$

Need distance function! What should $d(\boldsymbol{x}, \boldsymbol{z})$ be?

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

Full dataset: https://en.wikipedia.org/wiki/Iris_flower_data_set

Fisher Iris Dataset

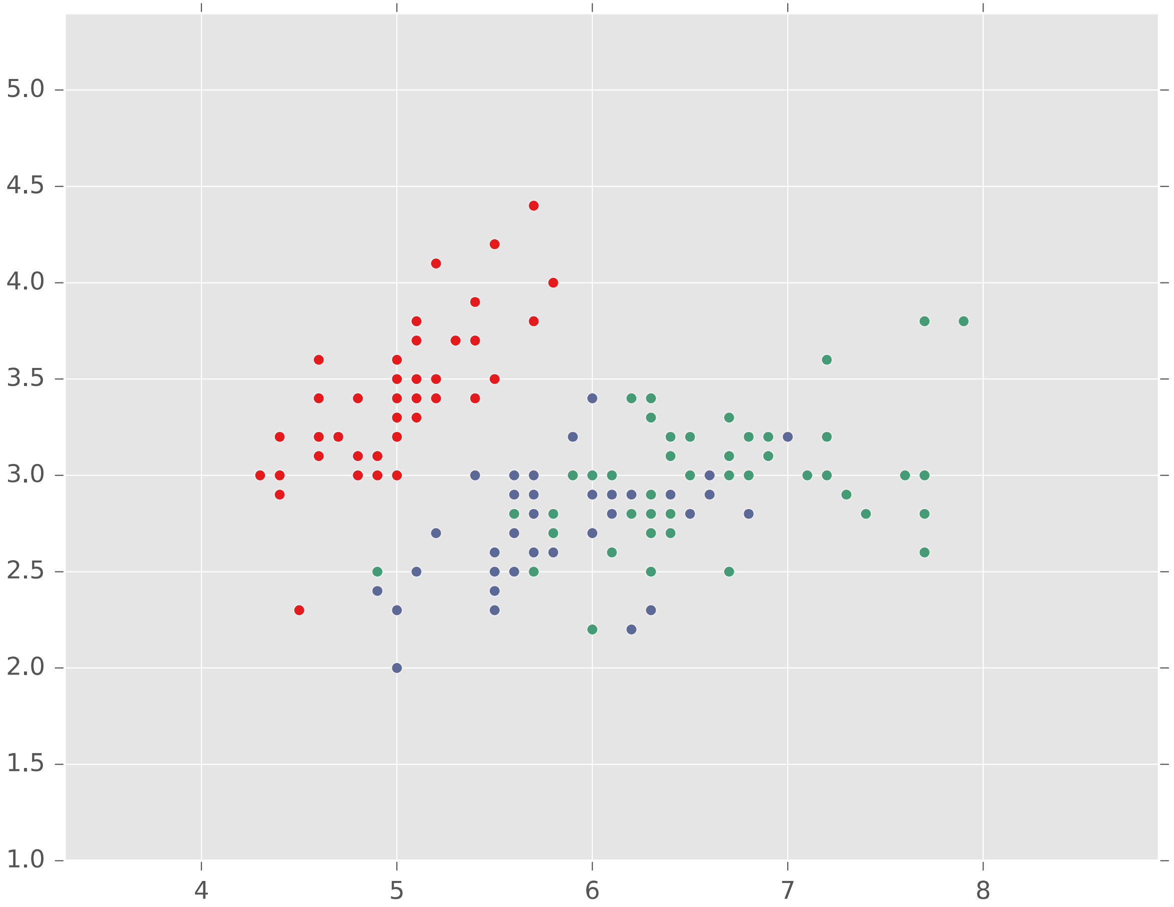
Fisher (1936) used 150 measurements of flowers from 3 different species: Iris setosa (0), Iris virginica (1), Iris versicolor (2) collected by Anderson (1936)

