QUEEN'S UNIVERSITY BELFAST
SCHOOL OF ELECTRONICS, ELECTRICAL ENGNIEERING AND COMPUTER SCIENCE

Data Analysis & Visualisation

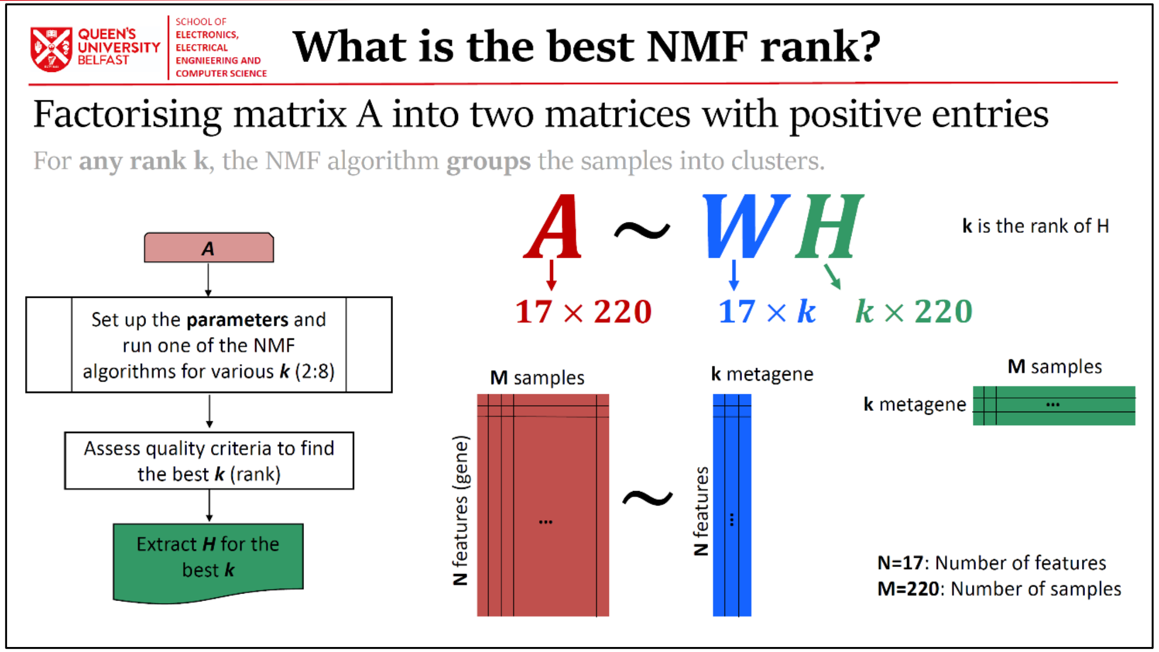
CSC3062

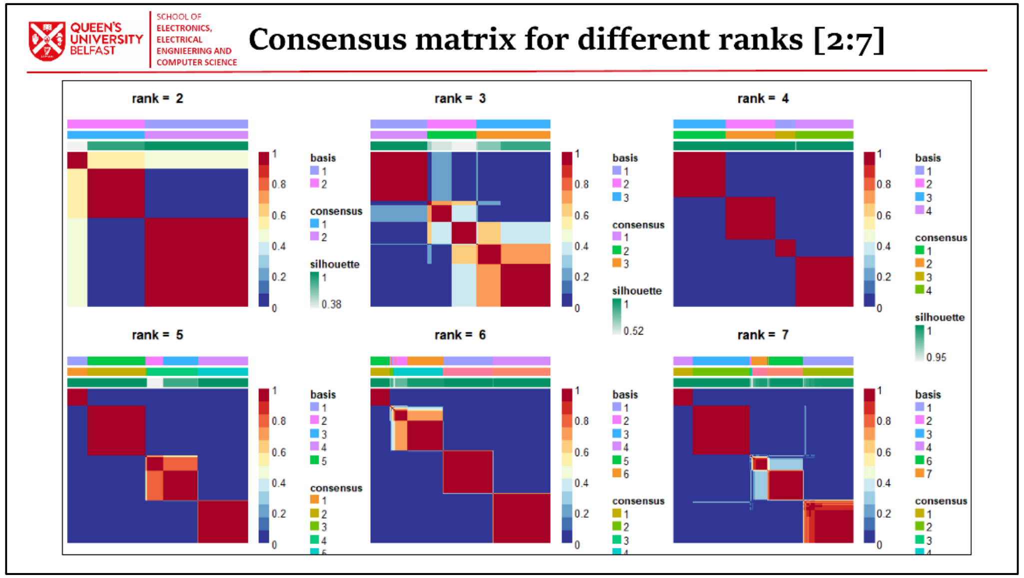
BEng (CS & SE), MEng (CS & SE), BIT & CIT

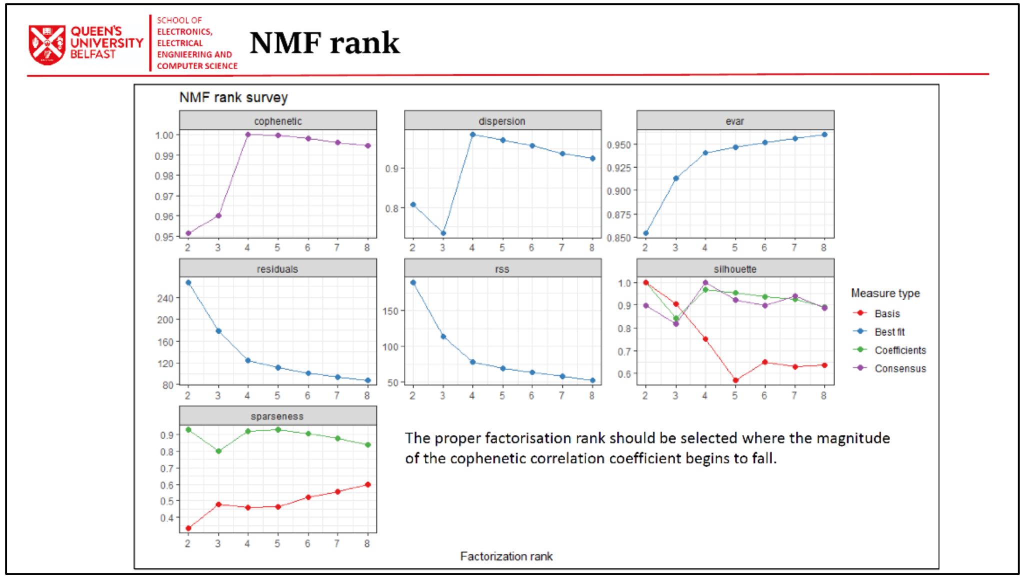
Dr Reza Rafiee

Semester 1 2019

Non-negative matrix factorisation (NMF)







