10-301/601: Introduction to Machine Learning Lecture 4 – Overfitting & KNNs

Henry Chai & Matt Gormley

9/12/22

Q & A:

What do we predict in leaf nodes with no training data?

- Well, there isn't really a majority label, so (if forced to) we could return a random label or a majority vote over the entire training dataset.
- This is related to the question of "what should we predict if some feature in our test dataset takes on a value we didn't observe in the training dataset?"
  - Going down a branch corresponding to an unseen feature value is like hitting a leaf node with no training data.

Q & A:

When training decision trees, what can we do if the set of values a feature can take on is really large or even infinite?

- If your features are real-valued, there's a clever trick that only considers $O(L)$ splits where $L =$ the # of values the feature takes on in the training set. Can you guess what it does?

Front Matter

- Announcements:
  - HW2 released 9/7, due 9/19 at 11:59 PM
  - HW1 exit poll released 9/8, you must respond by 9/15 in order to receive full credit.

Q & A:

Wait, I thought you said the exit poll wouldn't come out until a few days after HW1 was due?

- Ah yes, that's my bad: I forgot that we use your responses to the exit poll to guide our solution sessions.
- Follow-up: what's a "solution session"?
  - Oh, great question! Instead of releasing solutions to our HWs, our TAs will go through the solutions with you all (using your feedback from the exit polls) and field any questions you may have.
  - These sessions will be live-streamed and recorded but *the recording will only be available for 24 hours*.
  - All the details for HW1's solution session have been announced on Piazza.

Q & A:

Speaking of TAs...

- One of your TAs is my [lab partner, teammate, best friend, father's brother's nephew's cousin's former roommate, etc…]: is it alright if I bug them extra help on [LinkedIn, Snapgram, the bus, their way home, etc...]?
- No.
- **ABSOLUTELY NOT!**
  - You may only interact with our TAs about course content during course sanctioned events (e.g., OH or recitations) or on course sanctioned platforms (e.g., Piazza).

Recall: Mutual Information

- Mutual information describes how much information or clarity a particular feature provides about the label

$$I(Y; x_d) = H(Y) - \sum_{v \in V(x_d)} (f_v)\left(H\left(Y_{x_d=v}\right)\right)$$

where x_d is a feature

Y is the collection of all labels

$V(x_d)$ is the set of unique values of x_d

f_v is the fraction of inputs where $x_d = v$

$Y_{x_d=v}$ is the collection of labels where $x_d = v$

$H\left(Y_{x_d=v}\right)$ is the conditional entropy of Y given $x_d = v$

Mutual Information: Alternate Notation

- Mutual information describes how much information or clarity a particular feature provides about the label

$$I(Y; x_d) = H(Y) - \sum_{v \in V(x_d)} (f_v)\big(H(Y|x_d = v)\big)$$

where x_d is a feature

Y is the collection of all labels

$V(x_d)$ is the set of unique values of x_d

f_v is the fraction of inputs where $x_d = v$

$H(Y|x_d = v)$ is the conditional entropy of Y given $x_d = v$ i.e., the entropy of all labels corresponding to examples where $x_d = v$

Mutual Information: Alternate Notation

- Mutual information describes how much information or clarity a particular feature provides about the label

$$I(Y; x_d) = H(Y) - H(Y|x_d)$$

where x_d is a feature

Y is the collection of all labels

$V(x_d)$ is the set of unique values of x_d

$H(Y|x_d)$ is the conditional entropy of Y given x_d

Mutual Information: Example

x_d	y
1	1
1	1
0	0
0	0

$$I(Y; x_d) = H(Y) - \sum_{v \in V(x_d)} (f_v)\left(H\left(Y_{x_d=v}\right)\right)$$

$$= 1 - \frac{1}{2} H\left(Y_{x_d=0}\right) - \frac{1}{2} H\left(Y_{x_d=1}\right)$$

$$= 1 - \frac{1}{2}(0) - \frac{1}{2}(0) = 1$$

Mutual Information: Example

x_d	y
1	1
0	1
1	0
0	0

$$I(Y; x_d) = H(Y) - \sum_{v \in V(x_d)} (f_v)\left(H\left(Y_{x_d=v}\right)\right)$$

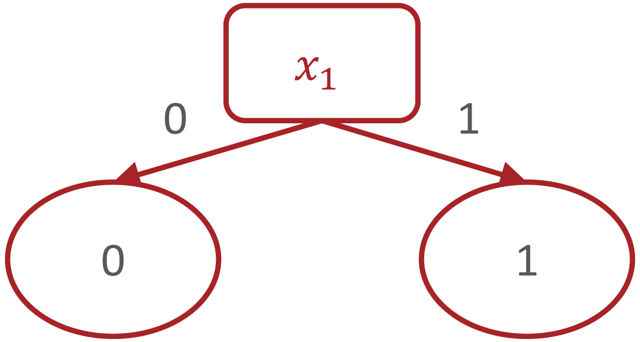
$$= 1 - \frac{1}{2} H\left(Y_{x_d=0}\right) - \frac{1}{2} H\left(Y_{x_d=1}\right)$$

$$= 1 - \frac{1}{2}(1) - \frac{1}{2}(1) = 0$$

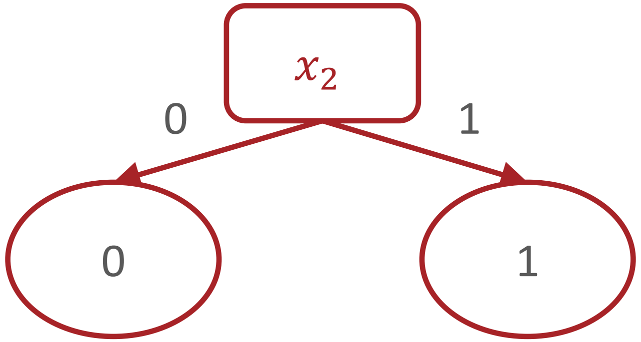
~~Poll Question 3:~~

Which feature would you split on using mutual information as the splitting criterion?

x_1	x_2	y
1	0	0
1	0	0
1	0	1
1	0	1
1	1	1
1	1	1
1	1	1
1	1	1



Mutual Information: 0



Mutual Information: $H(Y) - \frac{1}{2}H\left(Y_{x_2=0}\right) - \frac{1}{2}H\left(Y_{x_2=1}\right)$

$$= -\frac{2}{8}\log_2\frac{2}{8} - \frac{6}{8}\log_2\frac{6}{8} - \frac{1}{2}(1) - \frac{1}{2}(0) \approx 0.31$$

Bluntine & Niblett (1992) Splitting Criteria Experiments

Table 3. Error for different splitting rules (pruned trees).