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

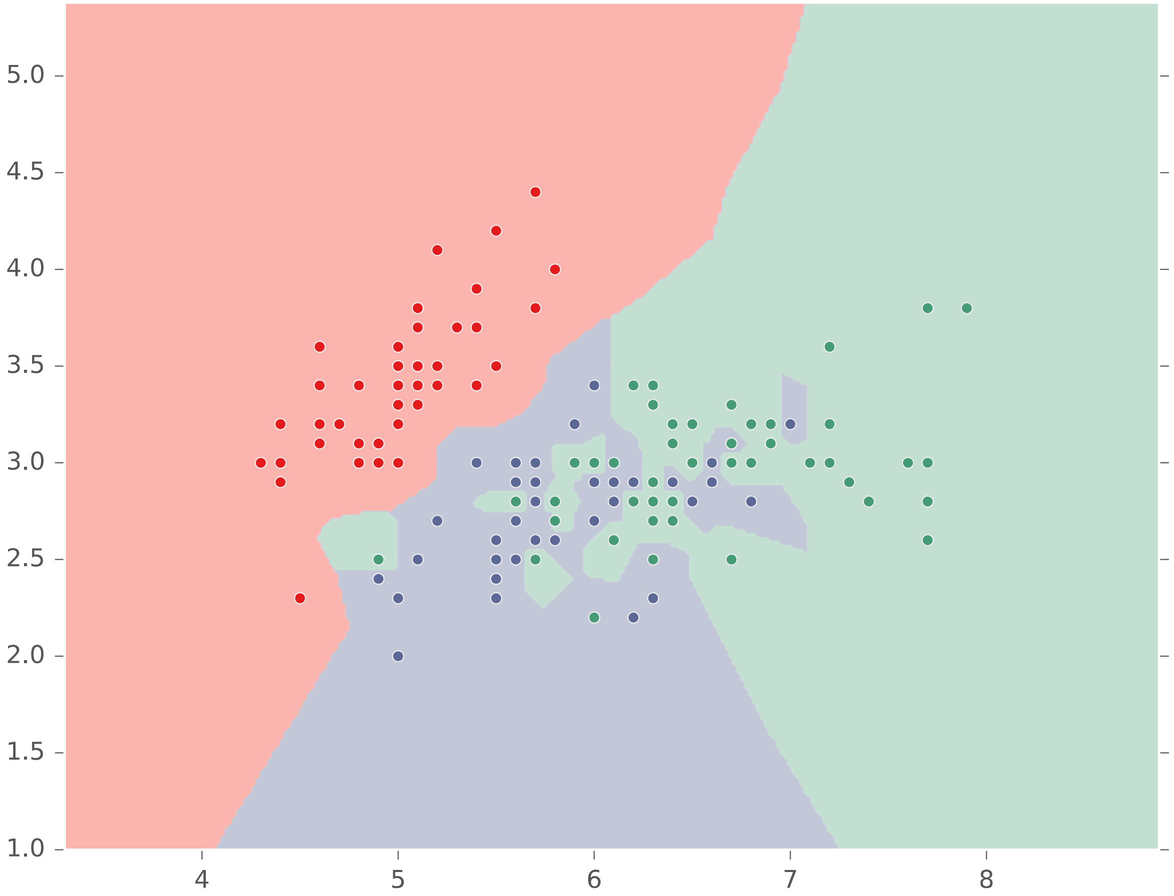
Full dataset: https://en.wikipedia.org/wiki/Iris_flower_data_set