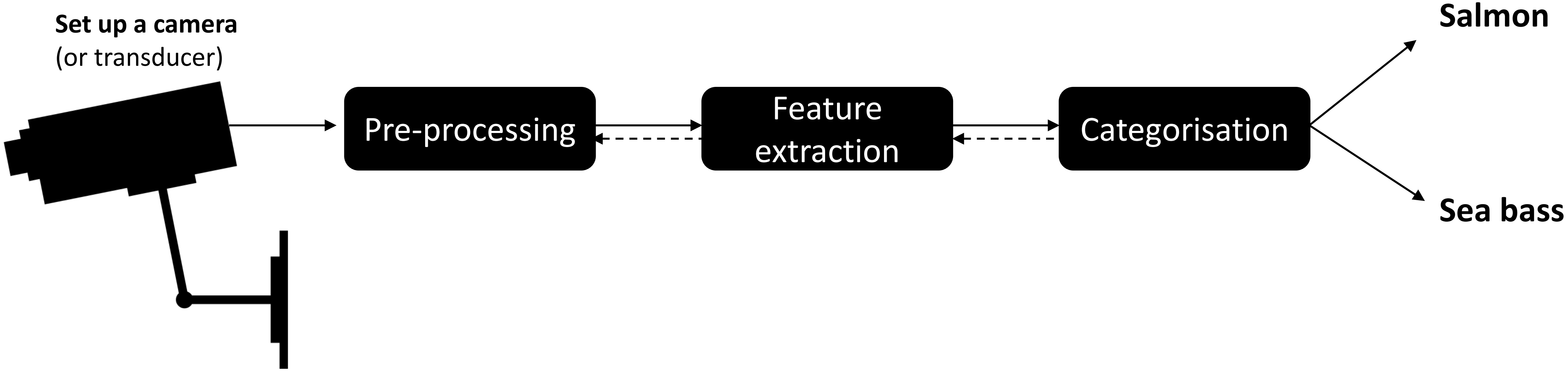
Pattern recognition systems

QUEEN'S UNIVERSITY BELFAST | SCHOOL OF ELECTRONICS, ELECTRICAL ENGNIEERING AND COMPUTER SCIENCE

Pattern recognition systems

Assume a system: measurement & observation

- A fish packing factory aims to automate the process of sorting incoming fish on a conveyor belt according to species.
- Pilot project: separating sea bass from salmon using optical sensing



Reference: Pattern Classification, R. O. Duda, P. E. Hart, D. G. Stork; Wiley

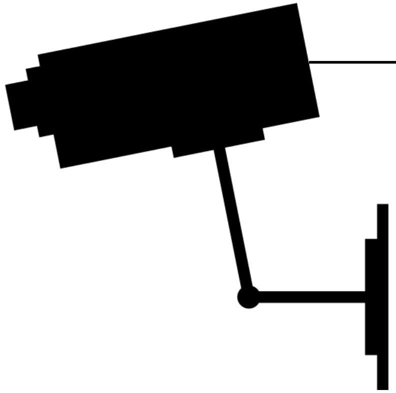
Pattern recognition systems

Assume a system: measurement & observation

- A fish packing factory aims to automate the process of sorting incoming fish on a conveyor belt according to species.
- Pilot project: separating sea bass from salmon using optical sensing

Object(s)

Set up a camera (or transducer)



Measurement

Pre-processing → Feature extraction → Categorisation → **Salmon** / **Sea bass**

Pre-processing: Filtering, removing noise and so on

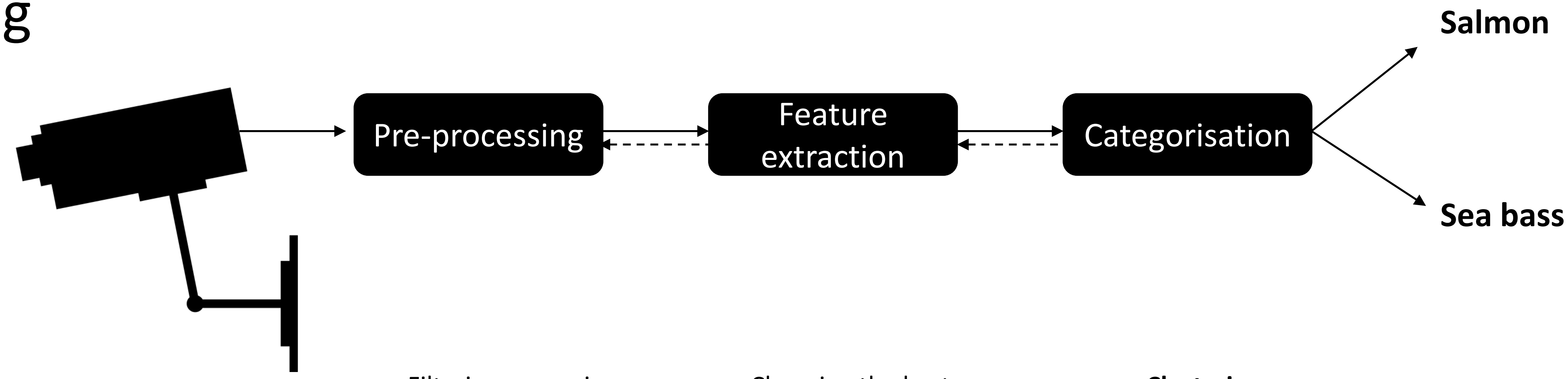
Feature extraction: Choosing the best features

Categorisation: **Clustering Classification**

Reference: Pattern Classification, R. O. Duda, P. E. Hart, D. G. Stork; Wiley

Pattern recognition systems

- Pilot project: separating sea bass from salmon using optical sensing



Object(s)

Measurement
1

Filtering, removing noise and so on
2

Choosing the best features
3

Clustering
Classification

Think and discuss about the three first stages of this pattern recognition system. What would you suggest for selecting features from an image?

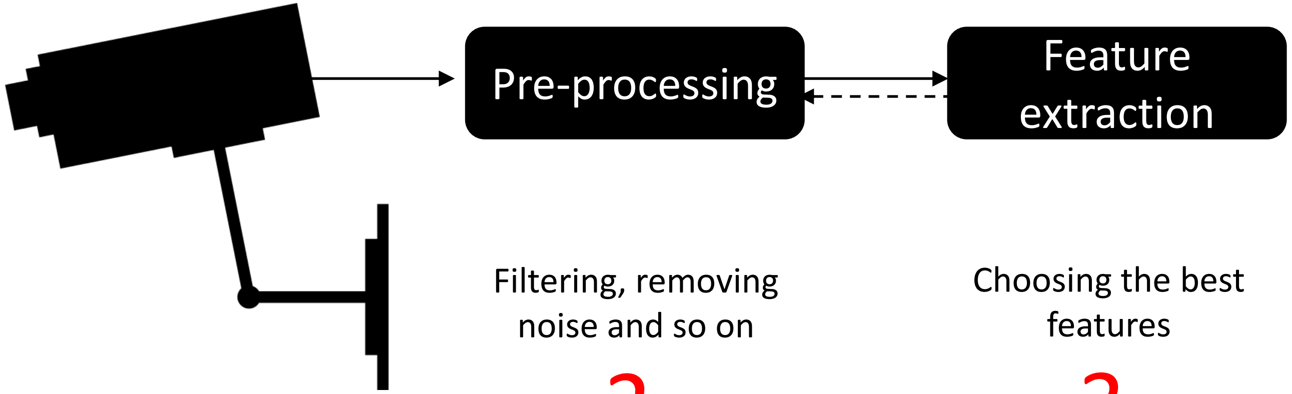
Reference: Pattern Classification, R. O. Duda, P. E. Hart, D. G. Stork; Wiley