Data Set	Splitting Rule			
	GINI	Info. Gain	Marsh.	Random
hypo	1.01 ± 0.29	0.95 ± 0.22	1.27 ± 0.47	7.44 ± 0.53
breast	28.66 ± 3.87	28.49 ± 4.28	27.15 ± 4.22	29.65 ± 4.97
tumor	60.88 ± 5.44	62.70 ± 3.89	61.62 ± 3.98	67.94 ± 5.68
lymph	24.44 ± 6.92	24.00 ± 6.87	24.33 ± 5.51	32.33 ± 11.25
LED	33.77 ± 3.06	32.89 ± 2.59	33.15 ± 4.02	38.18 ± 4.57
mush	1.44 ± 0.47	1.44 ± 0.47	7.31 ± 2.25	8.77 ± 4.65
votes	4.47 ± 0.95	4.57 ± 0.87	11.77 ± 3.95	12.40 ± 4.56
votes1	12.79 ± 1.48	13.04 ± 1.65	15.13 ± 2.89	15.62 ± 2.73
iris	5.00 ± 3.08	4.90 ± 3.08	5.50 ± 2.59	14.20 ± 6.77
glass	39.56 ± 6.20	50.57 ± 6.73	40.53 ± 6.41	53.20 ± 5.01
xd6	22.14 ± 3.23	22.17 ± 3.36	22.06 ± 3.37	31.86 ± 3.62
pole	15.43 ± 1.51	15.47 ± 0.88	15.01 ± 1.15	26.38 ± 6.92

Key takeaway: Gini impurity and Information gain (aka mutual information) are nearly identical

Bluntine & Niblett (1992) Splitting Criteria Experiments

Table 4. Difference and significance of error for GINI splitting rule versus others.

	Splitting Rule		
Data Set	Info. Gain	Marsh.	Random
hypo	−0.06 (0.82)	0.26 (0.99)	6.43 (1.00)
breast	−0.17 (0.23)	−1.51 (0.94)	0.99 (0.72)
tumor	1.81 (0.84)	0.74 (0.39)	7.06 (0.99)
lymph	−0.44 (0.83)	−0.11 (0.05)	7.89 (0.99)
LED	0.12 (0.17)	0.38 (0.41)	5.41 (0.99)
mush	0.00 (0.00)	5.86 (1.00)	7.32 (0.99)
votes	0.11 (0.55)	7.30 (0.99)	7.94 (0.99)
votes1	0.26 (0.47)	2.34 (0.98)	2.83 (0.99)
iris	−0.10 (0.67)	0.50 (0.90)	9.20 (0.99)
glass	1.01 (0.50)	0.96 (0.53)	13.64 (0.99)
xd6	0.04 (0.11)	−0.07 (0.20)	9.72 (0.99)
pole	0.03 (0.11)	−0.43 (0.83)	10.95 (0.99)

average difference in error between the splitting criterion

statistical significance of the difference according to a two-sided paired t-test

Decision Tree: Pseudocode

def train($\mathcal{D}$):

  store root = tree_recurse($\mathcal{D}$)

def tree_recurse($\mathcal{D}'$):

  q = new node()

  base case – if (SOME CONDITION):

  recursion – else:

    find best attribute to split on, x_d

    q.split = x_d

    for v in $V(x_d)$, all possible values of x_d:

$$\mathcal{D}_v = \left\{\left(x^{(n)}, y^{(n)}\right) \in \mathcal{D} \mid x_d^{(n)} = v\right\}$$

      q.children(v) = tree_recurse($\mathcal{D}_v$)