Nearest Neighbor on Fisher Iris Data



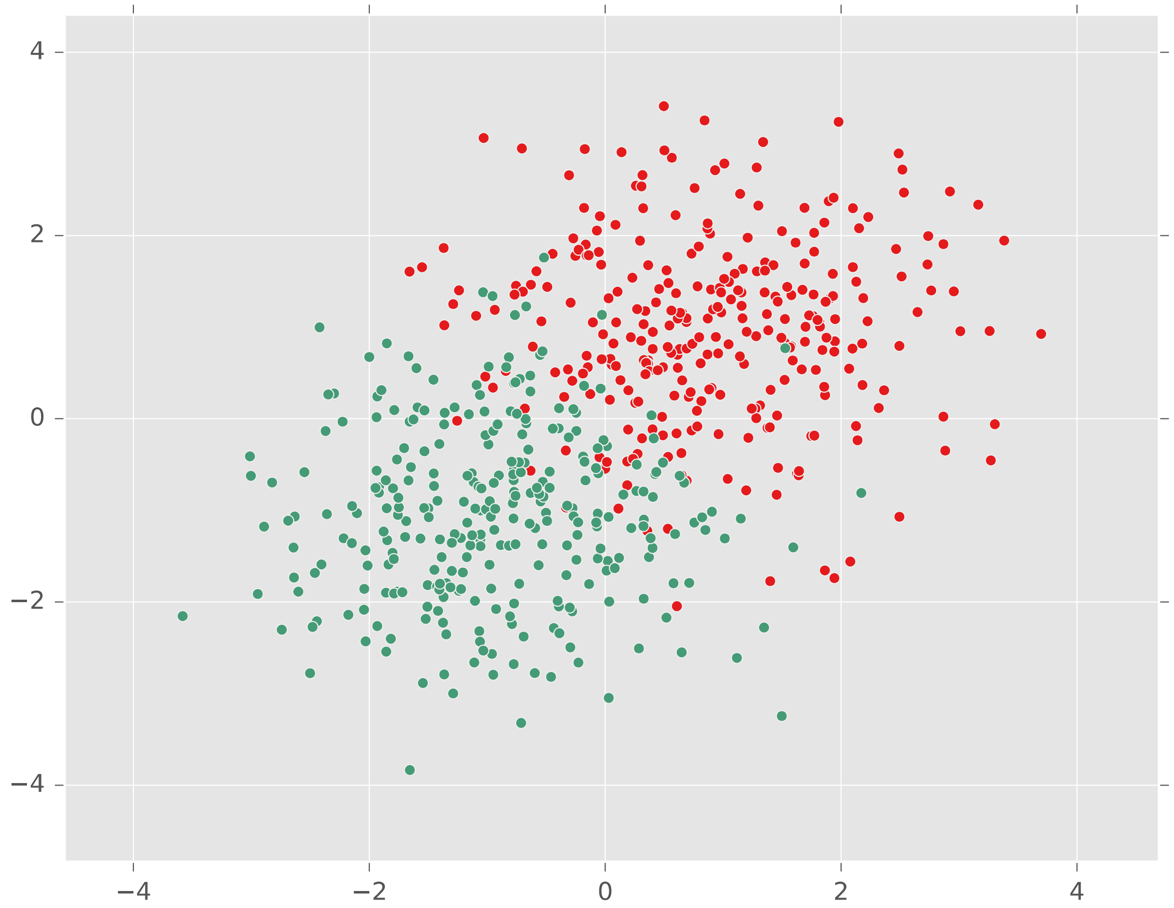
Slide credit: CMU MLD Matt Gormley

Nearest Neighbor on Fisher Iris Data

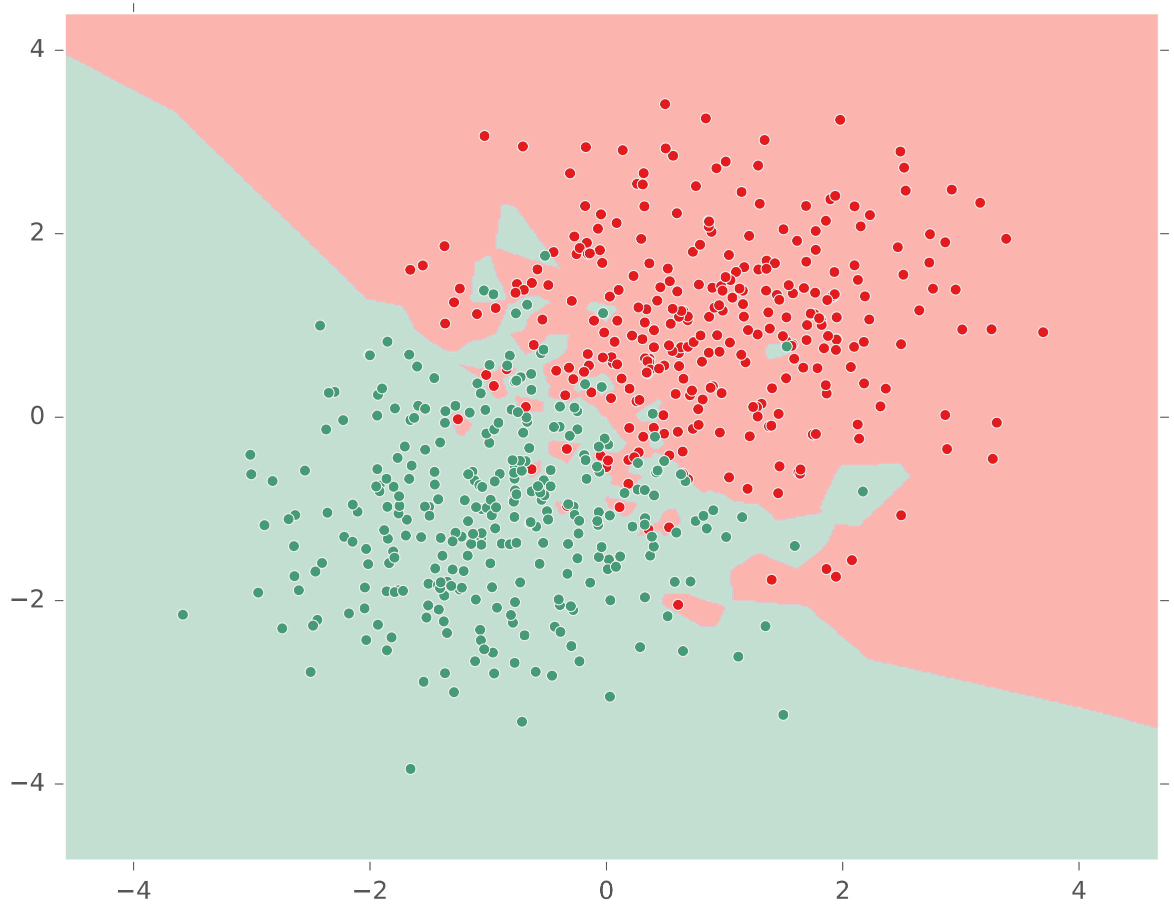