QUEEN'S UNIVERSITY BELFAST

SCHOOL OF ELECTRONICS, ELECTRICAL ENGNIEERING AND COMPUTER SCIENCE

Pattern recognition systems

Object(s)



Measurement

1

Filtering, removing noise and so on

2

Choosing the best features

3

- One fish per image (using a *Segmentation* technique a single fish extracted)
- No colouring information
- In our measurement using the camera, we could get different parameters of an image object such as the size and the lightness
- We know that **there are two classes** (groups) for each observation/object (i.e., fish): salmon vs. sea bass

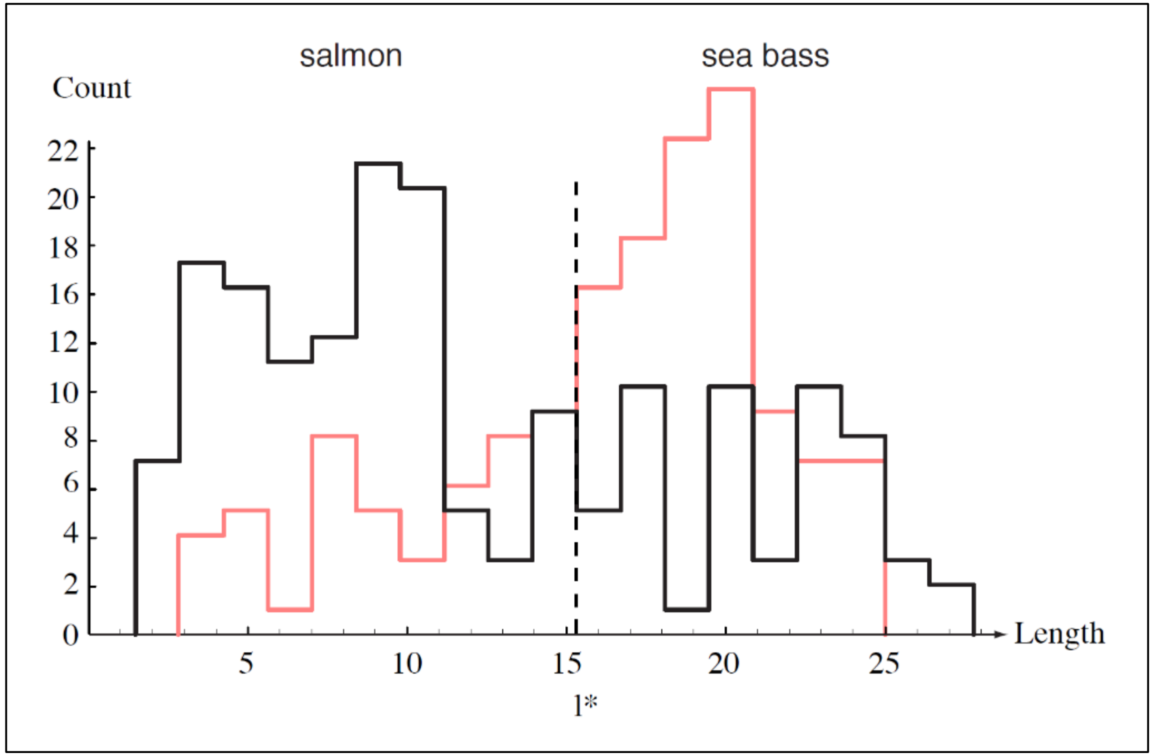
➡ **Supervised classification** or **classification**

Reference: Pattern Classification, R. O. Duda, P. E. Hart, D. G. Stork; Wiley

How to choose a feature?

No single threshold value of the ***length*** will serve to unambiguously discriminate between the two categories

There would be some errors if we use only ***length*** property as a feature



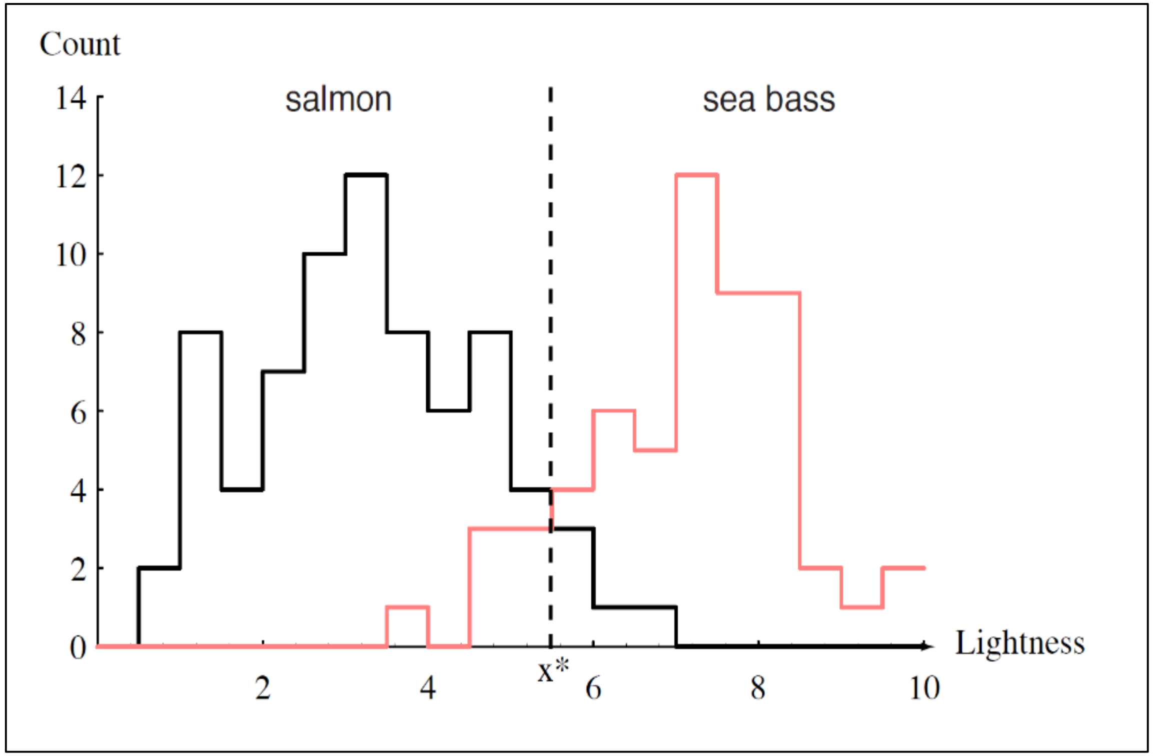
Histograms for the fish length for the two categories

Reference: Pattern Classification, R. O. Duda, P. E. Hart, D. G. Stork; Wiley

How to choose a feature?