  return q

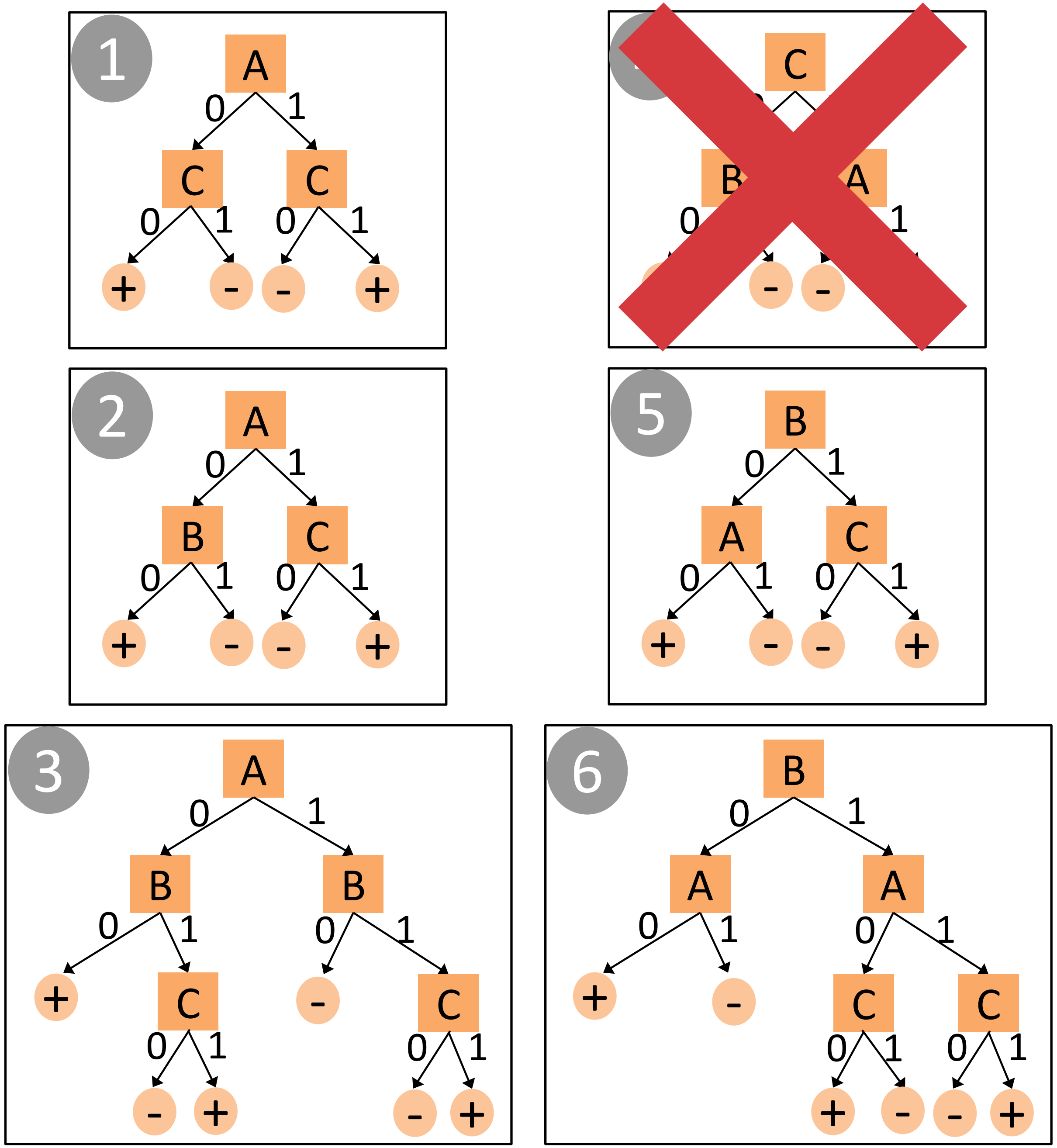
Decision Tree: Pseudocode

```
def train(𝒟):
    store root = tree_recurse(𝒟)
def tree_recurse(𝒟′):
    q = new node()
    base case – if (𝒟′ is empty OR
        all labels in 𝒟′ are the same OR
        all features in 𝒟′ are identical OR
        some other stopping criterion):
        q.label = majority_vote(𝒟′)


    recursion – else:
    return q
```

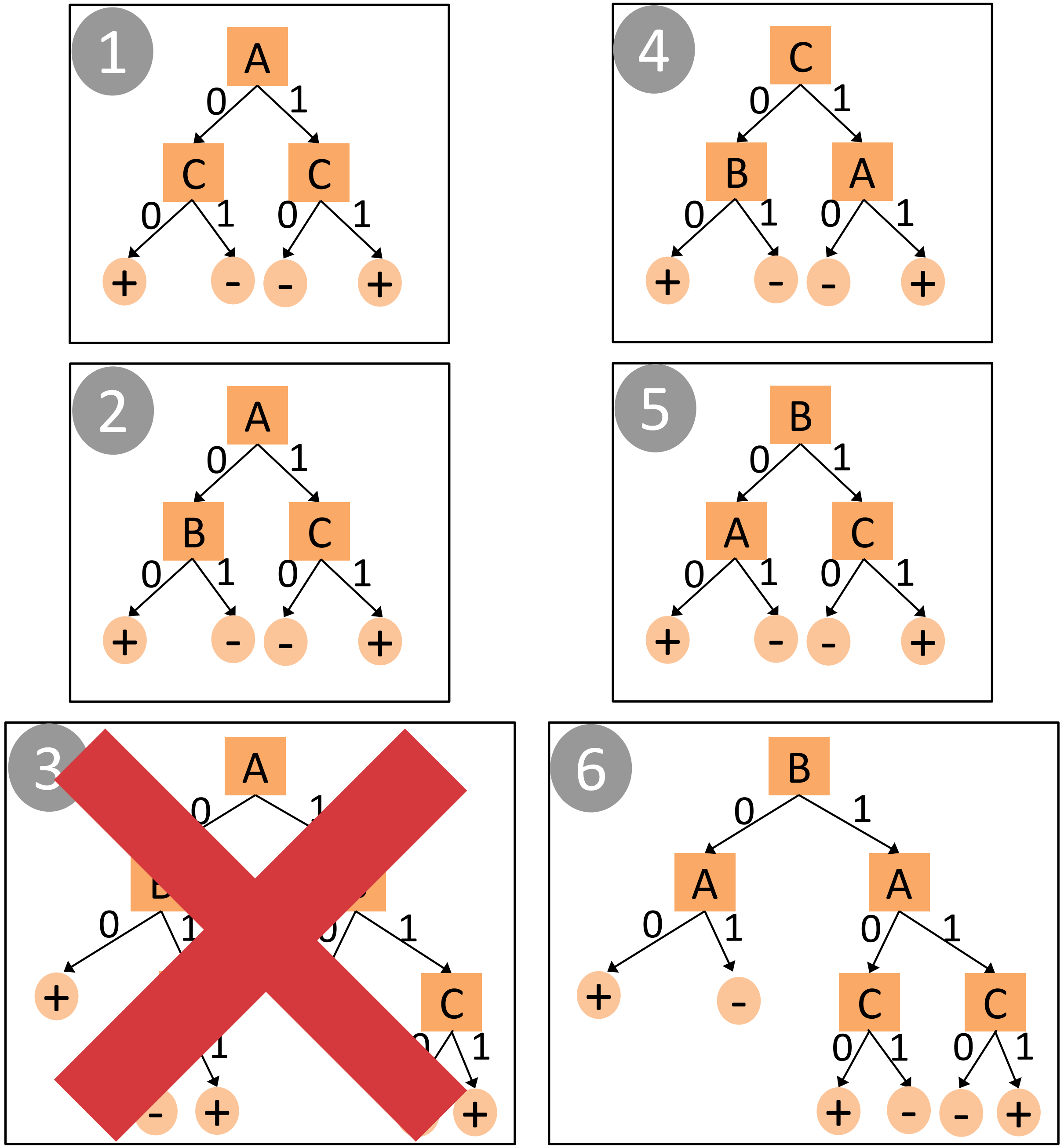
Poll Question 1: Which decision tree would you learn if you used training error rate as the splitting criterion? Break ties alphabetically.

