MSc - Data Mining

Topic 06 : Classification

Part 04 : K Nearest Neighbours

Dr Bernard Butler and Dr Kieran Murphy

Department of Computing and Mathematics, WIT.

(bbutler@tssg.org; kmurphy@wit.ie)

Spring Semester, 2021

Outline

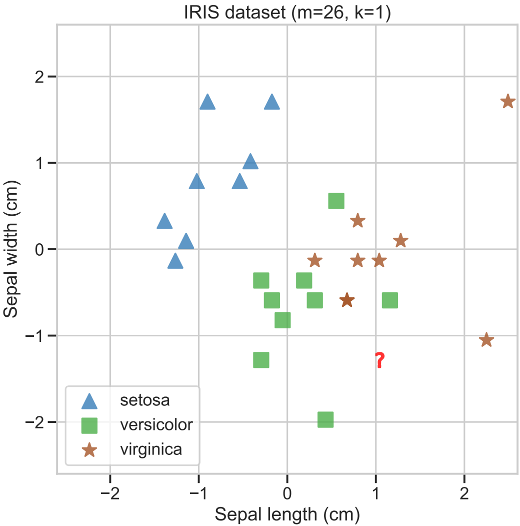
- K Nearest Neighbours (KNN) algorithm

k-Nearest Neighbour Methods

General Idea

- Given m labeled observations (▲, ■, and ★), how should we classify a new unlabelled observation (**?**)?
- We could use the labels of the k-nearest neighbouring points.
  - "Nearest" means distance — how should we calculate this?
  - How do we pick the value for k?
  - What is our decision rule?

  Assign new observation to most frequent occurring class in k-nearest neighbours.
  - What to do if there is a tie?

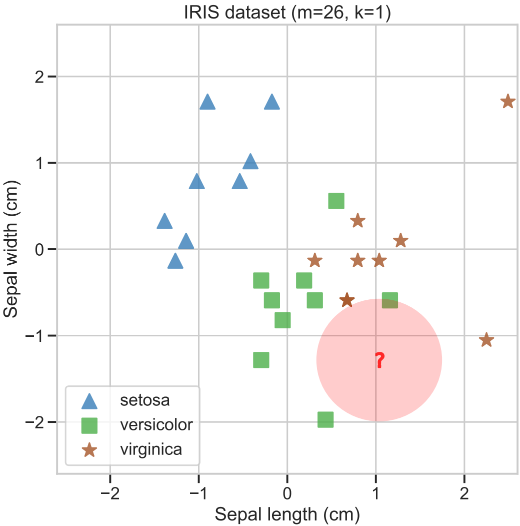


k-Nearest Neighbour Methods

General Idea

- Given m labeled observations (▲, ■, and ★), how should we classify a new unlabelled observation (?)?
- We could use the labels of the k-nearest neighbouring points.
  - “Nearest” means distance — how should we calculate this?
  - How do we pick the value for k?
  - What is our decision rule?

  Assign new observation to most frequent occurring class in k-nearest neighbours.
  - What to do if there is a tie?



Distance Functions

We frequently want to measure how close/near/similar two points (think observations/instances/cases) are. For this we need a distance function.

Distance Function

A distance function, $D(a,b)$, is any function that satisfies the properties:

non-negativity:	$D(a,b) \geq 0$, distance between any two points is non-negative and is only zero if $a = b$.
symmetric:	$D(a,b) = D(b,a)$
triangular inequality:	$D(a,c) \leq D(a,b) + D(b,c)$

Distance Functions

We frequently want to measure how close/near/similar two points (think observations/instances/cases) are. For this we need a distance function.

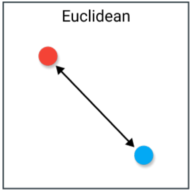
Distance Function

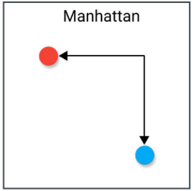
A distance function, $D(a, b)$, is any function that satisfies the properties:

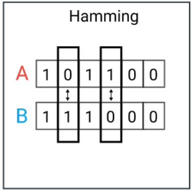
non-negativity: $D(a, b) \geq 0$, distance between any two points is non-negative and is only zero if $a = b$.

symmetric: $D(a, b) = D(b, a)$

triangular inequality: $D(a, c) \leq D(a, b) + D(b, c)$







Distance Functions

We frequently want to measure how close/near/similar two points (think observations/instances/cases) are. For this we need a distance function.