No single threshold value of the ***lightness*** will serve to unambiguously discriminate between the two categories

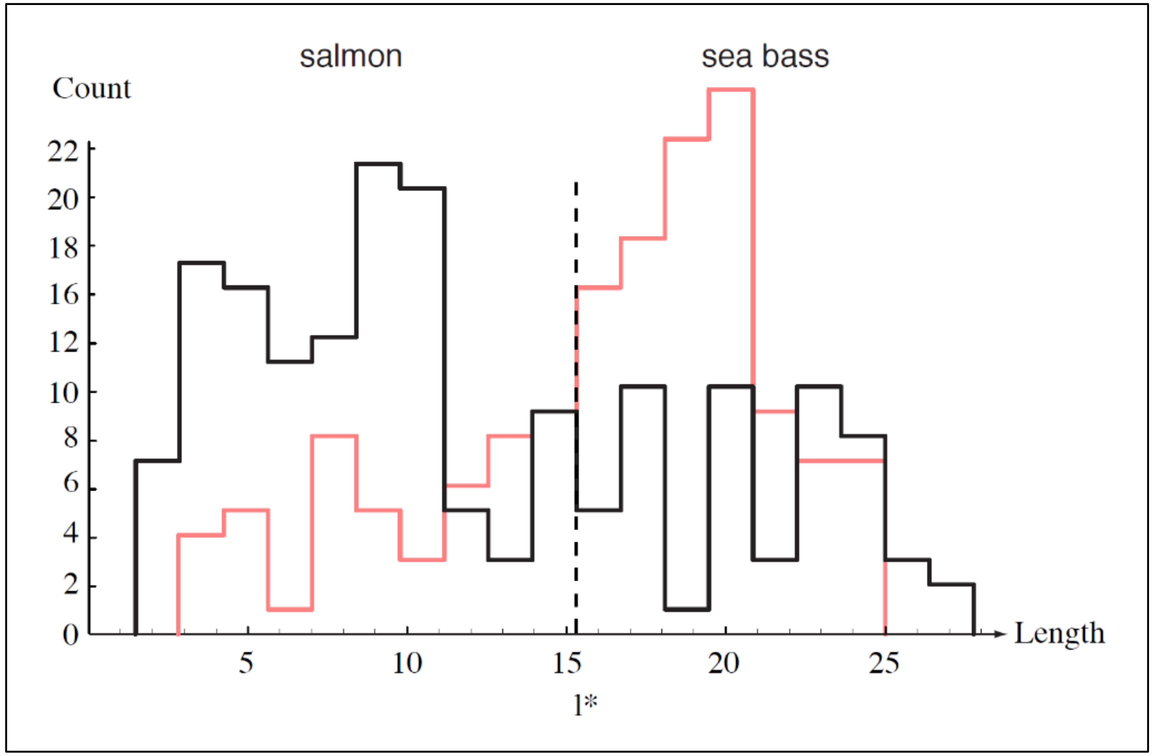
There would be some errors if we use only ***lightness*** property as a feature



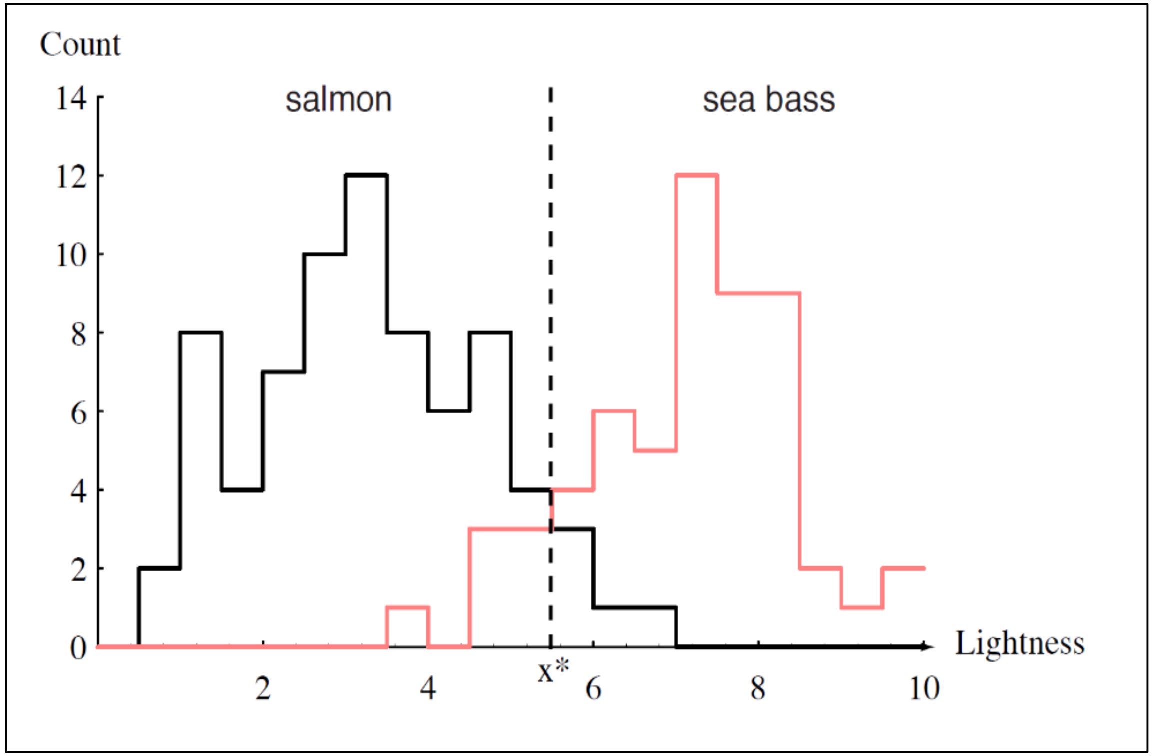
Histograms for the fish lightness for the two categories

Reference: Pattern Classification, R. O. Duda, P. E. Hart, D. G. Stork; Wiley

How to choose a feature?