Slide credit: CMU MLD Matt Gormley

Nearest Neighbor on Gaussian Data

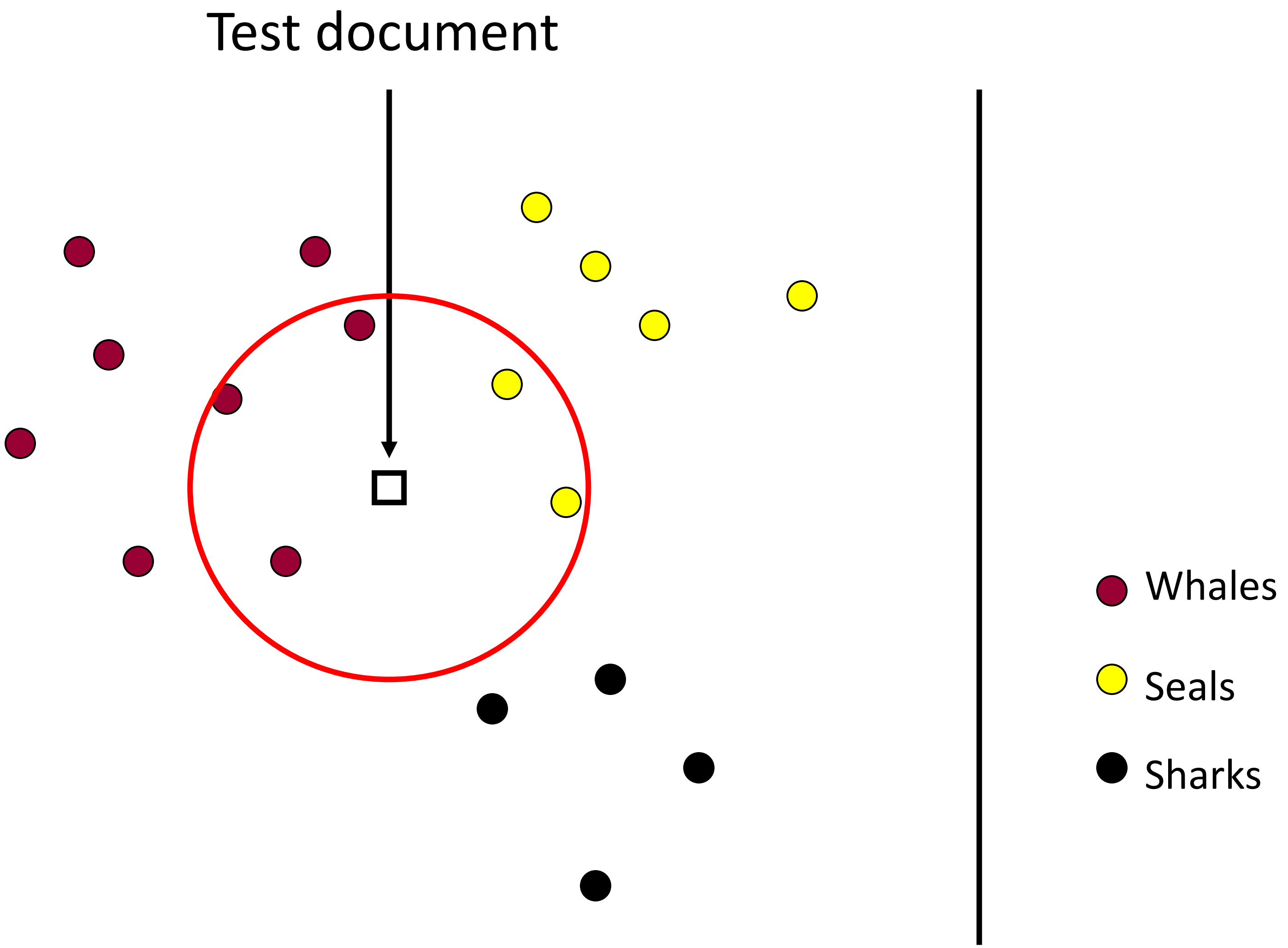


Nearest Neighbor on Gaussian Data



Slide credit: CMU MLD Matt Gormley

kNN classifier (k=5)



