10-301/601: Introduction to Machine Learning Lecture 5 – KNNs

Henry Chai

5/14/25

Front Matter

- Announcements:
  - HW1 released on 5/13, due 5/16 at 11:59 PM
    - You will submit your homework to Gradescope
      1. Submit your code to the “programming” submission slot
      2. Submit a PDF with your answers to the questions “written” submission slot
         - **You must use LaTeX to typeset your responses!**

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

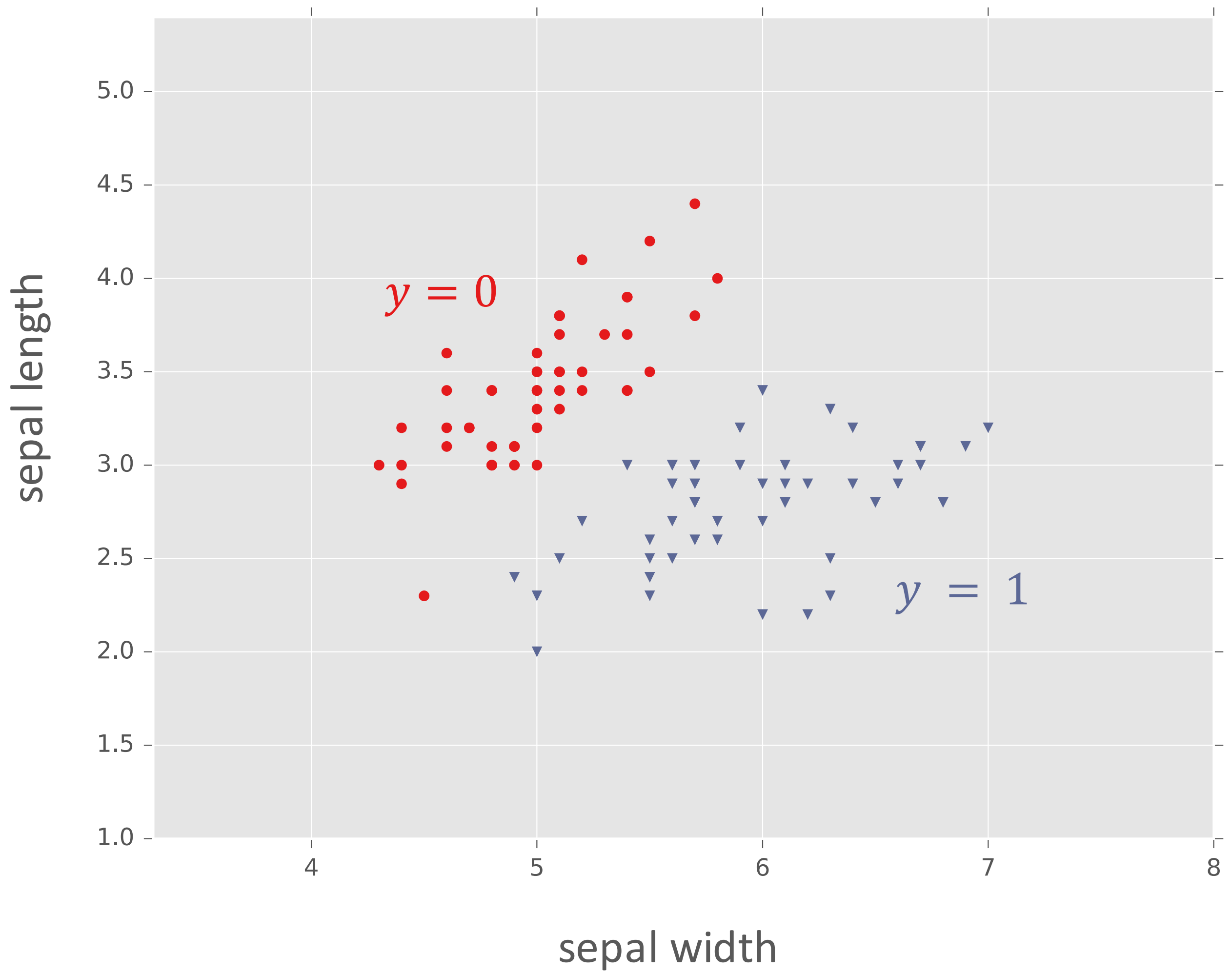
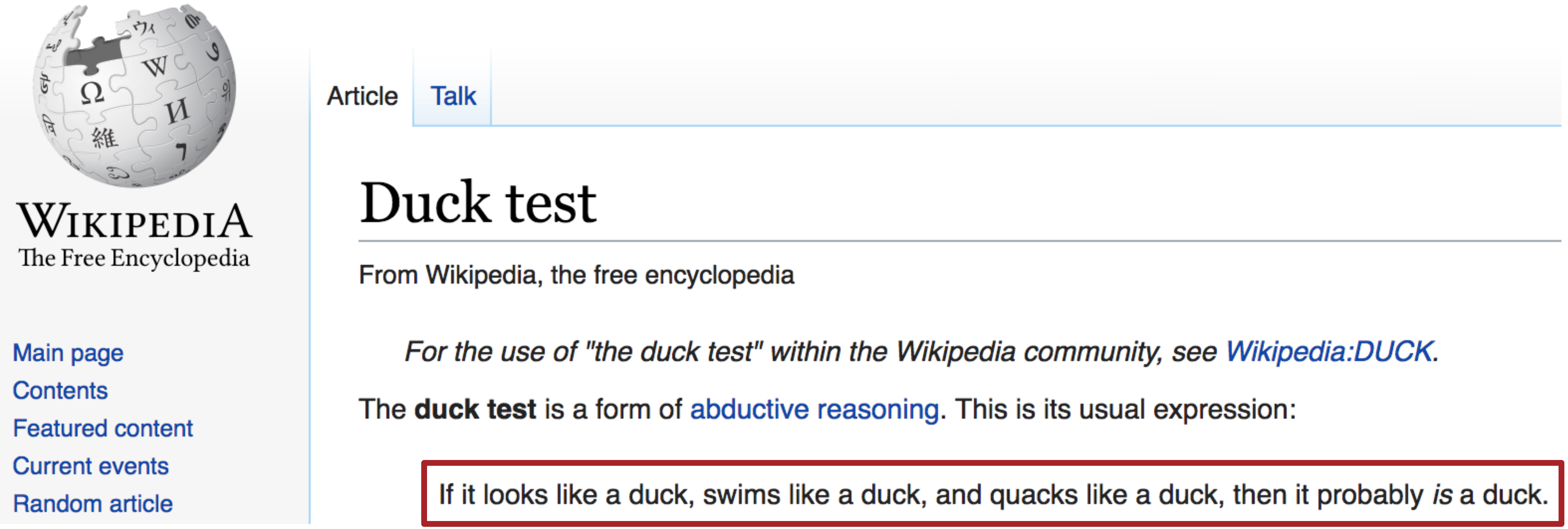