Histograms for the fish length for the two categories



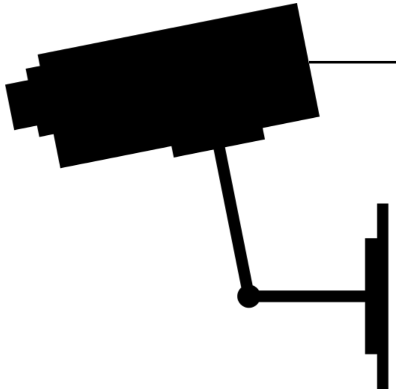
Histograms for the fish lightness for the two categories

Reference: Pattern Classification, R. O. Duda, P. E. Hart, D. G. Stork; Wiley

Pattern recognition systems

Object(s)
(samples)

Set up a camera
(or transducer)



Measurement

Pre-processing

Filtering, removing noise and so on

Feature extraction

Choosing the best features

x_1: **lightness**
x_2: **width**

Categorisation

Clustering
Classification

Salmon

Sea bass

The aim is to partition the feature space into two regions

Feature space: two dimensions $X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

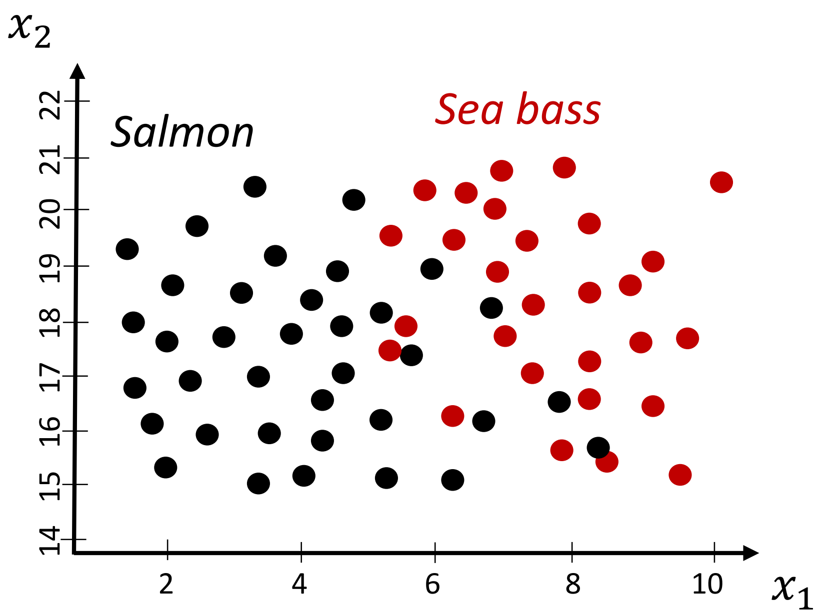
Suppose that we measure the feature vectors for our samples

Reference: Pattern Classification, R. O. Duda, P. E. Hart, D. G. Stork; Wiley

Feature space; lightness & width

x_1: lightness
x_2: width

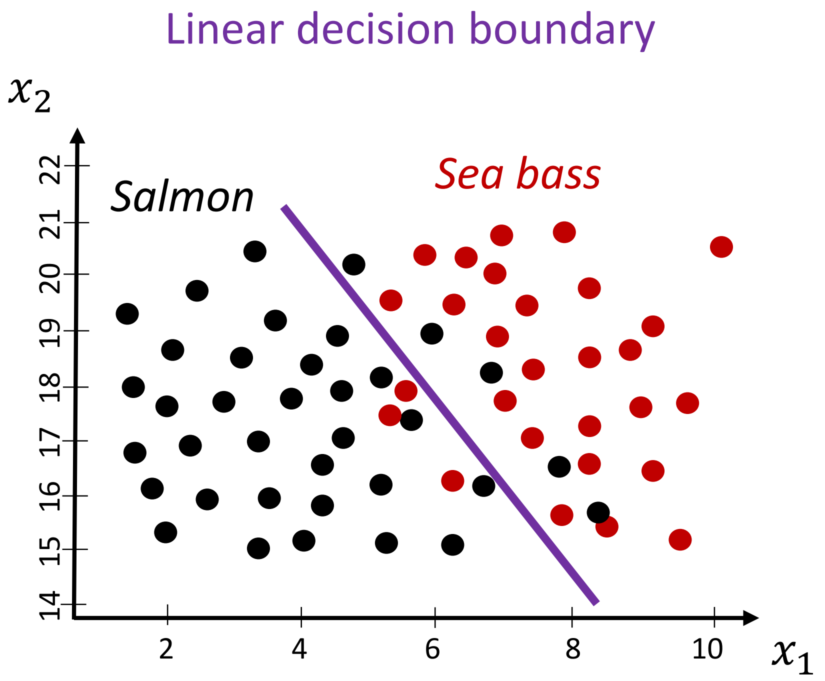
$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$



Decision boundary (line)

x_1: lightness
x_2: width

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

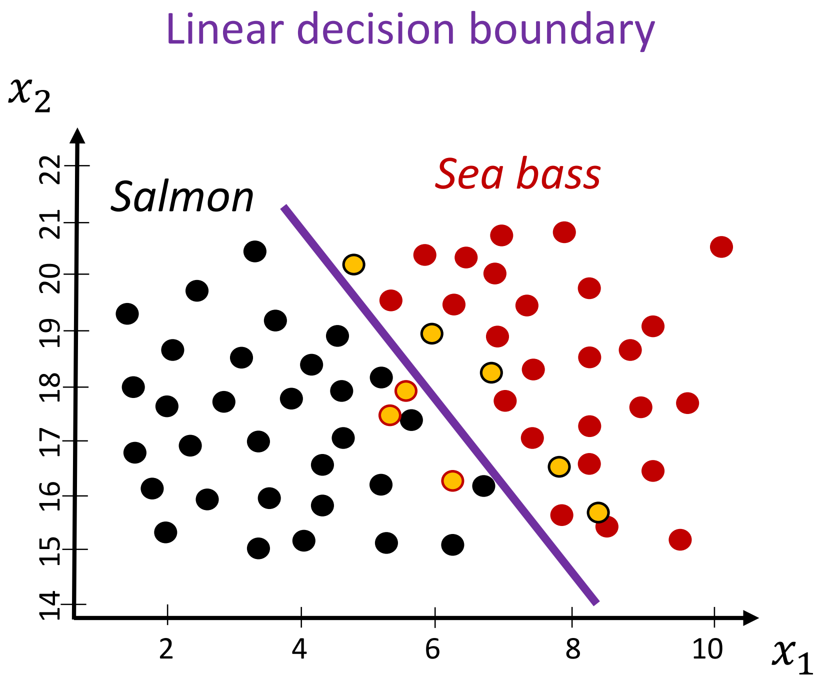


This plot suggests the following rule for categorising (separating) a fish:
Classify a fish as *salmon* if the feature vector of this fish **falls below** the line (this line is called decision boundary)