Nearest Neighbor Classification

Given a training dataset $\mathcal{D} = \{y^{(n)}, \boldsymbol{x}^{(n)}\}_{n=1}^{N}$, $y \in \{1, \dots, C\}$, $\boldsymbol{x} \in \mathbb{R}^M$

and a test input $\boldsymbol{x}_{test}$, predict the class label, $\hat{y}_{test}$:

1) Find the closest point in the training data to $\boldsymbol{x}_{test}$

$$n = \operatorname*{argmin}_n d(\boldsymbol{x}_{test}, \boldsymbol{x}^{(n)})$$

2) Return the class label of that closest point

$$\hat{y}_{test} = y^{(n)}$$

Need distance function! What should $d(\boldsymbol{x}, \boldsymbol{z})$ be?

k-Nearest Neighbor Classification

Given a training dataset $\mathcal{D} = \{y^{(n)}, \boldsymbol{x}^{(n)}\}_{n=1}^{N}$, $y \in \{1, \dots, C\}$, $\boldsymbol{x} \in \mathbb{R}^M$

and a test input $\boldsymbol{x}_{test}$, predict the class label, $\hat{y}_{test}$:

1) Find the closest k points in the training data to $\boldsymbol{x}_{test}$

$$\mathcal{N}_k(\boldsymbol{x}_{test}, \mathcal{D})$$

2) Return the class label of that closest point

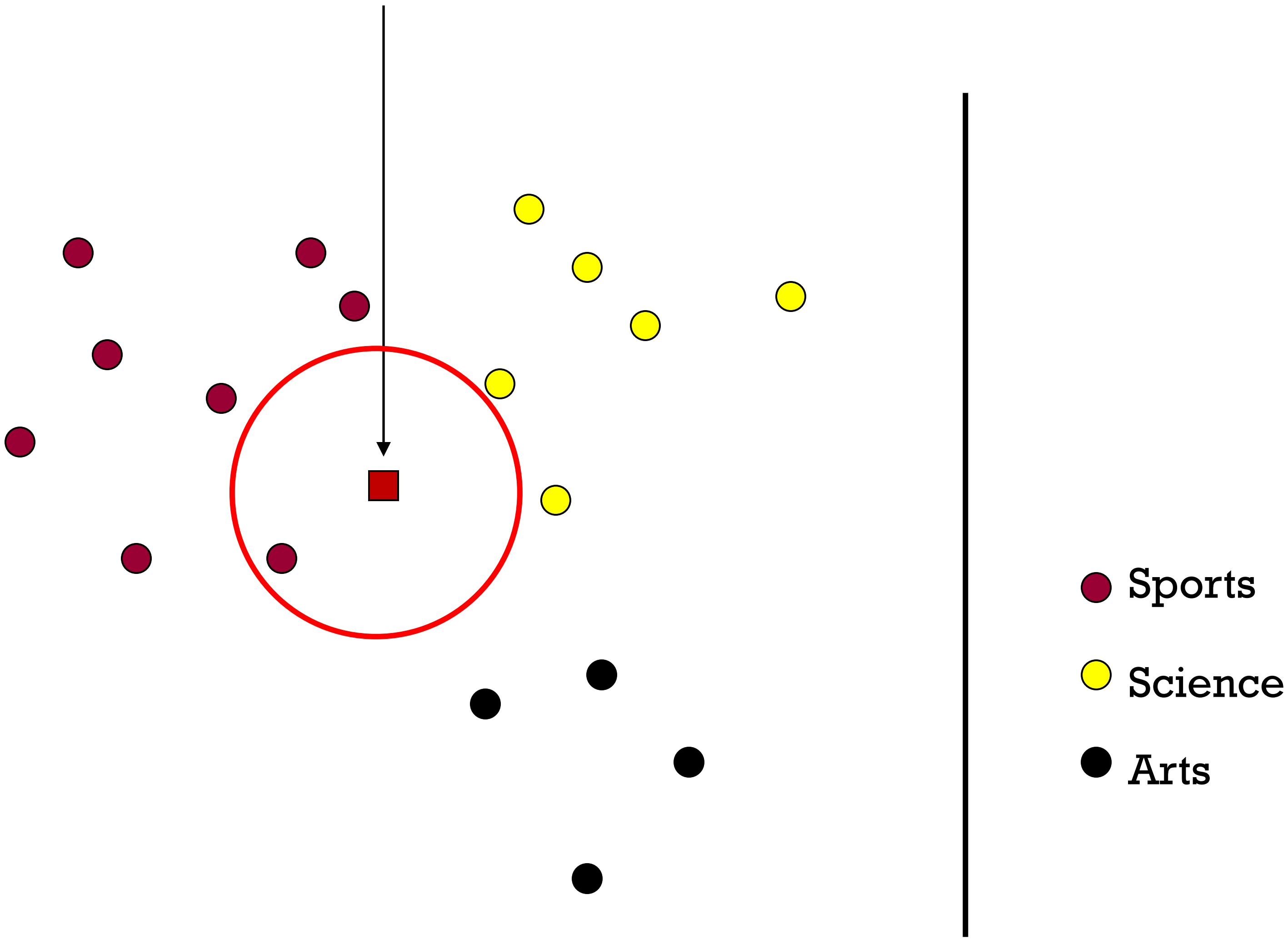
$$\hat{y}_{test} = \underset{c}{\operatorname{argmax}}\, p(Y = c \mid \boldsymbol{x}_{test}, \mathcal{D}, k)$$