A	B	C	y
0	0	0	+
0	0	1	+
0	1	0	−
0	1	1	+
1	0	0	−
1	0	1	−
1	1	0	−
1	1	1	+



Poll Question 2: Which decision tree is the smallest decision tree that achieves the lowest possible training error?

A	B	C	y
0	0	0	+
0	0	1	+
0	1	0	–
0	1	1	+
1	0	0	–
1	0	1	–
1	1	0	–
1	1	1	+



Decision Trees: Pros & Cons

- Pros
  - Interpretable
  - Efficient (computational cost and storage)
  - Can be used for classification and regression tasks
  - Compatible with categorical and real-valued features
- Cons
  - Learned greedily: each split only considers the immediate impact on the splitting criterion
    - Not guaranteed to find the smallest (fewest number of splits) tree that achieves a training error rate of 0.
  - Liable to overfit!

Decision Trees: Inductive Bias

- The **inductive bias** of a machine learning algorithm is the principle by which it generalizes to unseen examples
- What is the inductive bias of the ID3 algorithm?
  - Try to find the smallest tree that achieves a **training error rate of 0** with high mutual information features at the top
- Occam's razor: try to find the "simplest" (e.g., smallest decision tree) classifier that explains the training dataset

Overfitting

- Overfitting occurs when the classifier (or model)…
  - is too complex
  - fits noise or "outliers" in the training dataset as opposed to the actual pattern of interest
  - doesn't have enough inductive bias pushing it to generalize
- Underfitting occurs when the classifier (or model)…
  - is too simple
  - can't capture the actual pattern of interest in the training dataset
  - has too much inductive bias

Recall: Different Kinds of Error

- Training error rate = $err(h, \mathcal{D}_{train})$
- Test error rate = $err(h, \mathcal{D}_{test})$
- True error rate = $err(h)$

  = the error rate of h on all possible examples

  - In machine learning, this is the quantity that we care about but, in most cases, it is unknowable.
- Overfitting occurs when $err(h) > err(h, \mathcal{D}_{train})$
  - $err(h) - err(h, \mathcal{D}_{train})$ can be thought of as a measure of overfitting

Overfitting in Decision Trees

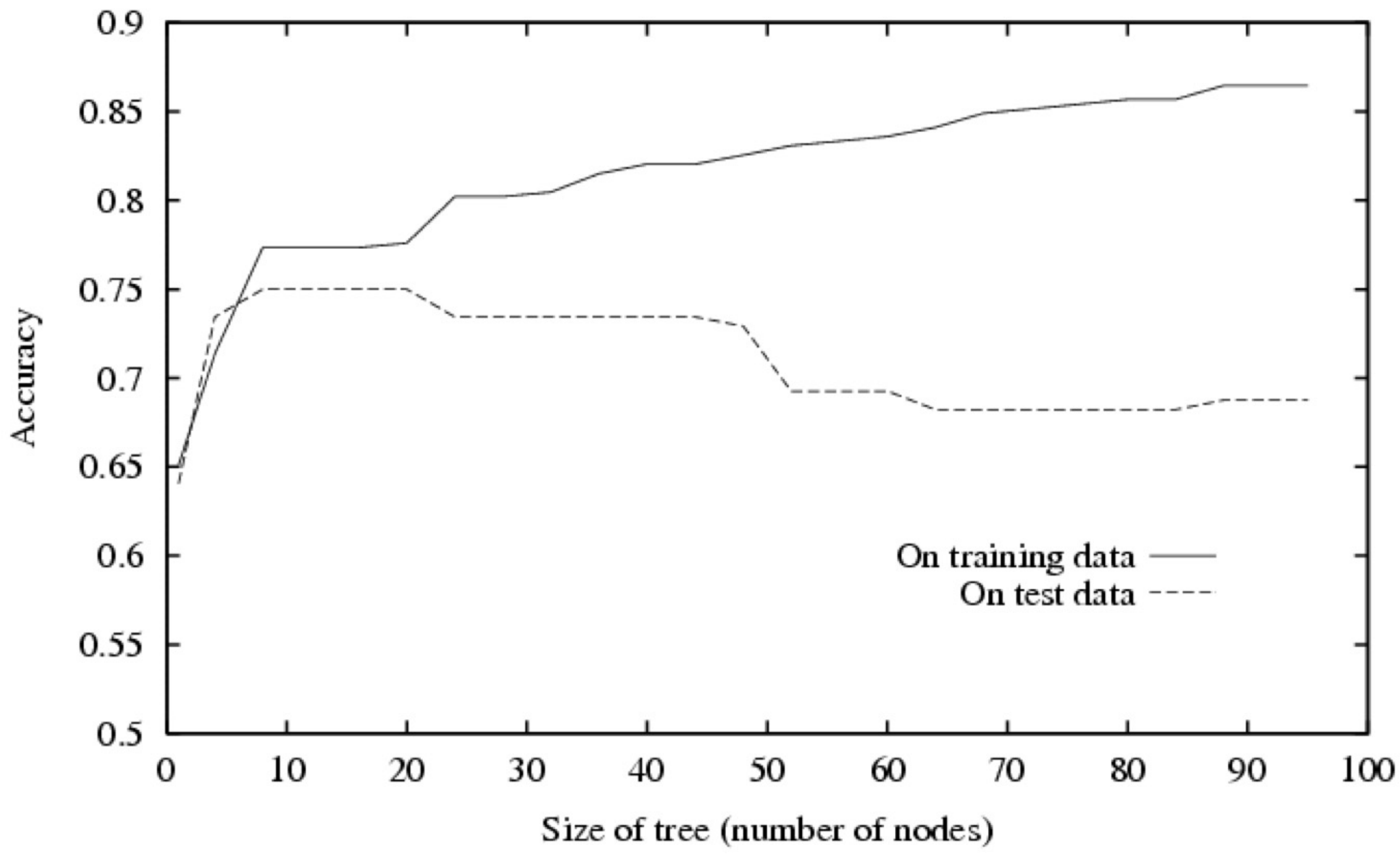


Figure courtesy of Tom Mitchell

Combatting Overfitting in Decision Trees

- Heuristics:
  - Do not split leaves past a fixed depth, δ
  - Do not split leaves with fewer than c data points
  - Do not split leaves where the maximal information gain is less than τ
- Take a majority vote in impure leaves