Classification error

x_1: lightness
x_2: width

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

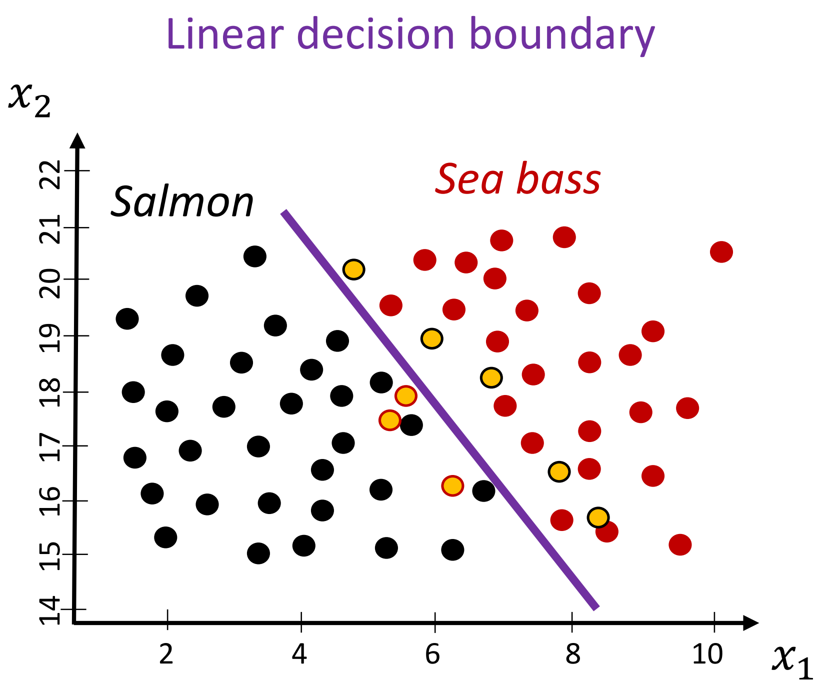


This plot suggests the following rule for categorising (separating) a fish:
Classify a fish as ***sea bass*** if the feature vector of this fish **falls above** the line (this line is called decision boundary)

How to reduce the classification error?

x_1: lightness
x_2: width

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

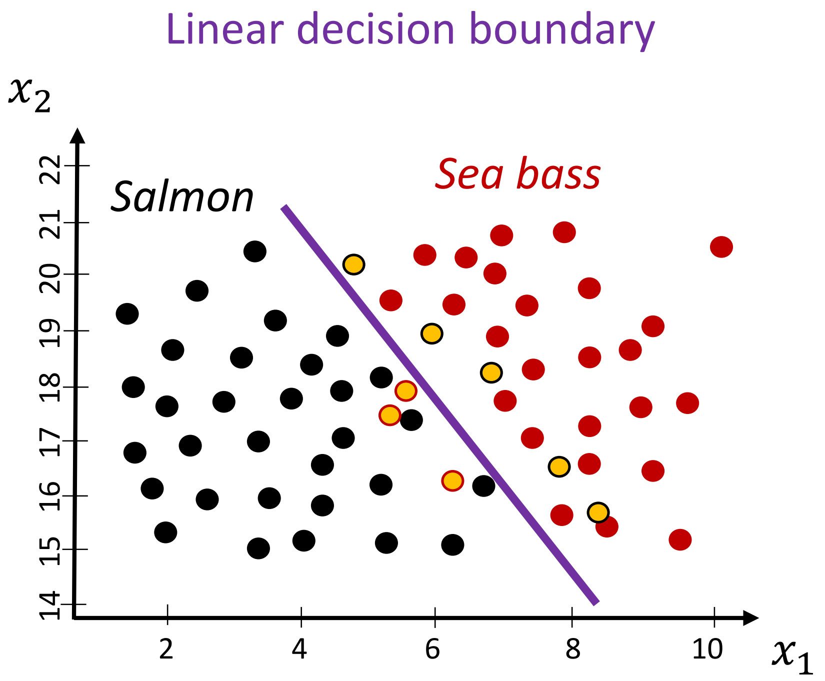


Any suggestions to reduce the classification error (i.e., to improve the accuracy of the classification)?

How to reduce the classification error?

x_1: lightness
x_2: width

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$



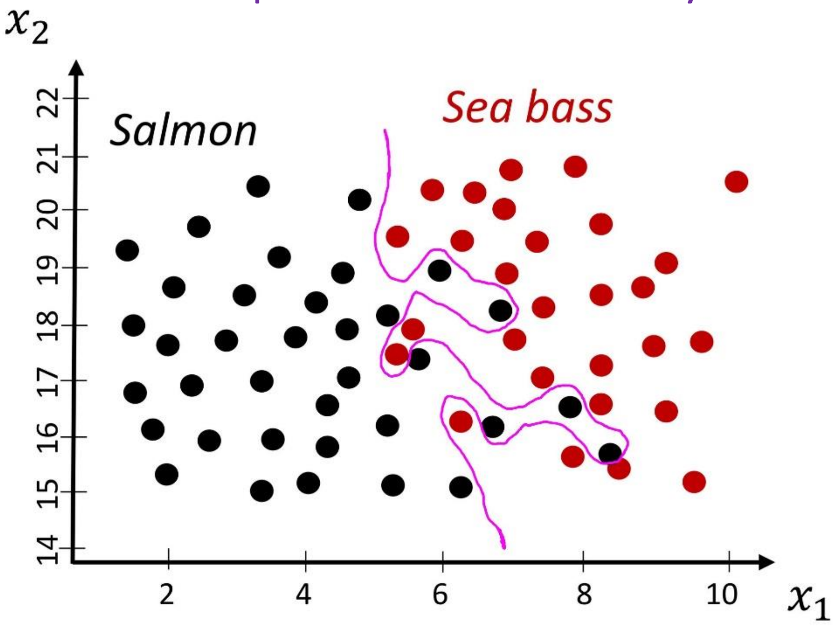
- **Include extra features such as the shape parameters of the fish**
  - E.g., the placement of the eyes (as expressed as a proportion of the mouth-to-tail distance)
  - Some features might be redundant
- **Choose a non-linear decision boundary instead of using a simple straight line!?**

How to reduce the classification error?

x_1: lightness
x_2: width

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

Complex decision boundary



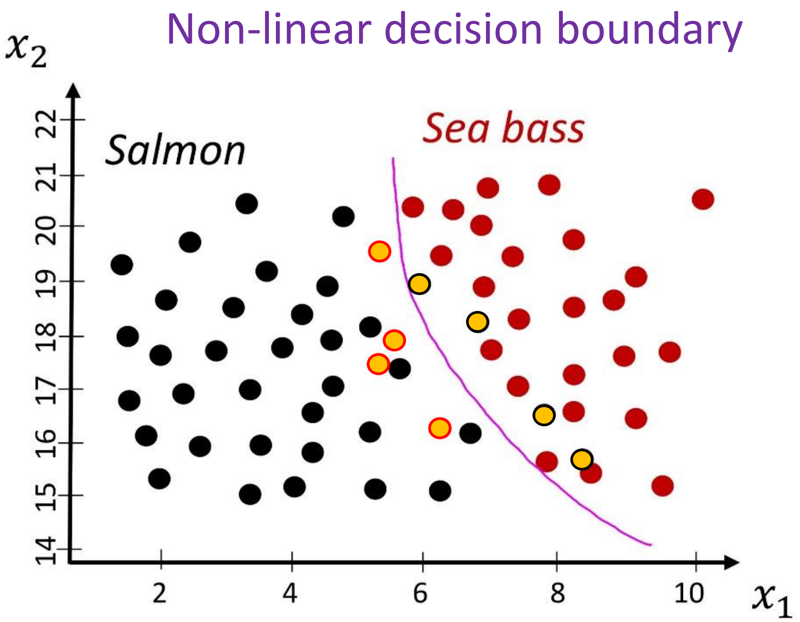
- **Include extra features such as the shape parameters of the fish**
  - E.g., the placement of the eyes (as expressed as a proportion of the mouth-to-tail distance)
  - Some features might be redundant
- **Choose a complex or non-linear decision boundary instead of using a simple straight line!?**