$$= \underset{c}{\operatorname{argmax}} \frac{1}{k} \sum_{i \in \mathcal{N}_k(\boldsymbol{x}_{test}, \mathcal{D})} \mathbb{I}(y^{(i)} = c)$$

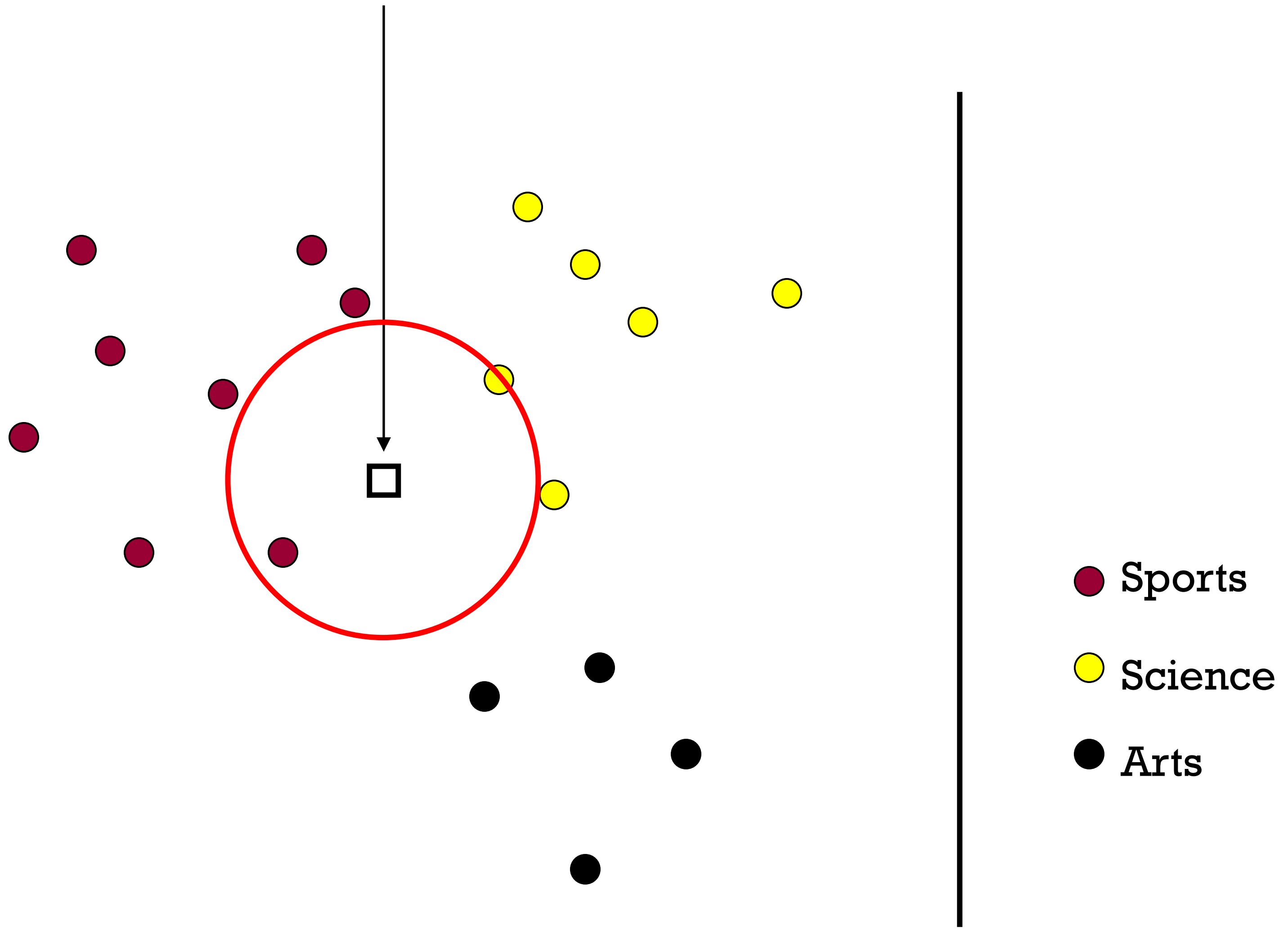
$$= \underset{c}{\operatorname{argmax}} \frac{k_c}{k},$$