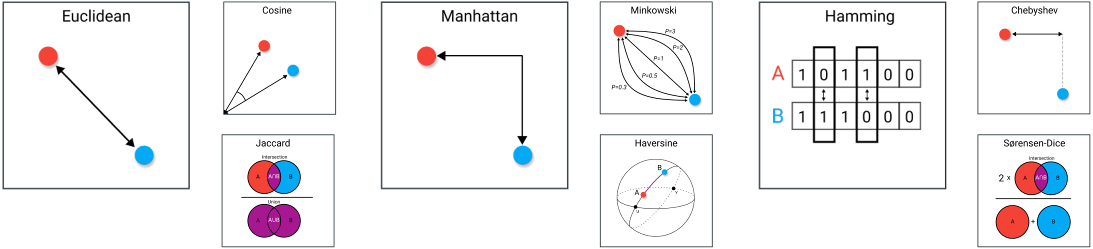
Distance Function

A distance function, $D(a, b)$, is any function that satisfies the properties:

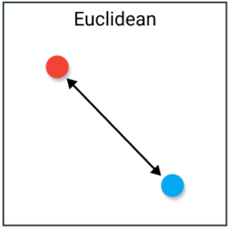
non-negativity: $D(a, b) \geq 0$, distance between any two points is non-negative and is only zero if $a = b$.

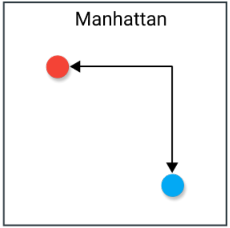
symmetric: $D(a, b) = D(b, a)$

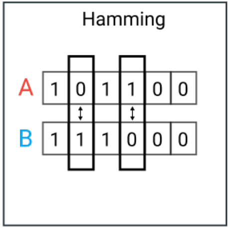
triangular inequality: $D(a, c) \leq D(a, b) + D(b, c)$



Distance Functions — Euclidean, Manhattan, and Hamming







- Pythagorean theorem

$$D(\boldsymbol{a}, \boldsymbol{b}) = \sqrt{\sum_{i=1}^{n} \left[a^{(i)} - b^{(i)}\right]^2}$$

- "As the crow flies"
- Features should be normalised before use

✔ Most commonly used metric.

✘ Becomes less useful for large dimensions

- Taxi-cab distance

$$D(\boldsymbol{a}, \boldsymbol{b}) = \sum_{i=1}^{n} \left|a^{(i)} - b^{(i)}\right|$$

✔ Seems to work better than Euclidean for high-dimensional data

✔ Suitable for datasets with discrete and/or binary features.

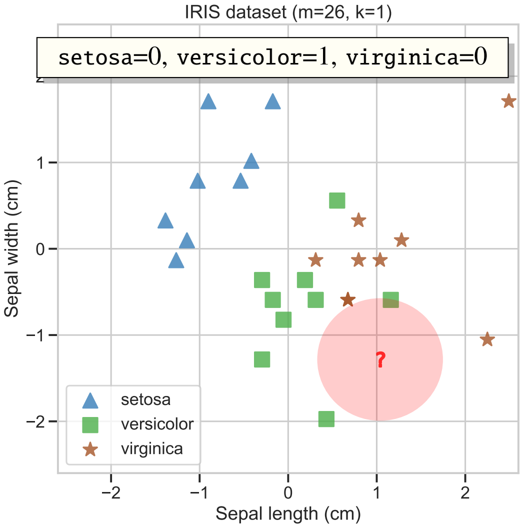
- Count of the number of differences (bits/letters/levels etc) between two points.

✔ Can be used between categorical variables.

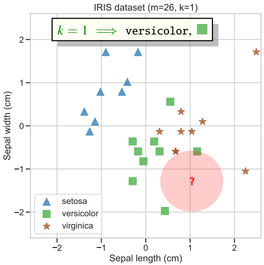
✘ Difficult to use when two vectors are not of equal length.

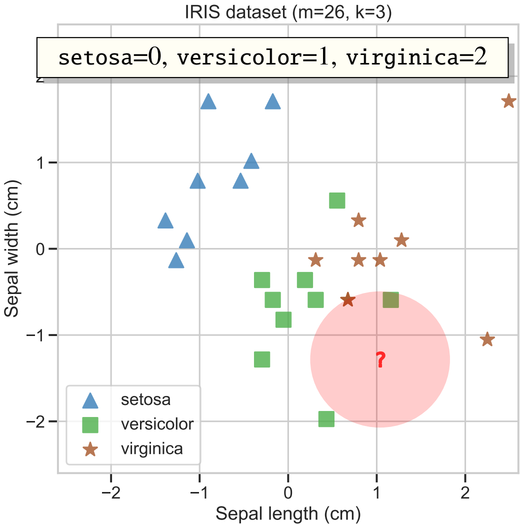
✘ Should not be used when magnitude is important.