Figure courtesy of Matt Gormley



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:

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

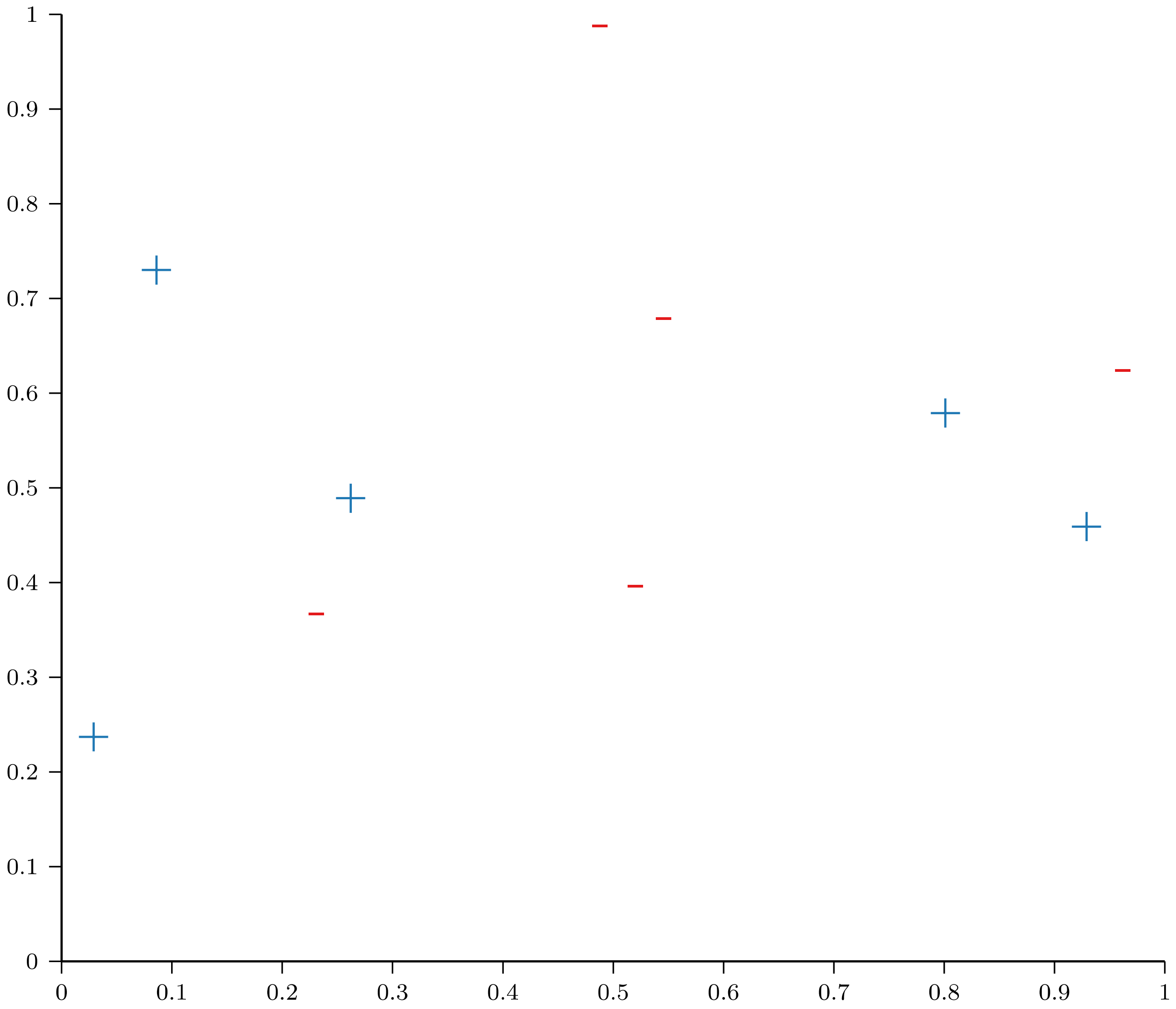
- An alternative is the Manhattan distance:

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

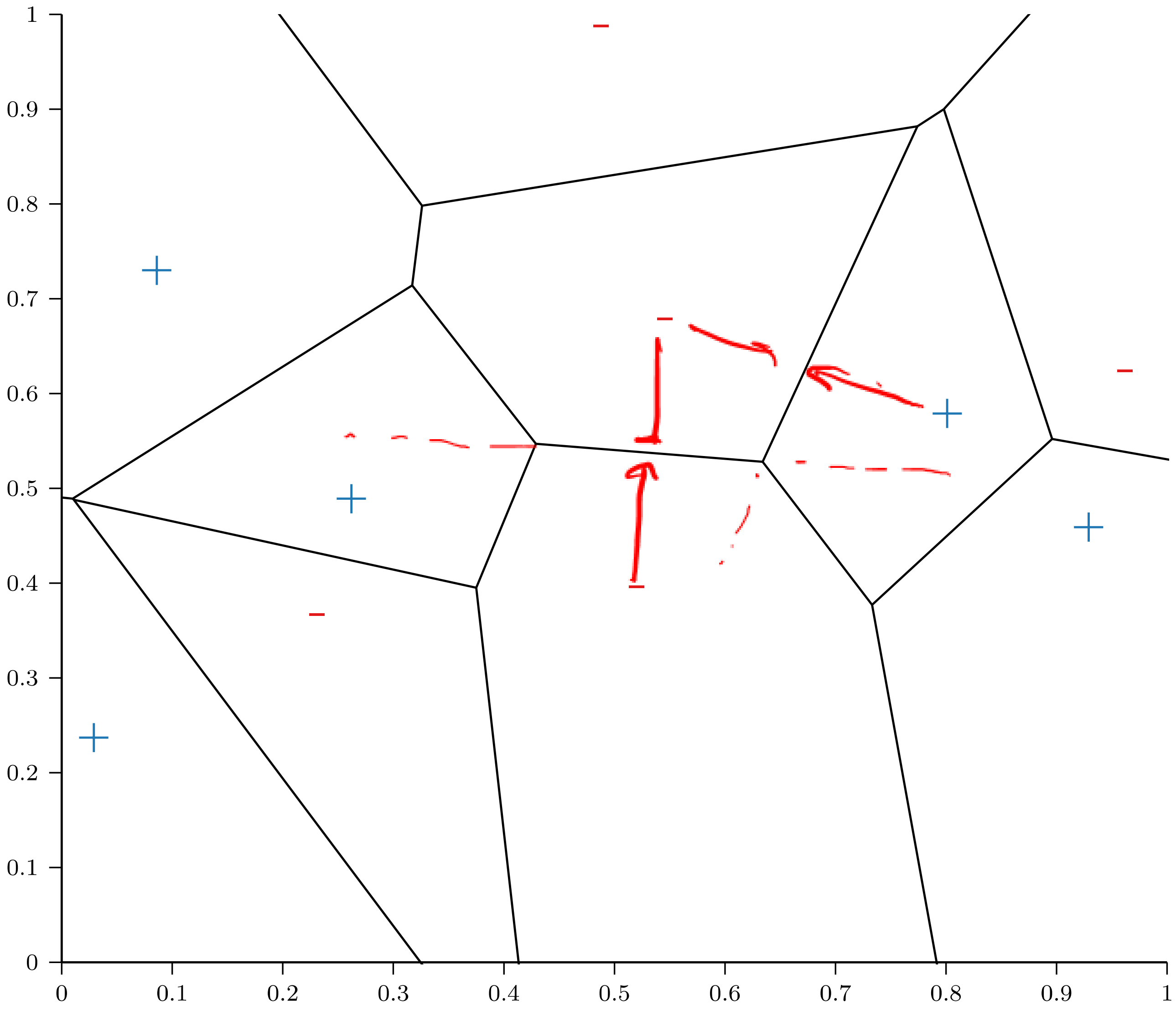
Nearest Neighbor: Pseudocode

```
def train(𝒟):
    Store D
def predict(x′):
    find the closest data point to x⃗′ in D
             x⃗(i)
    return y(i)
```