where k_c is the number of the k-neighbors with class label c

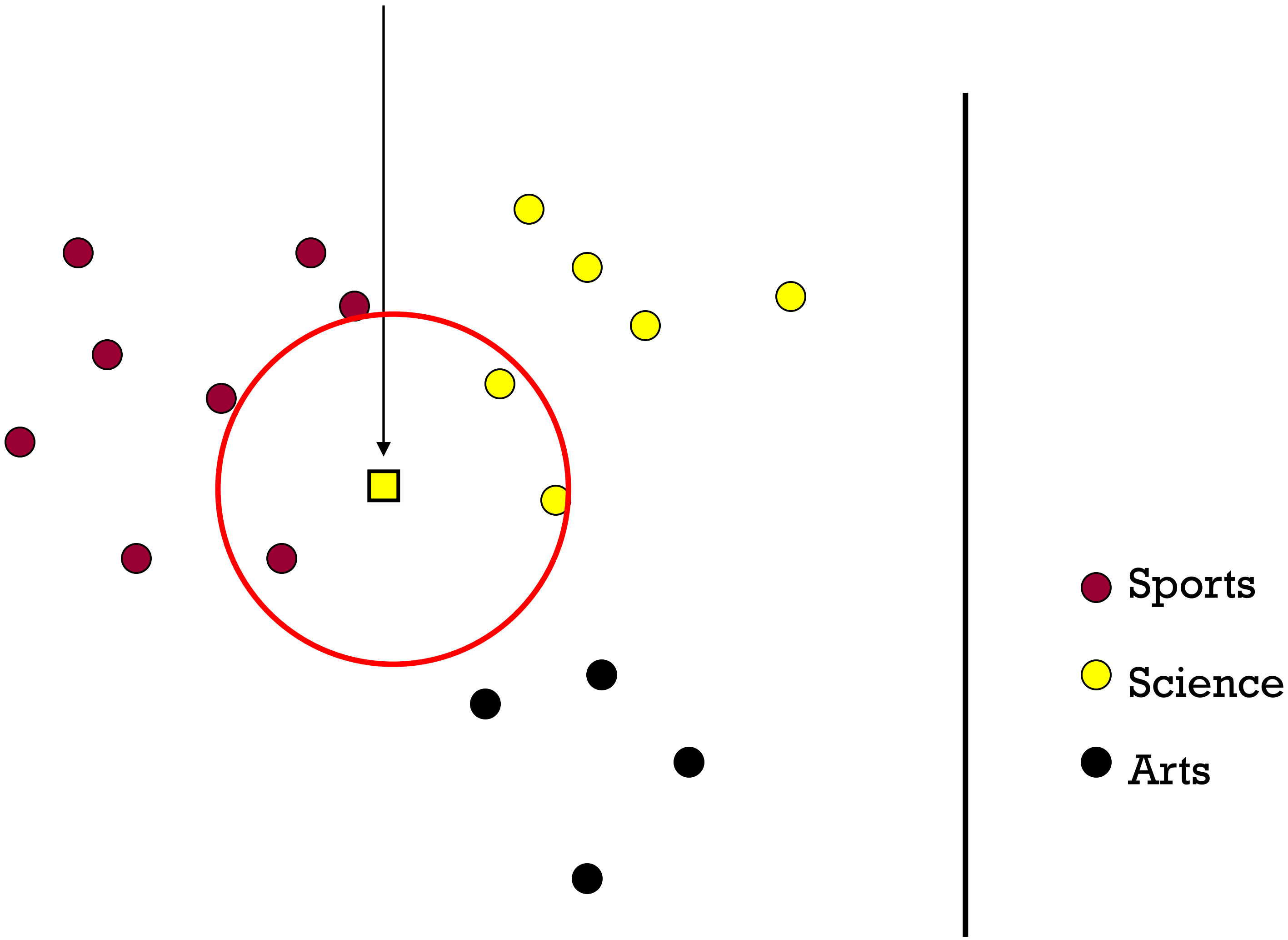
1-Nearest Neighbor (kNN) classifier



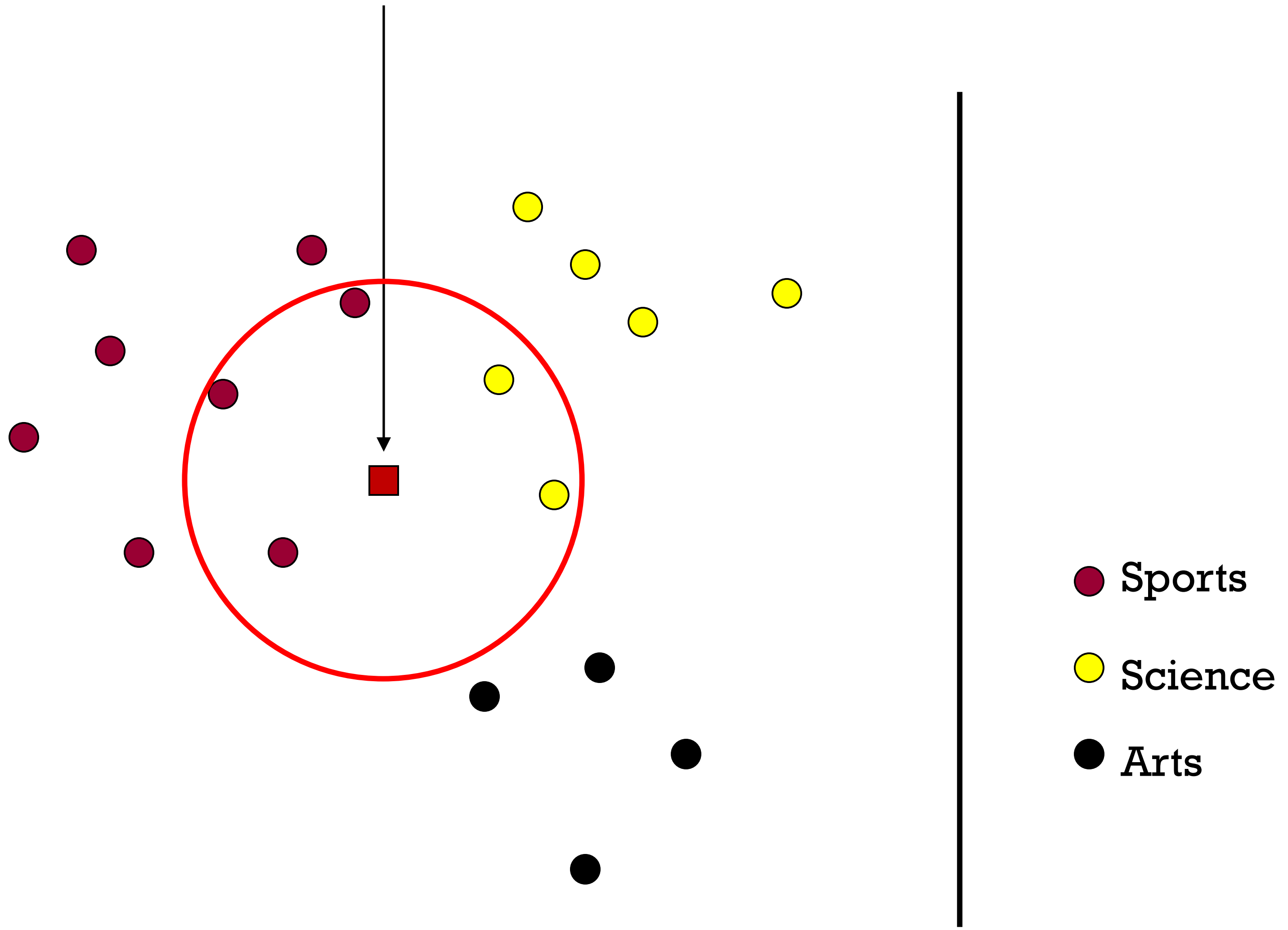
2-Nearest Neighbor (kNN) classifier



3-Nearest Neighbor (kNN) classifier



5-Nearest Neighbor (kNN) classifier



What is the best k?

How do we choose a learner that is accurate and also generalizes to unseen data?

- Larger k → predicted label is more stable
- Smaller k → predicted label is more affected by individual training points

But how to choose k?