Effect of k

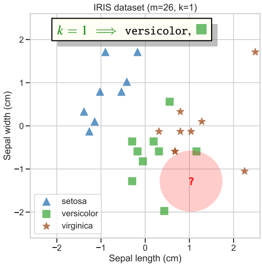

IRIS dataset (m=26, k=1)
setosa=0, versicolor=1, virginica=0
Sepal width (cm)
Sepal length (cm)
?
setosa
versicolor
virginica

Effect of k

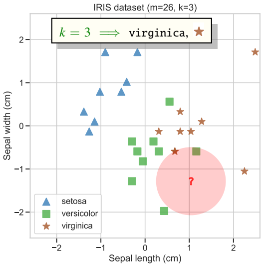

IRIS dataset (m=26, k=1)
$k = 1 \implies$ versicolor,
Sepal width (cm)
Sepal length (cm)
setosa
versicolor
virginica
?


IRIS dataset (m=26, k=3)
setosa=0, versicolor=1, virginica=2
Sepal width (cm)
Sepal length (cm)
setosa
versicolor
virginica
?

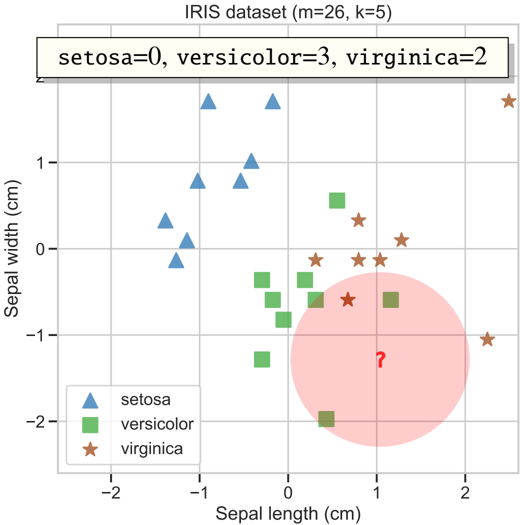
Effect of k



IRIS dataset (m=26, k=1)
$k = 1 \implies$ versicolor,
Sepal width (cm)
Sepal length (cm)
setosa
versicolor
virginica

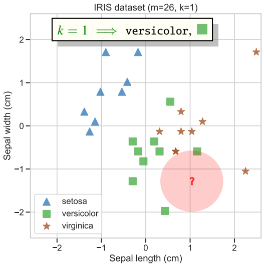


IRIS dataset (m=26, k=3)
$k = 3 \implies$ virginica,
Sepal width (cm)
Sepal length (cm)
setosa
versicolor
virginica

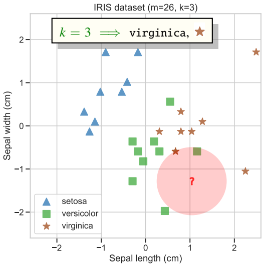


IRIS dataset (m=26, k=5)
setosa=0, versicolor=3, virginica=2
Sepal width (cm)
Sepal length (cm)
setosa
versicolor
virginica

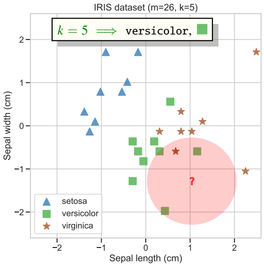
Effect of k



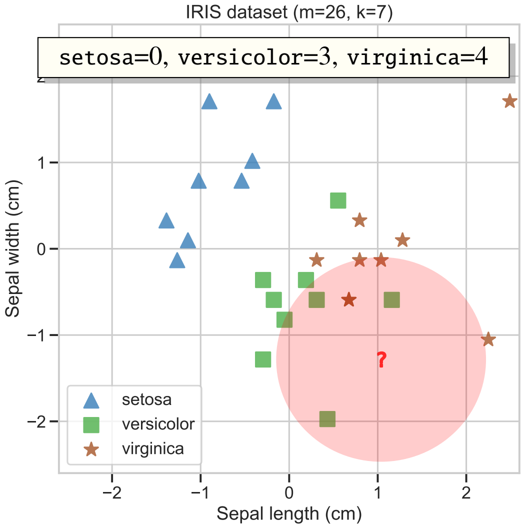
IRIS dataset (m=26, k=1)
k = 1 ⟹ versicolor,
setosa
versicolor
virginica
?
Sepal width (cm)
Sepal length (cm)