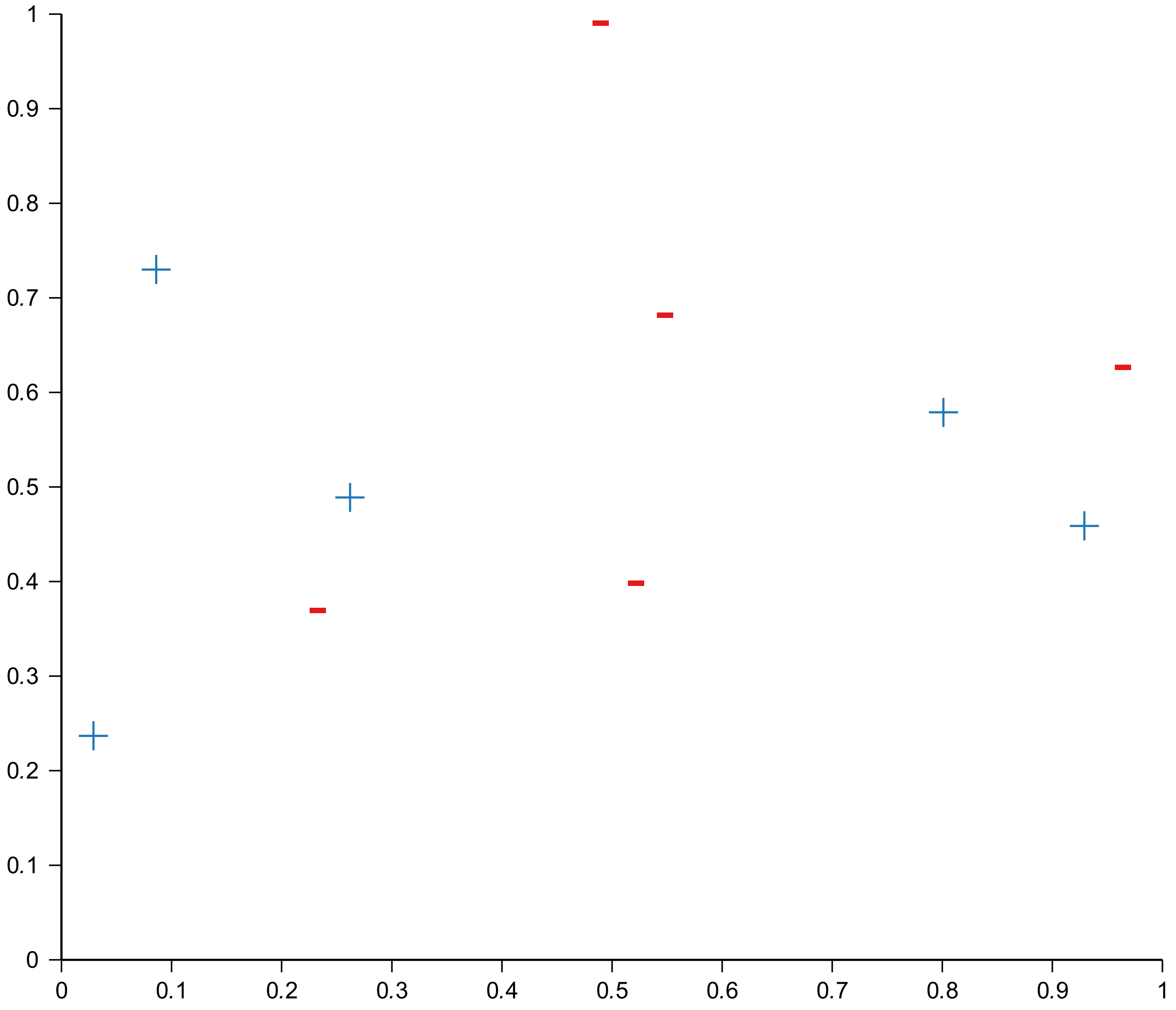
Nearest Neighbor: Example



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***

$\vdots$

- Always has zero training error…

Generalization of Nearest Neighbor (Cover and Hart, 1967)

- Claim: under certain conditions, as $N \rightarrow \infty$, with high probability, the true error rate of the nearest neighbor model $\leq 2 \ *$ the Bayes error rate (the optimal classifier)
- Interpretation: “In this sense, it may be said that half the classification information in an infinite sample set is contained in the nearest neighbor.”