Combatting Overfitting in Decision Trees

- Pruning:
  - First, learn a decision tree
  - Then, evaluate each split using a “validation” dataset by comparing the validation error rate with and without that split
  - Greedily remove the split that most decreases the validation error rate
  - Stop if no split is removed

Pruning Decision Trees

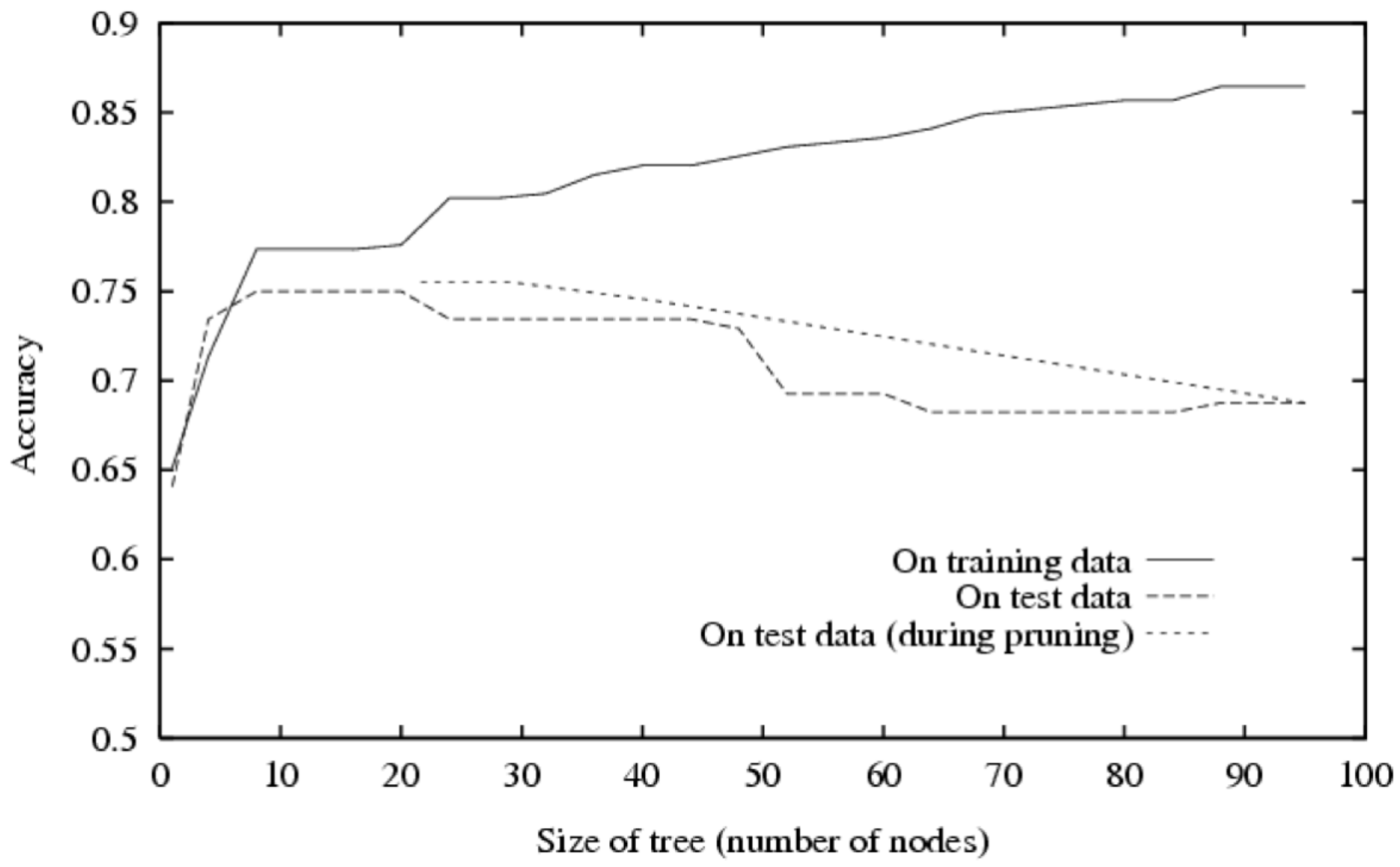


Figure courtesy of Tom Mitchell

Decision Tree 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

Poll Question 3:

What questions do you have?