IRIS dataset (m=26, k=3)
k = 3 ⟹ virginica,
setosa
versicolor
virginica
?
Sepal width (cm)
Sepal length (cm)

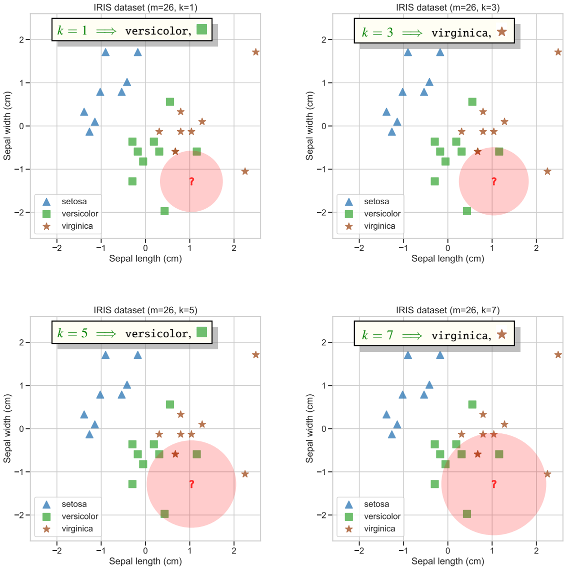


IRIS dataset (m=26, k=5)
k = 5 ⟹ versicolor,
setosa
versicolor
virginica
?
Sepal width (cm)
Sepal length (cm)



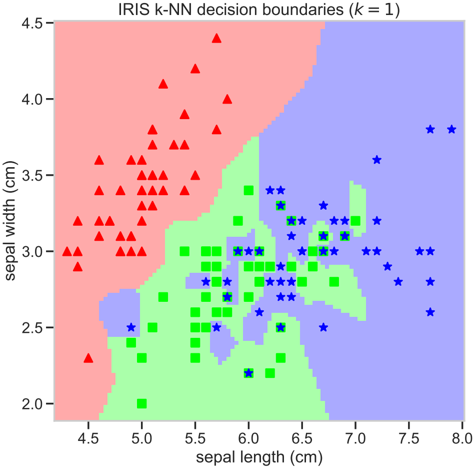
IRIS dataset (m=26, k=7)
setosa=0, versicolor=3, virginica=4
setosa
versicolor
virginica
?
Sepal width (cm)
Sepal length (cm)

Effect of k

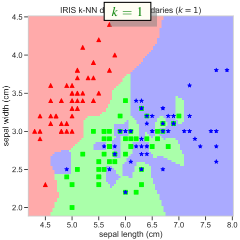


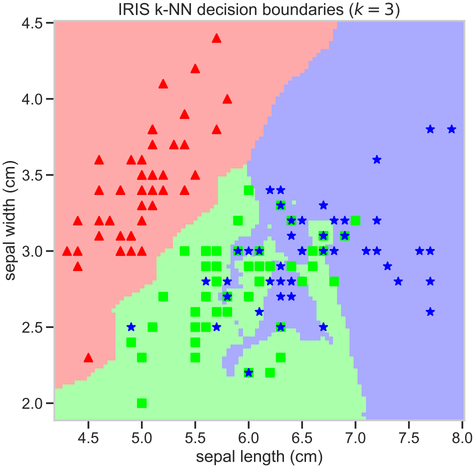
- k in range in $1, \ldots, \sqrt{m}$,
- $k = \sqrt{m}$ is often optimal.
- To reduce probability of a tie, pick k so that is is not a multiple of the number of classes (here 3).
- General rule to resolve ties is to reduce k by one. (Note $k = 1$ will never tie.)
- A small value of k means that noise will have a higher influence on the result and a large value makes prediction computationally expensive.
- Small $k \Rightarrow$ high variance, large $k \Rightarrow$ high bias.