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1053964

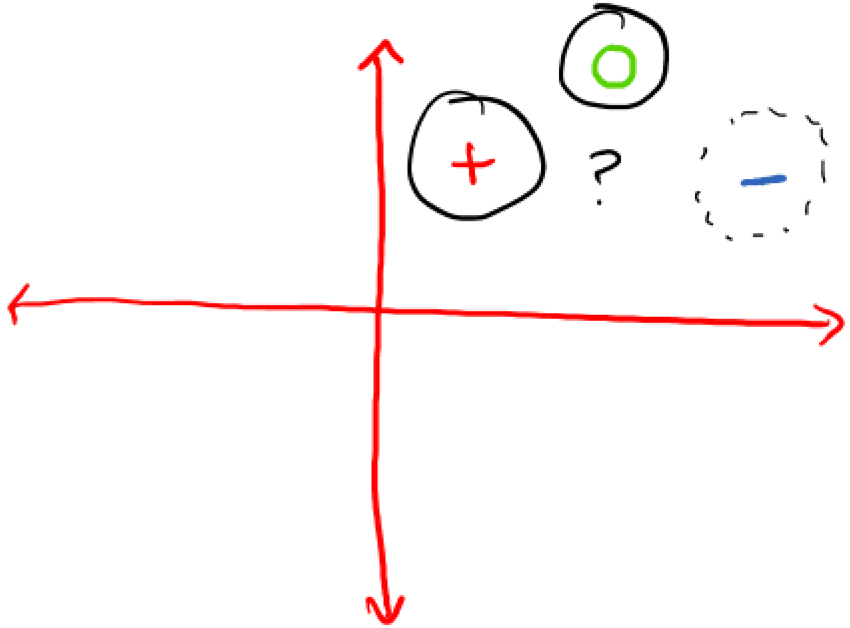
But why limit ourselves to just one neighbor?

- Claim: under certain conditions, as $N \to \infty$, with high probability, the true error rate of the nearest neighbor model $\leq 2 \ *$ the Bayes error rate (the optimal classifier)
- Interpretation: "In this sense, it may be said that half the classification information in an infinite sample set is contained in the nearest neighbor."

Source: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1053964

k-Nearest Neighbors (kNN)

- Classify a point as the most common label among the labels of the k nearest training points
- Tie-breaking (in case of even k and/or more than 2 classes)



Lecture 5 + 6 Polls

0 surveys completed

0 surveys underway

Suppose you have a kNN model with $k > 1$ and 3 possible classes. Which of the following tie-breaking methods is *guaranteed* to break a tie in the majority vote? Select all that apply

- Weight the votes by distance
- Remove the furthest neighbor
- Add another neighbor
- Use a different distance metric
- None of the above

k-Nearest Neighbors (kNN): Pseudocode

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

    store D

def predict(x'):

    return majority_vote(labels of the k nearest neighbors to $\vec{x}'$ in D)