There is an issue of ***generalisation*** when we are using a complex decision boundary to perfectly separate the objects

How to reduce the classification error?

x_1: lightness
x_2: width

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$



- **Include extra features such as the shape parameters of the fish**
  - E.g., the placement of the eyes (as expressed as a proportion of the mouth-to-tail distance)
  - Some features might be redundant
- **Choose a complex or non-linear decision boundary instead of using a simple straight line!?**

There is an issue of ***generalisation*** when we are using a complex decision boundary to perfectly separate the objects

Classification vs. clustering

Classification

Objects/samples with classes → Pre-processing → Feature selection → Find a decision boundary → Final model

It's called model training or classifier training stage

Clustering

Objects/samples with no classes → Pre-processing → Feature selection → Feature space partitioning

Feature selection for high separability

QUEEN'S UNIVERSITY BELFAST

SCHOOL OF ELECTRONICS, ELECTRICAL ENGNIEERING AND COMPUTER SCIENCE

Classification – training vs. prediction

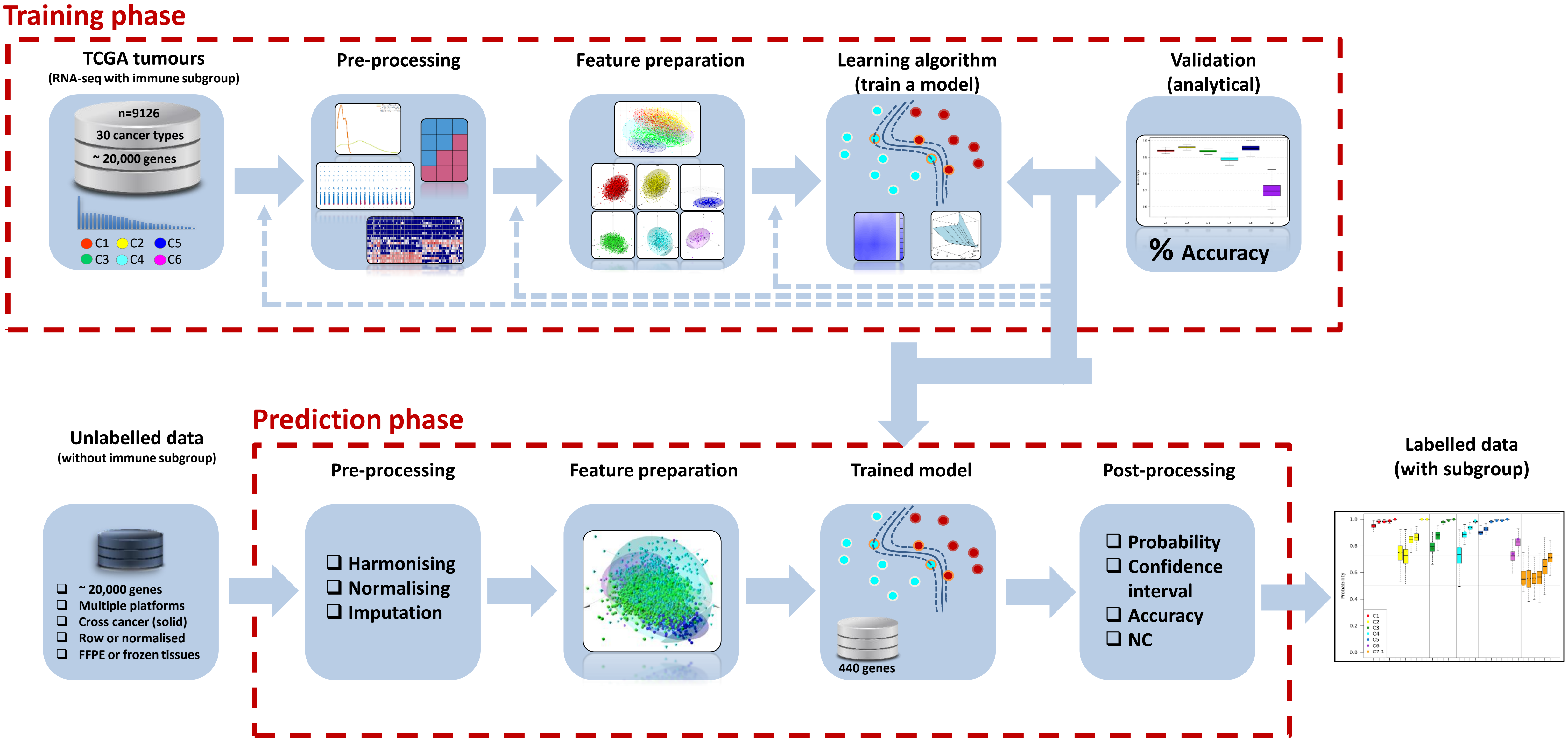


Figure 1.11 |Data pre-processing stage. a, Training phase. **b**, Prediction phase.

Unsupervised learning

Unsupervised clustering

- What is clustering?
- Why would we want to cluster?
- How would you determine clusters?
- How can you do this efficiently?

Clustering - concept

Basic idea: group together similar objects/samples/data

Organising unlabelled data into similar groups called clusters

Clustering or grouping

Cluster analysis or clustering is the task of grouping/partitioning a set of instances/objects/data points in such a way that data points in the same group are more similar to each other than to those in other groups

Clustering - concept

What could "similar" mean?