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

Real-valued Features

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

Source: 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

Source: https://en.wikipedia.org/wiki/Iris_flower_data_set

Fisher Iris Dataset

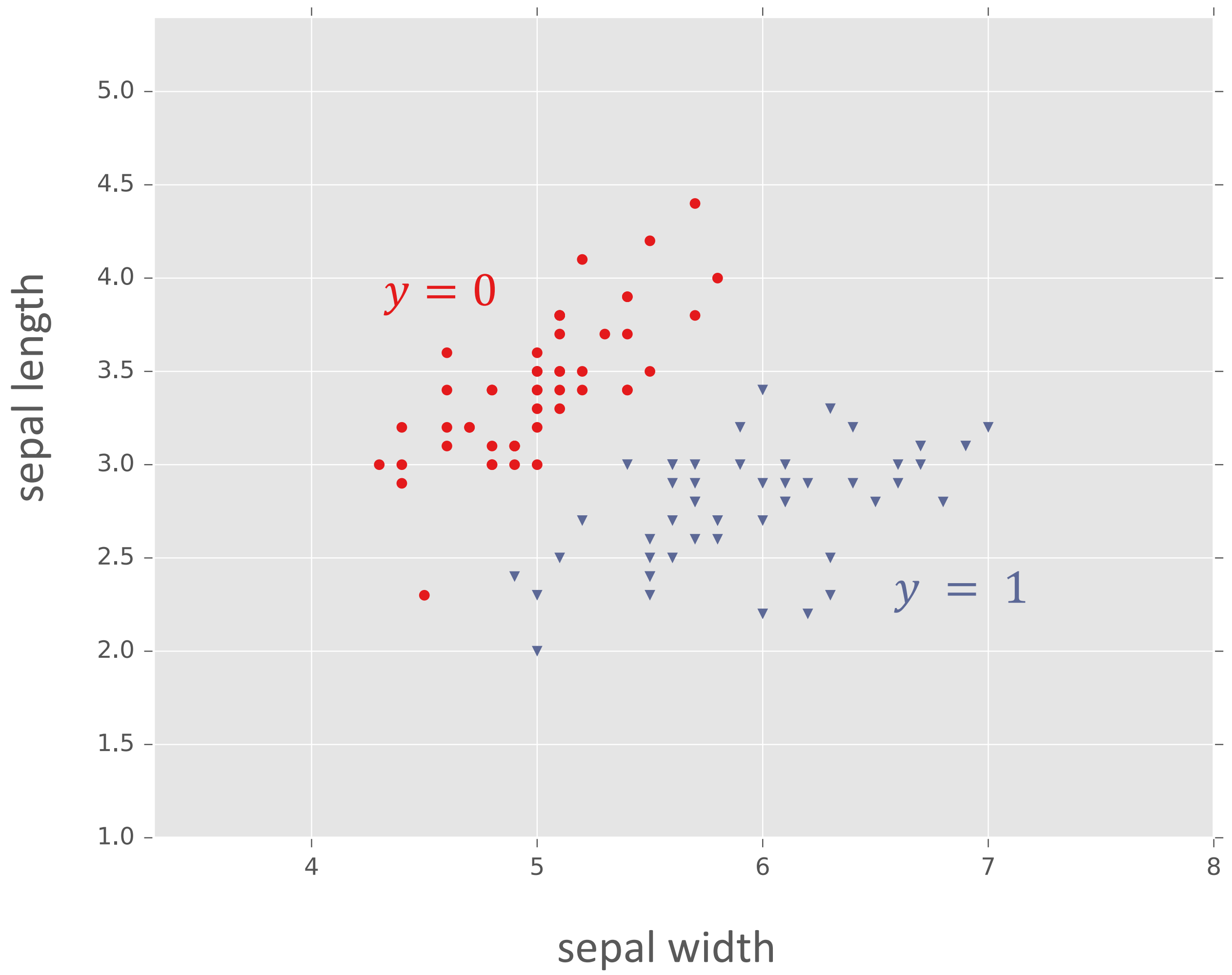
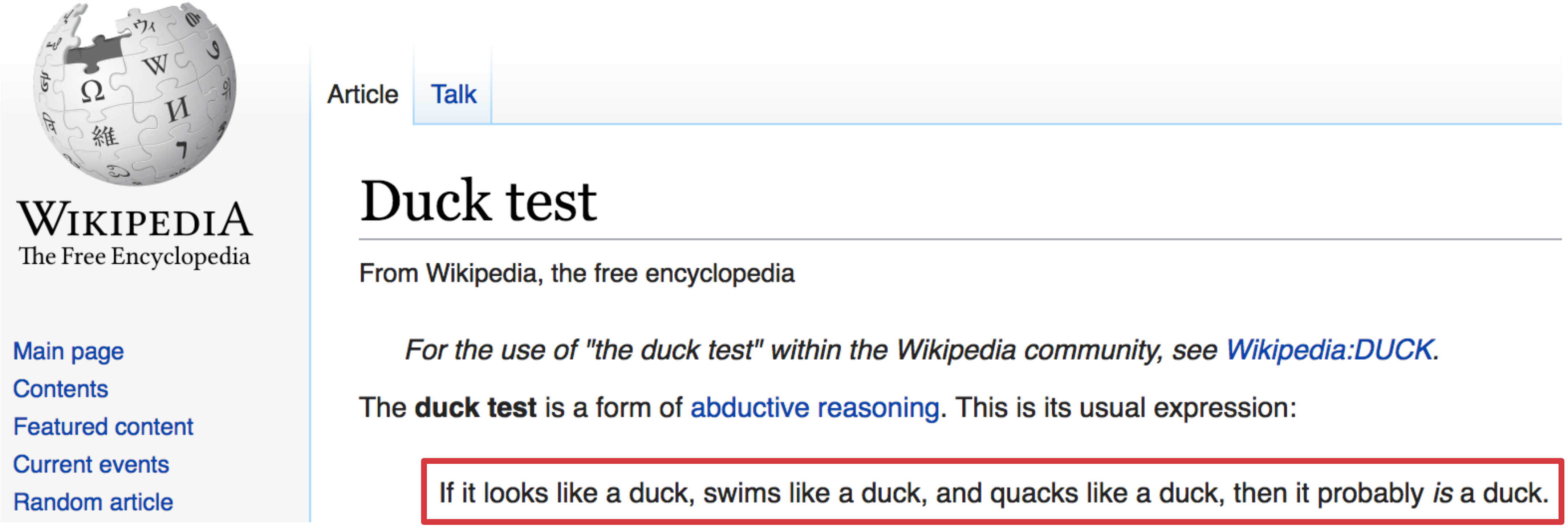


Figure courtesy of Matt Gormley


WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article

Article Talk

Duck test

From Wikipedia, the free encyclopedia

For the use of "the duck test" within the Wikipedia community, see Wikipedia:DUCK.

The **duck test** is a form of abductive reasoning. This is its usual expression:

If it looks like a duck, swims like a duck, and quacks like a duck, then it probably *is* a duck.