label tie-breaking

distance tie-breaking

kNN on Fisher Iris Data

Figure courtesy of Matt Gormley

kNN on Fisher Iris Data

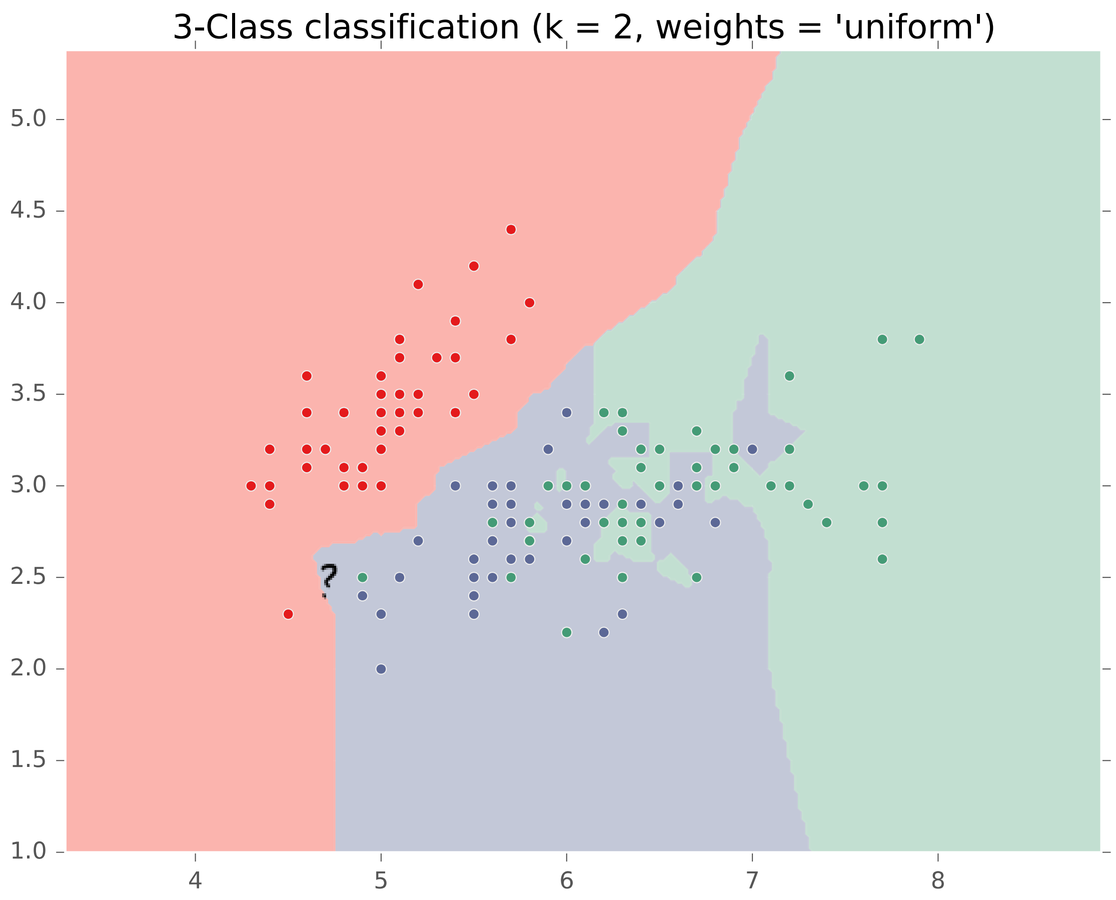


Figure courtesy of Matt Gormley

kNN on Fisher Iris Data

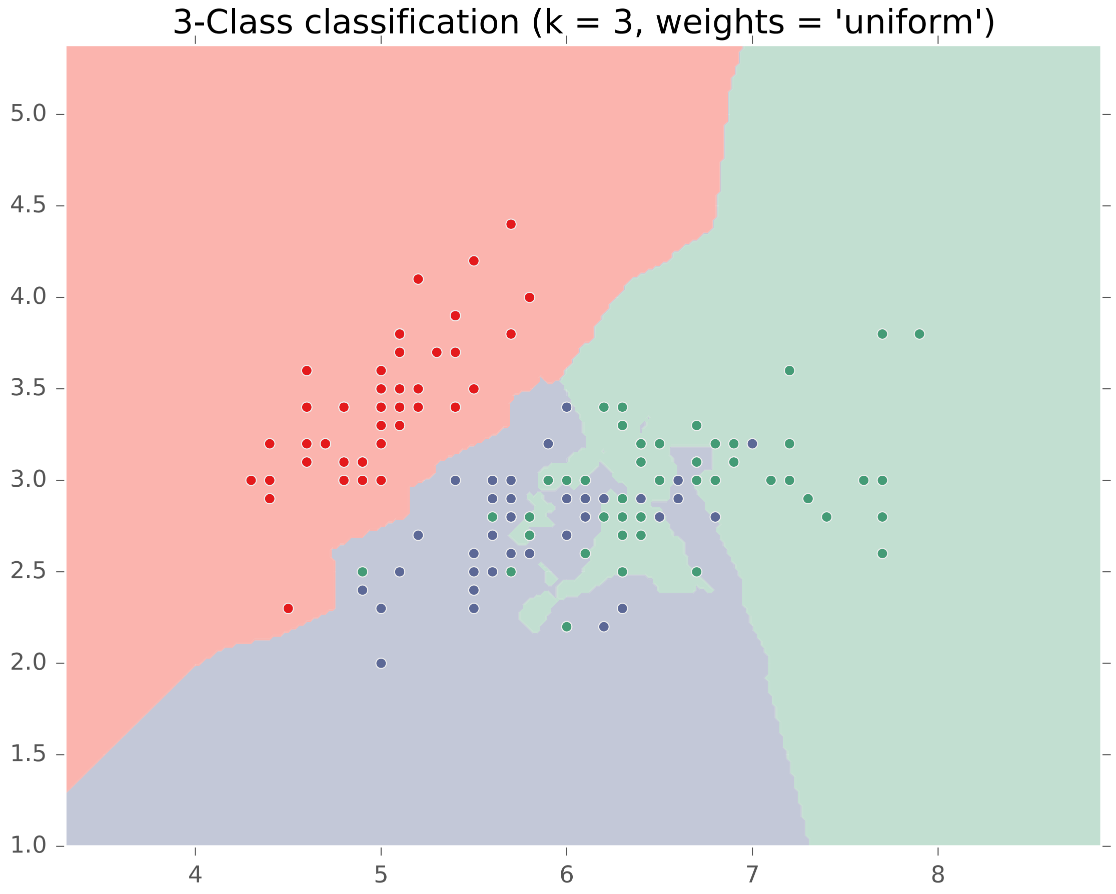


Figure courtesy of Matt Gormley

kNN on Fisher Iris Data

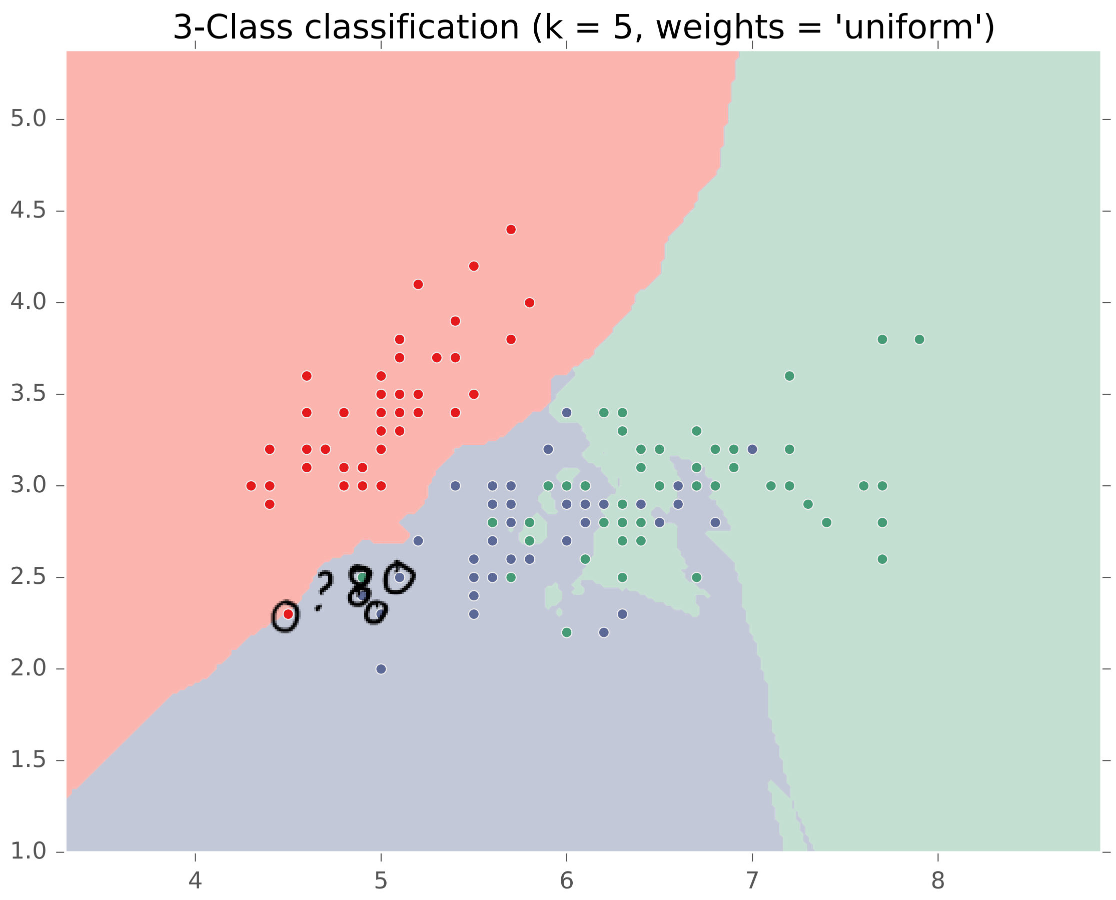


Figure courtesy of Matt Gormley

kNN on Fisher Iris Data

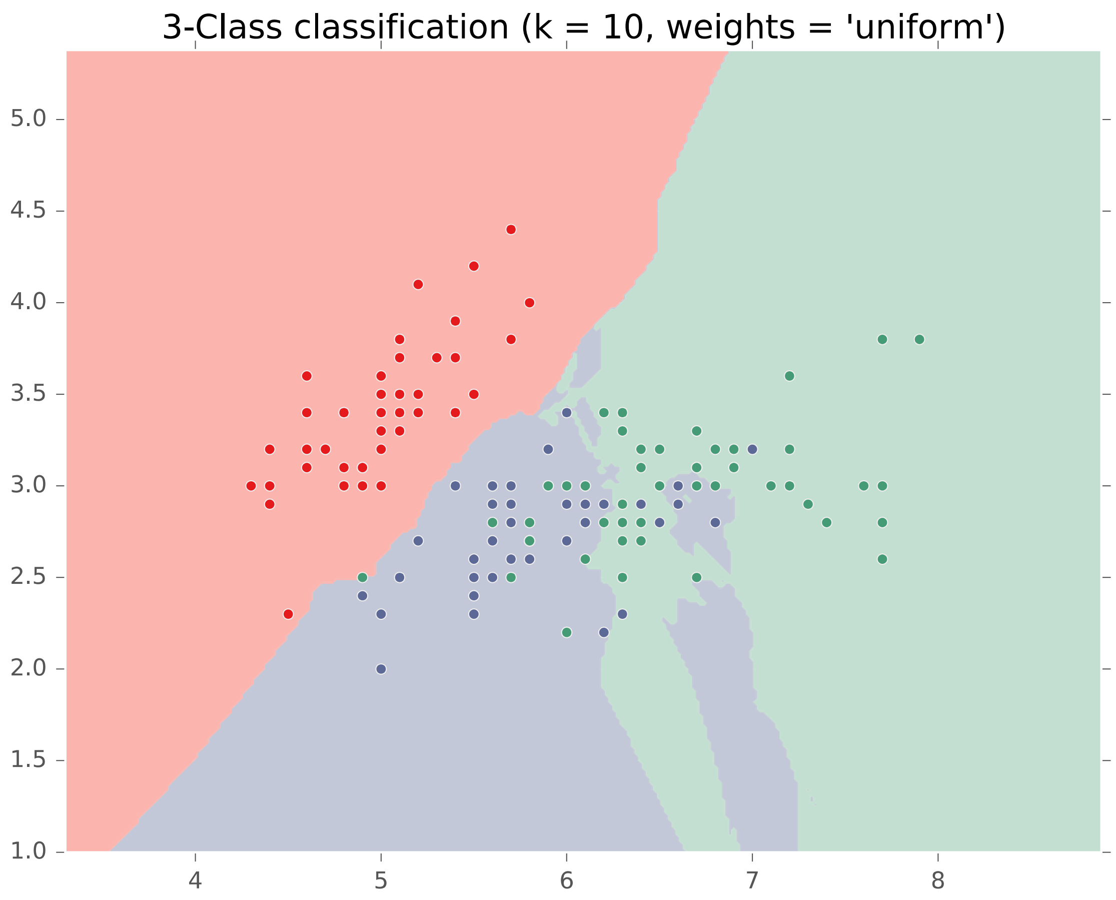


Figure courtesy of Matt Gormley