Effect of k on Decision Boundary

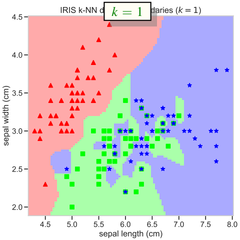


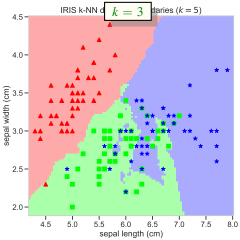
Effect of k on Decision Boundary

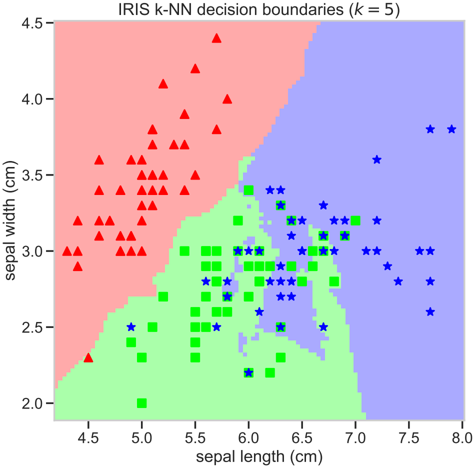




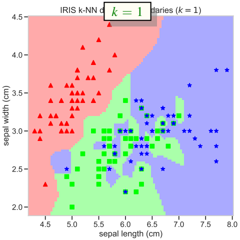
Effect of k on Decision Boundary

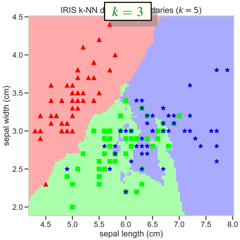


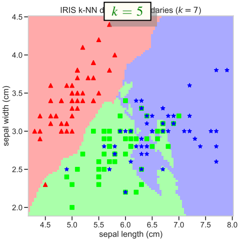


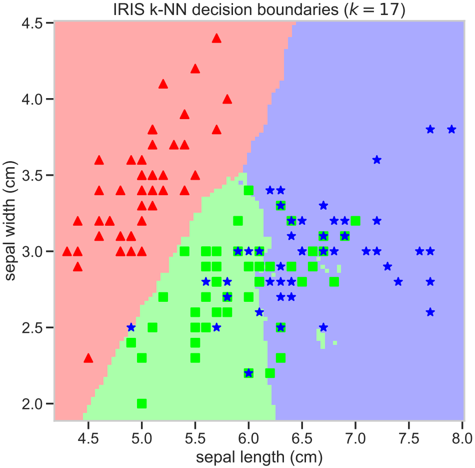


Effect of k on Decision Boundary

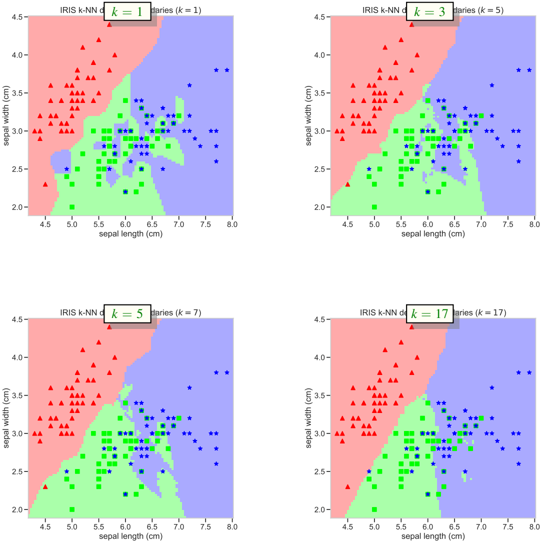








Effect of k on Decision Boundary



- A small value of k means that noise will have a higher influence on the result and a large value makes prediction computationally expensive.
  $\Longrightarrow$ rougher decision boundaries
- A large value of k means more observations are included in the decision so noise is averaged out
  $\Longrightarrow$ smoother decision boundaries
- Small $k \Rightarrow$ high variance, large $k \Rightarrow$ high bias.

k-Nearest Neighbour Methods — Review

When to Consider

- Observations/instances map to points in $\mathbb{R}^n$ (quantitative/numerical features)
- Less than 20 features/attributes per instance (low dimensionality)
- Lots of training data (more points means closer neighbours)

Advantages

- Training is very fast (instantaneous, since lazy learner)
- Learn complex target functions
- Do not lose information (lazy learner)

Disadvantages

- Slow at query time (uses training data not model to predict)
- Memory-based technique (must pass over (nearly) all points for each classification)
- Easily fooled by irrelevant features/attributes

Hyper-Parameters

- Distance metric (Euclidean — “as the crow flies”)
- Number of neighbours, k (Increasing k reduces variance, increases bias)

Outline

1. Resources

Resources

- 9 Distance Measures in Data Science
  `towardsdatascience.com/9-distance-measures-in-data-science-918109d069fa`
  Non-technical comparison of common distances functions (source of images used here).