The Duck Test

Source: https://en.wikipedia.org/wiki/Duck_test

The Duck Test for Machine Learning

- Classify a point as the label of the "most similar" training point
- Idea: given real-valued features, we can use a distance metric to determine how similar two data points are
- A common choice is Euclidean distance:

$$d(\boldsymbol{x}, \boldsymbol{x}') = \|\boldsymbol{x} - \boldsymbol{x}'\|_2 = \sqrt{\sum_{d=1}^{D} (x_d - x'_d)^2}$$

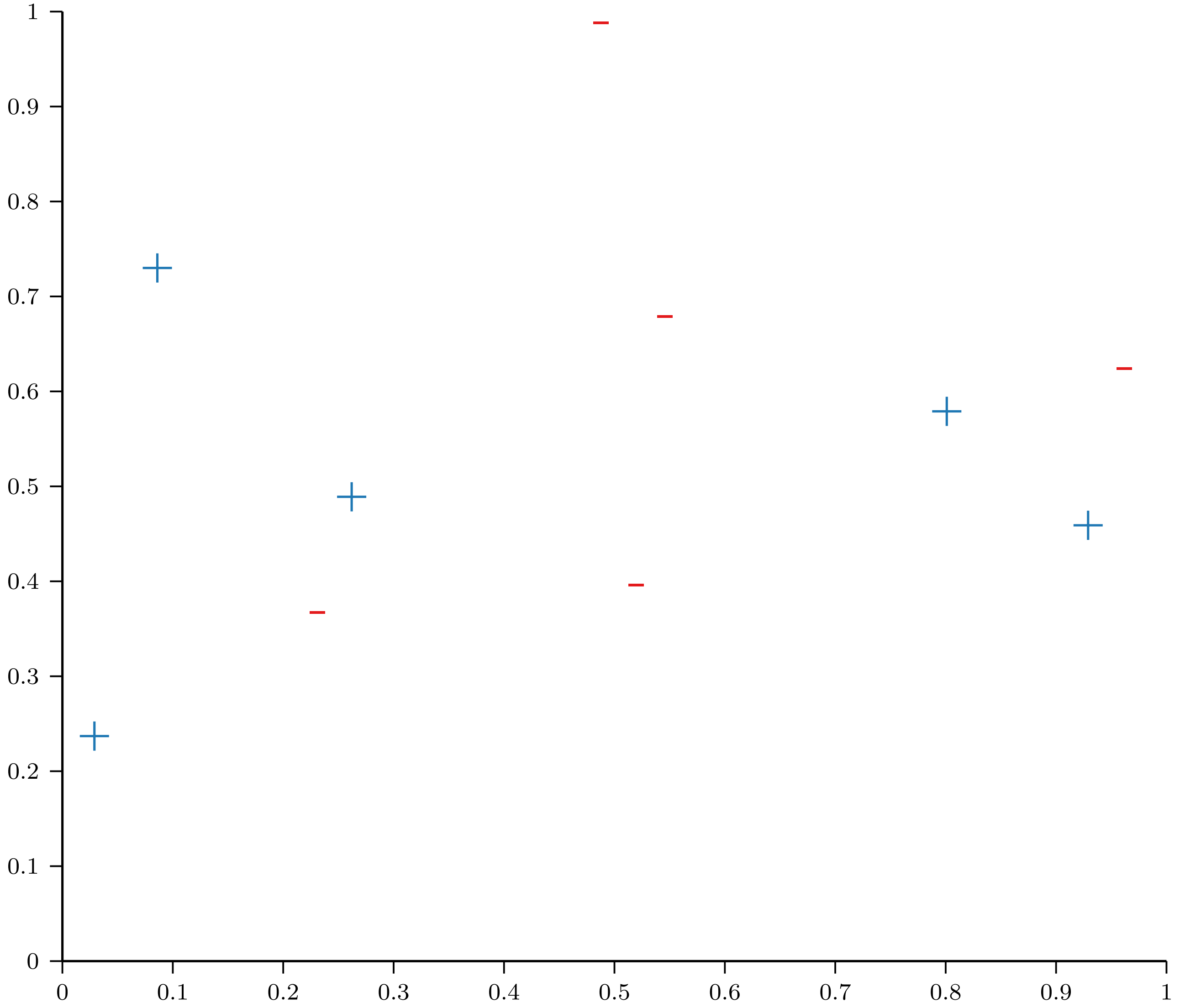
- An alternative is the Manhattan distance:

$$d(\boldsymbol{x}, \boldsymbol{x}') = \|\boldsymbol{x} - \boldsymbol{x}'\|_1 = \sum_{d=1}^{D} |x_d - x'_d|$$

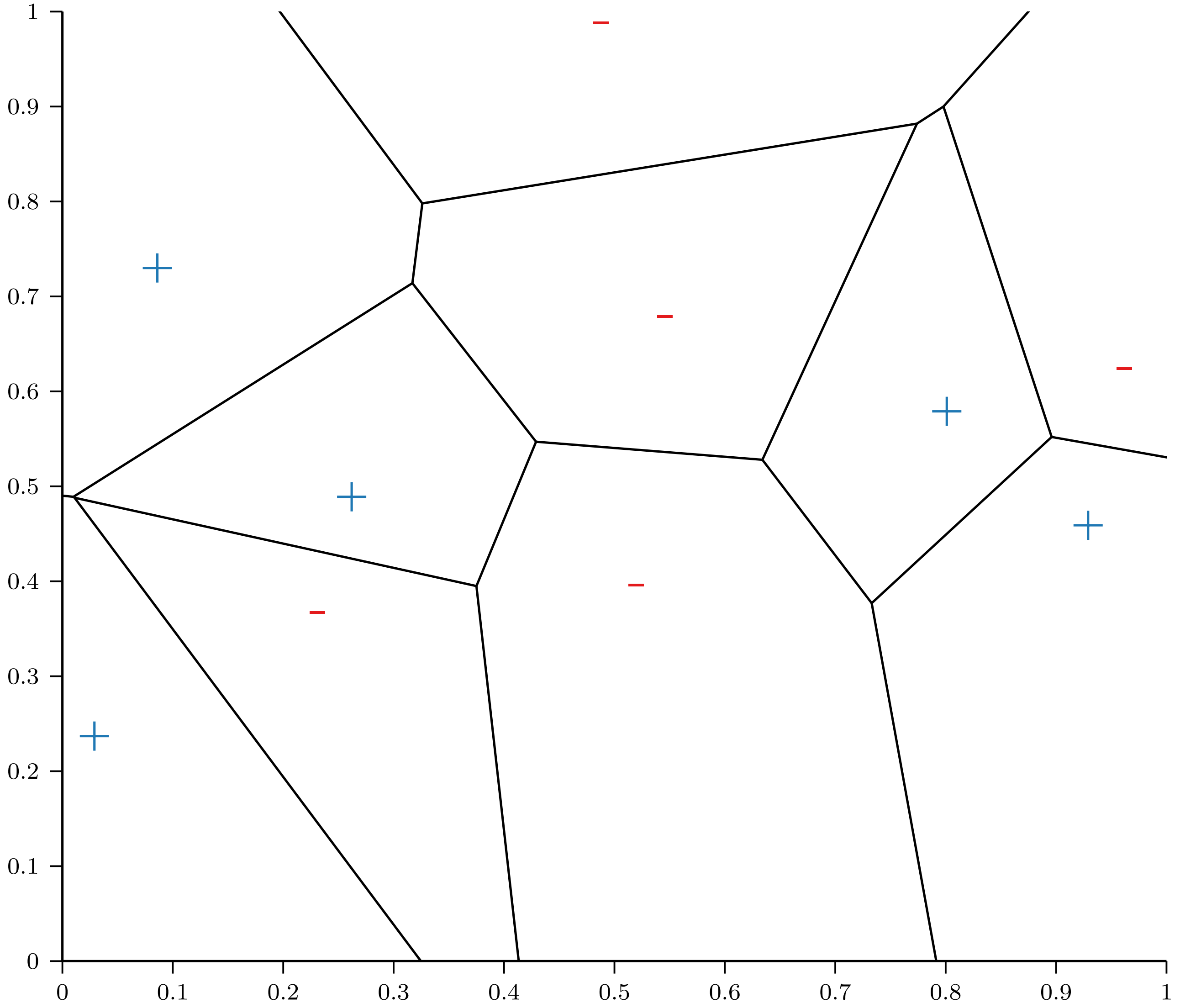
Nearest Neighbor: Pseudocode

```
def train(𝒟):
      store 𝒟
def predict(x′):
      find the nearest neighbor to x′ in 𝒟, x^(i)
      return y^(i)
```