kNN on Fisher Iris Data

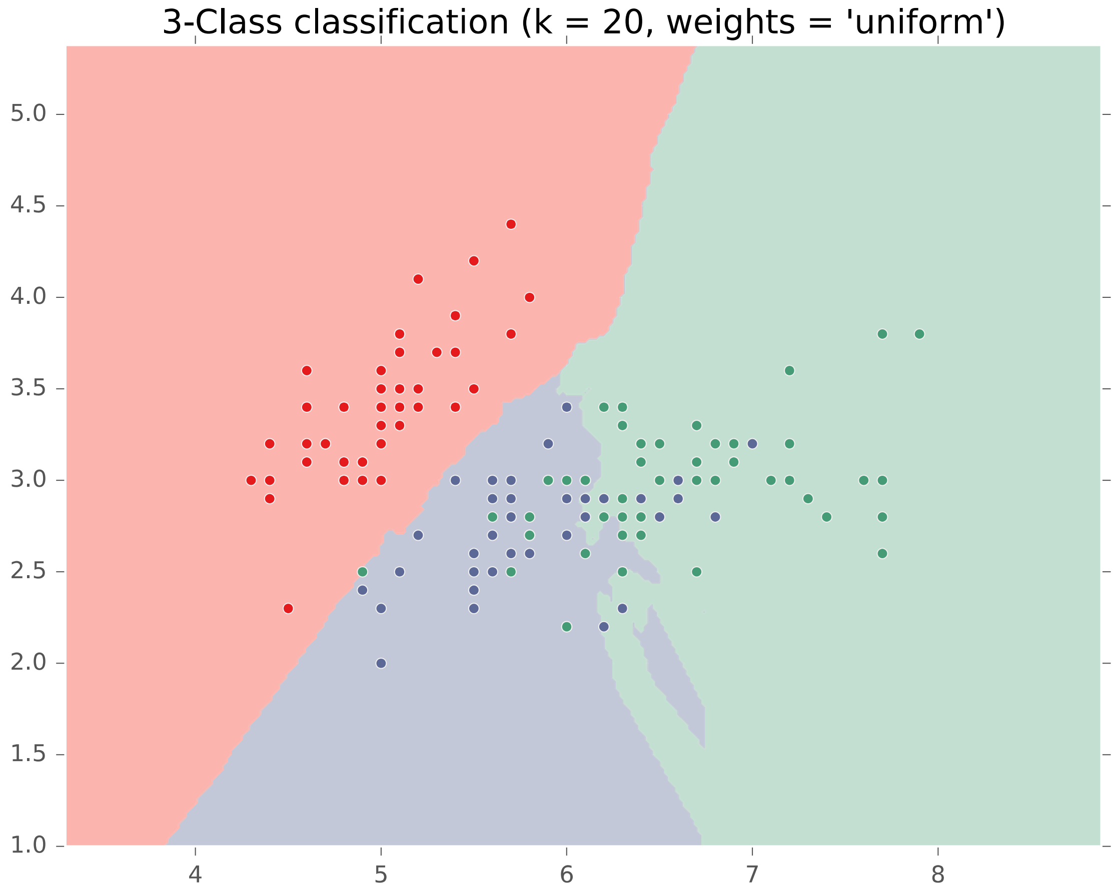


Figure courtesy of Matt Gormley

kNN on Fisher Iris Data

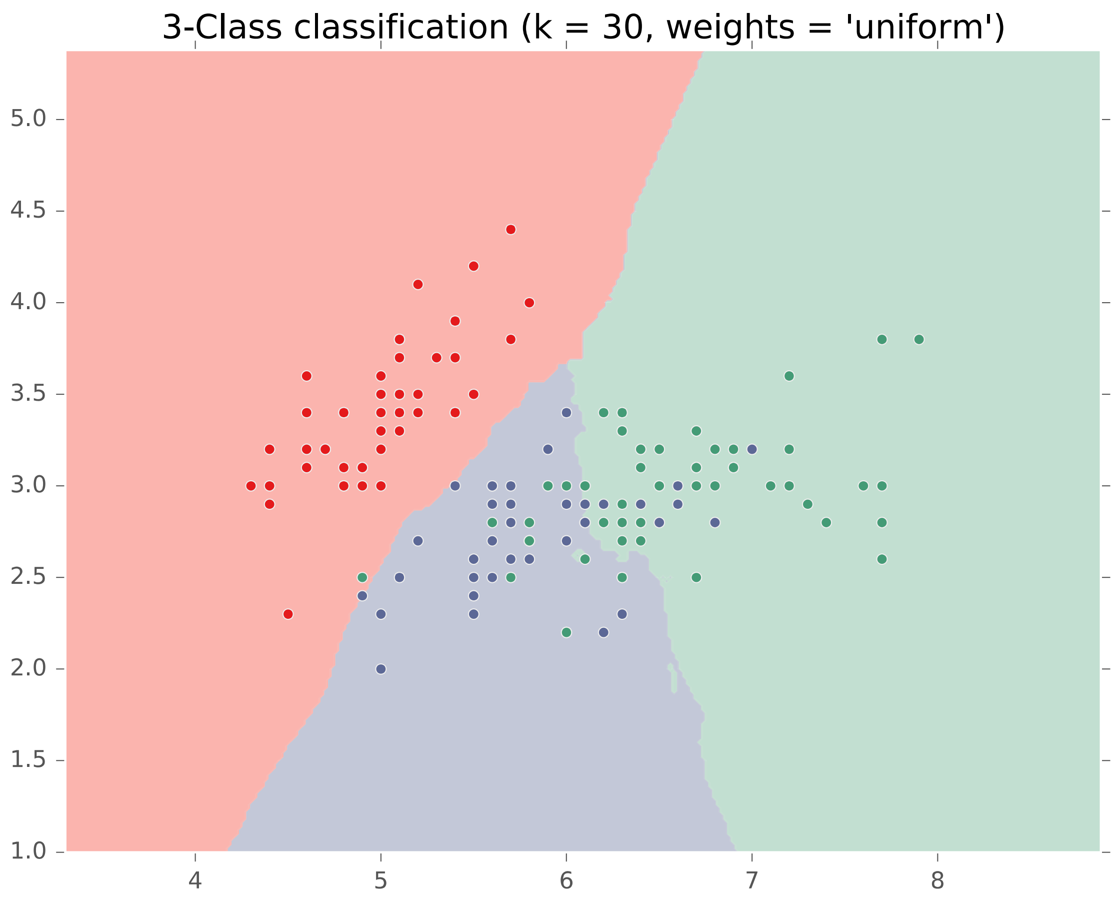


Figure courtesy of Matt Gormley

kNN on Fisher Iris Data

Figure courtesy of Matt Gormley

kNN on Fisher Iris Data

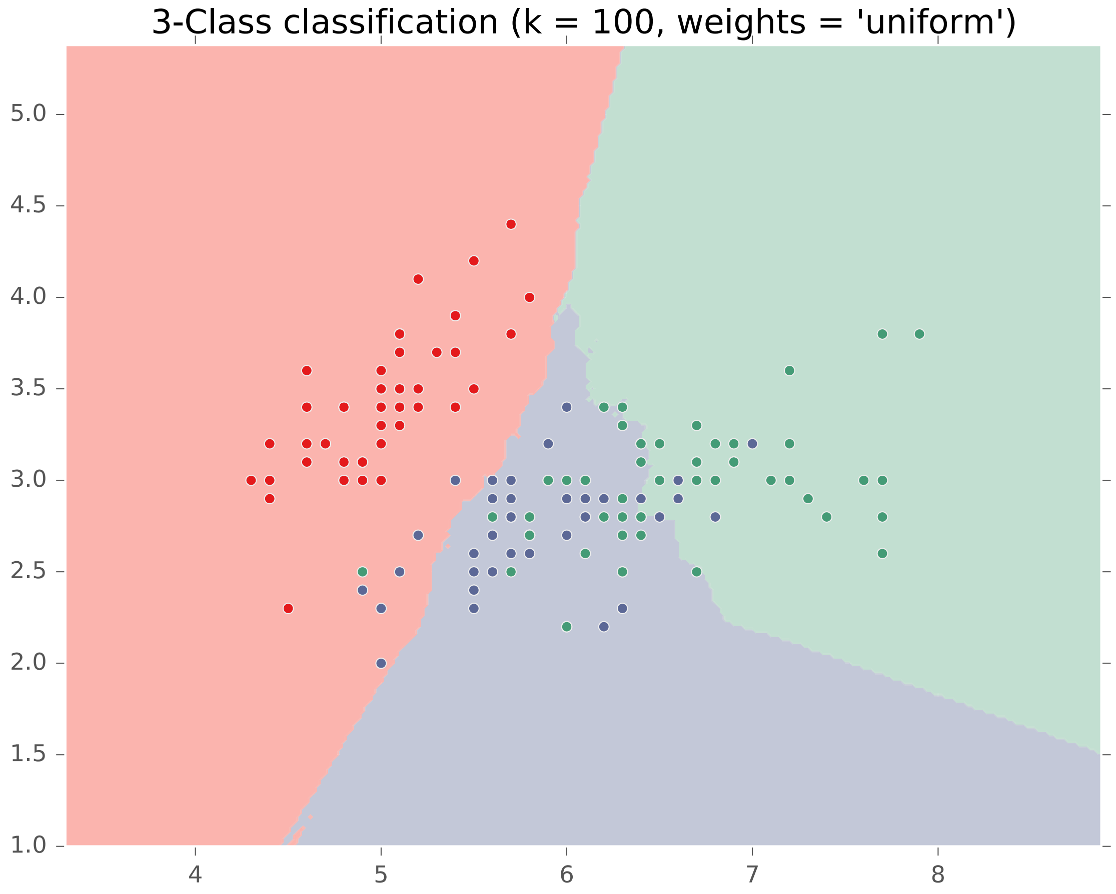


Figure courtesy of Matt Gormley

kNN on Fisher Iris Data

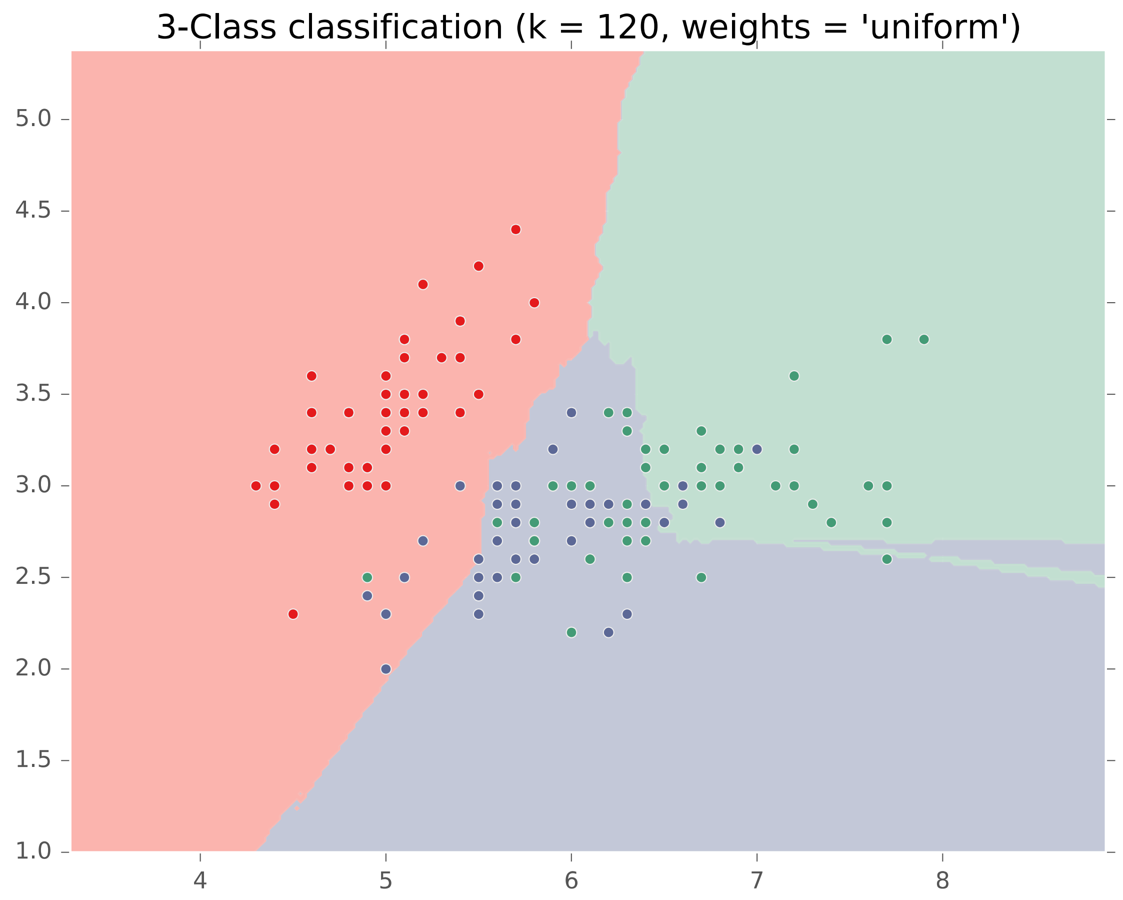


Figure courtesy of Matt Gormley

kNN on Fisher Iris Data

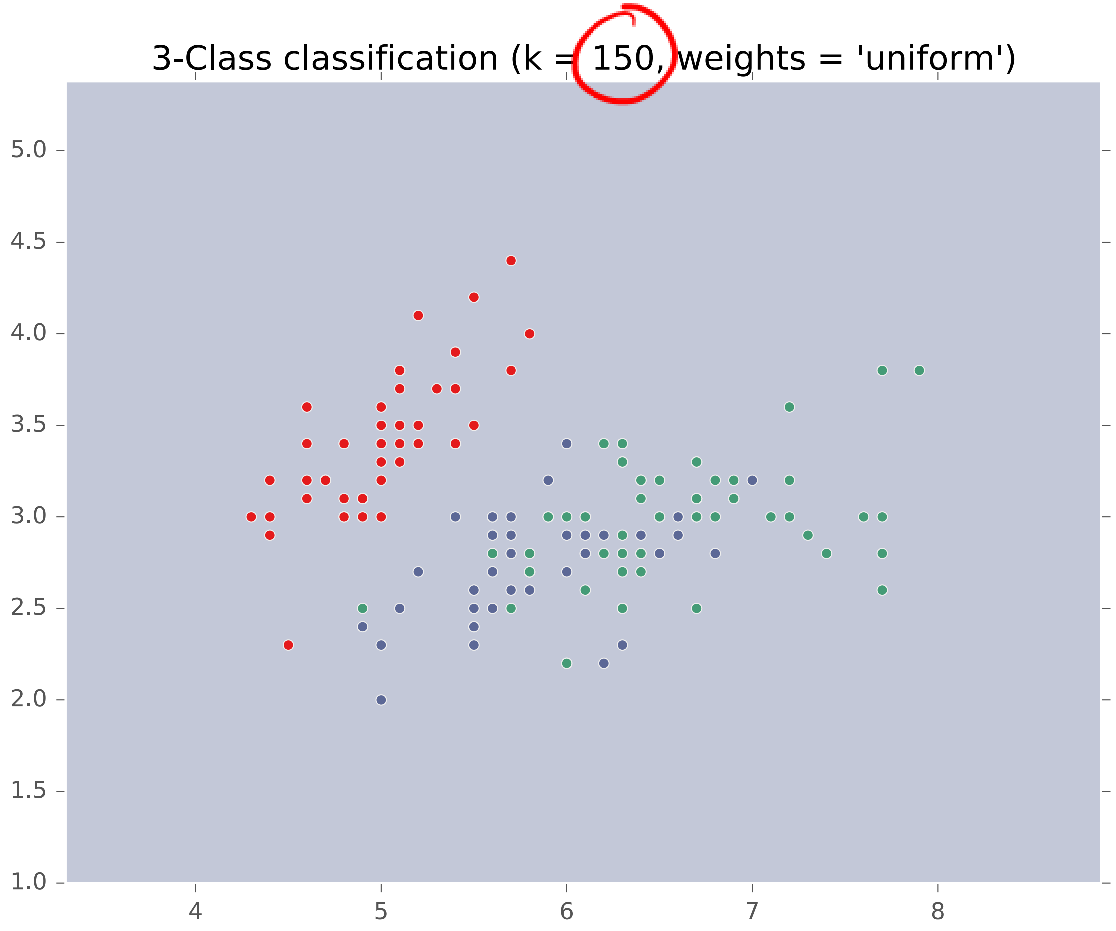


Figure courtesy of Matt Gormley

Setting k

- When $k = 1$:
  - many, complicated decision boundaries
  - may overfit
- When $k = N$:
  - no decision boundaries; always predicts the most common label in the training data
  - may underfit
- k controls the complexity of the hypothesis set $\Longrightarrow k$ affects how well the learned hypothesis will generalize

kNN and Categorical Features

- kNNs are compatible with categorical features, either by:

1. Converting categorical features into binary ones:

Cholesterol
Normal
Normal
Abnormal

→

Normal Cholesterol?	Abnormal Cholesterol?
1	0
1	0
0	1

2. Using a distance metric that works over categorical features e.g., the Hamming distance:

$$d(\boldsymbol{x}, \boldsymbol{x}') = \sum_{d=1}^{D} \mathbb{1}(x_d \neq x'_d)$$

kNN: Inductive Bias

Key Takeaways

- Real-valued features and decision boundaries
- Nearest neighbor model and generalization guarantees
- kNN “training” and prediction
- Effect of k on model complexity
- kNN inductive bias