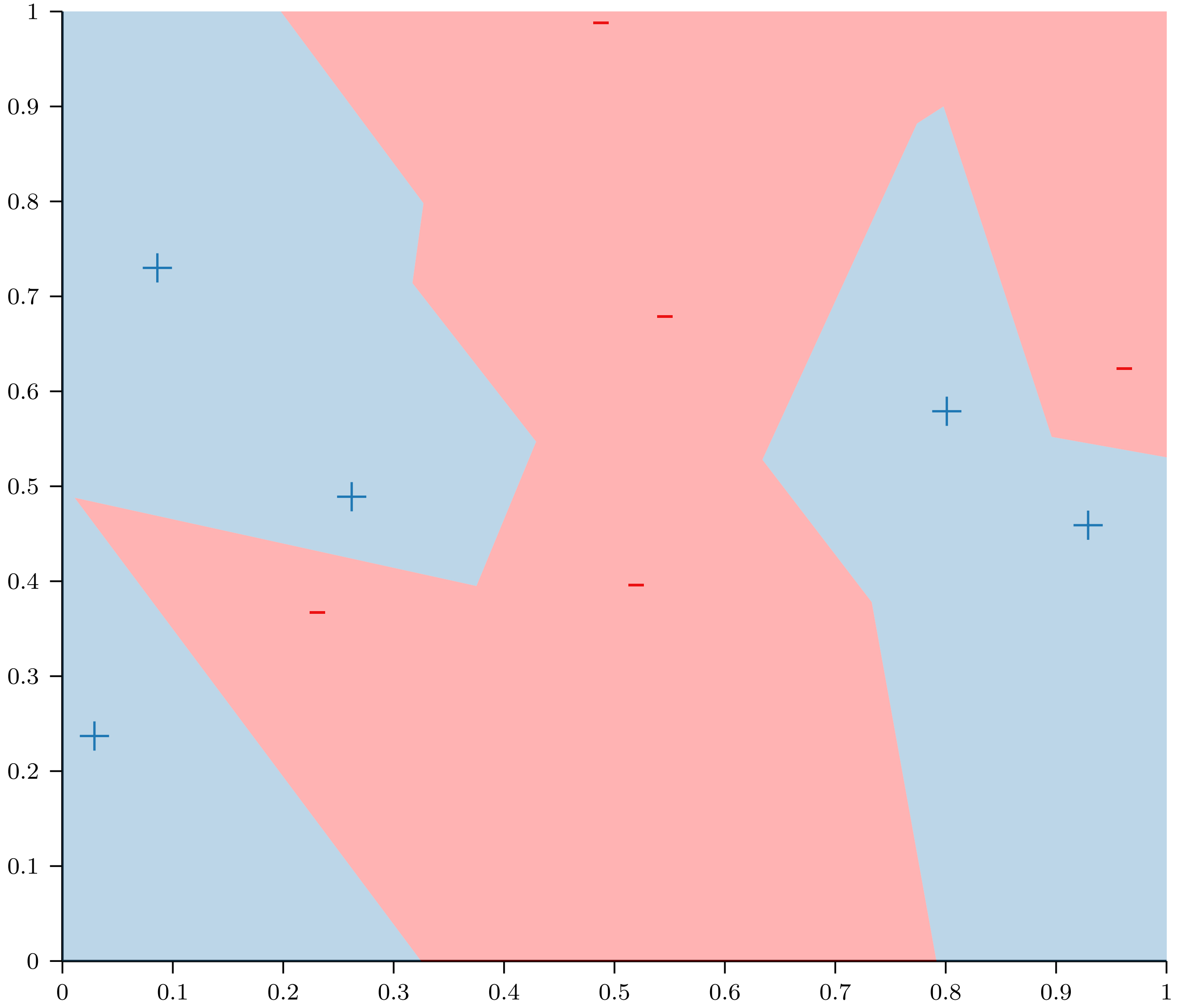
Nearest Neighbor: Example



Nearest Neighbor: Example



Nearest Neighbor: Example



The Nearest Neighbor Model

- Requires no training!
- Always has zero training error!
  - ***A data point is always its own nearest neighbor***

⋮

- Always has zero training error…