Clustering - concept

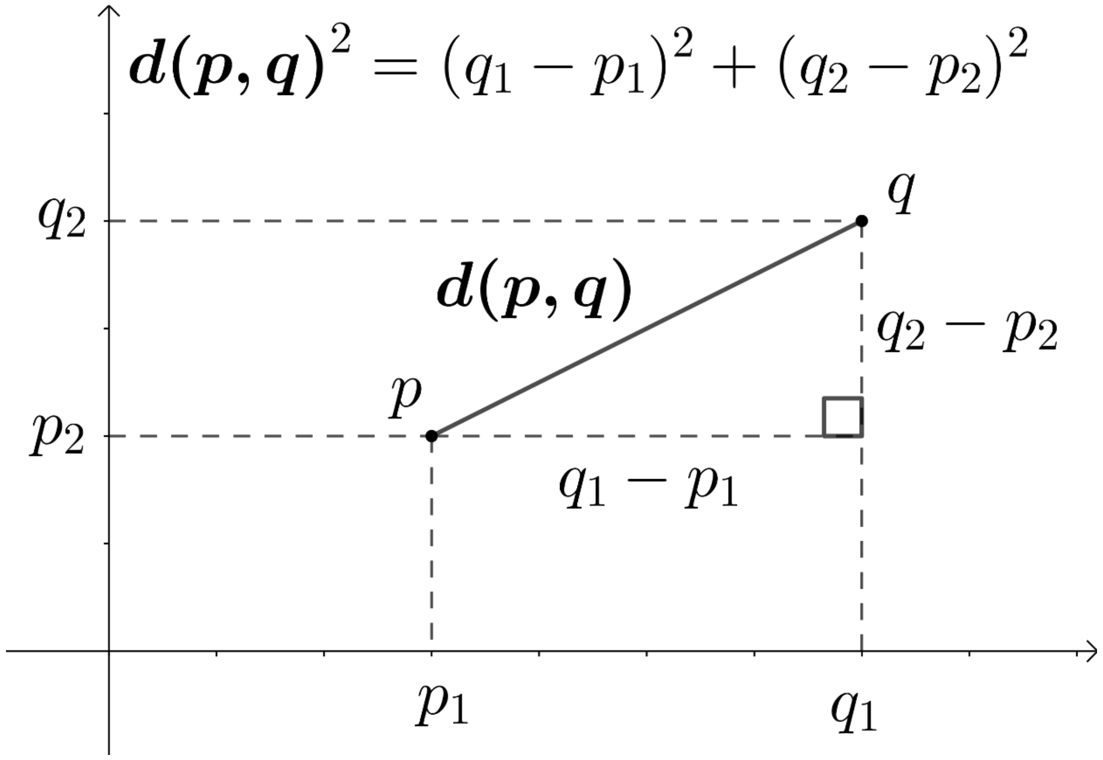
What could "similar" mean?

- One option: Euclidean distance (squared)

Clustering - similarity

What could "similar" mean?

- One option: Euclidean distance (squared)



Euclidean distance in $\mathcal{R}^2$

Two dimensions

if **p** = (p_1, p_2) and **q** = (q_1, q_2) then the distance is given by

$$d(p,q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}$$

Clustering - similarity

What could "similar" mean?

- One option: Euclidean distance (squared)

Euclidean distance in $\mathcal{R}^n$

n dimensions

$$\boldsymbol{X} = (x_1, x_2, \dots, x_n) \quad \boldsymbol{Y} = (y_1, y_2, \dots, y_n)$$

Then the Euclidean distance is given by

$$d(p,q) = \sqrt{(y_1 - x_1)^2 + (y_2 - x_2)^2 + \cdots + (y_n - x_n)^2}$$

$$d(\boldsymbol{X},\boldsymbol{Y}) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

Clustering - similarity

What could "similar" mean?

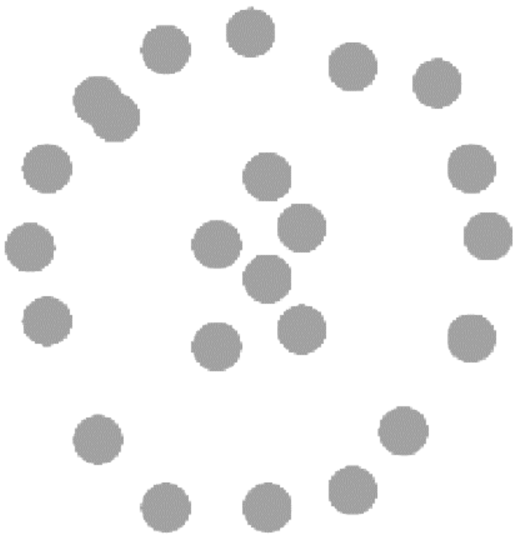
- One option: Euclidean distance (squared)
- Clustering results are remarkably dependent on the measure of similarity (or distance) between data points to be clustered

Chebyshev distance measures distance assuming only the most significant dimension is relevant.
Manhattan distance measures distance following only axis-aligned directions.
Minkowski distance is a generalization that unifies Euclidean distance, Manhattan distance, and Chebyshev distance

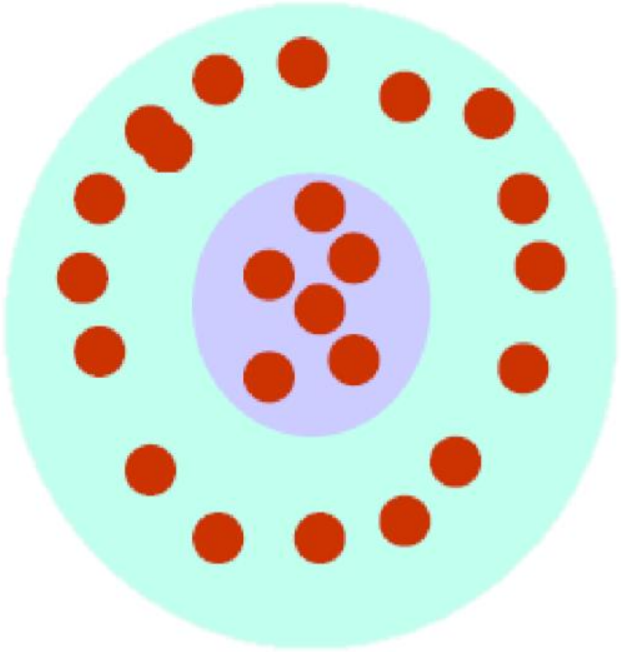
A cluster is a collection of data points which are "similar" between them, and "dissimilar" to data points in other clusters.

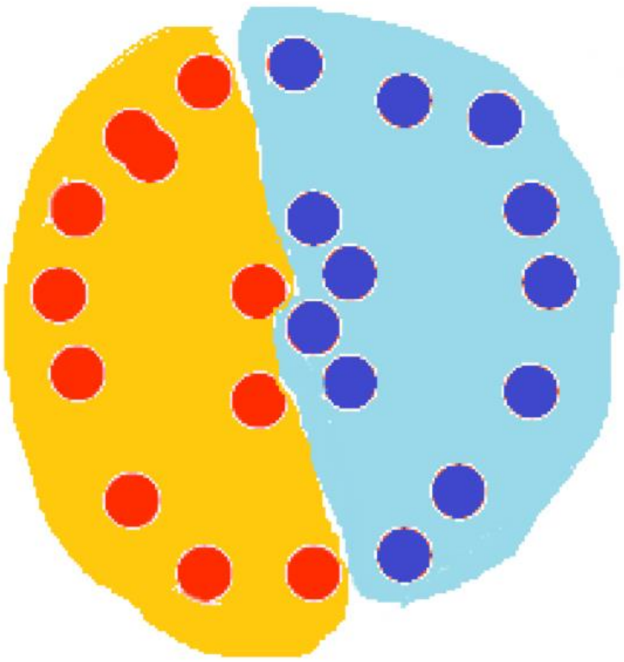
Clustering – cluster/group

Two different clustering results (i.e., clusters)



Original data points





Clustering - some applications

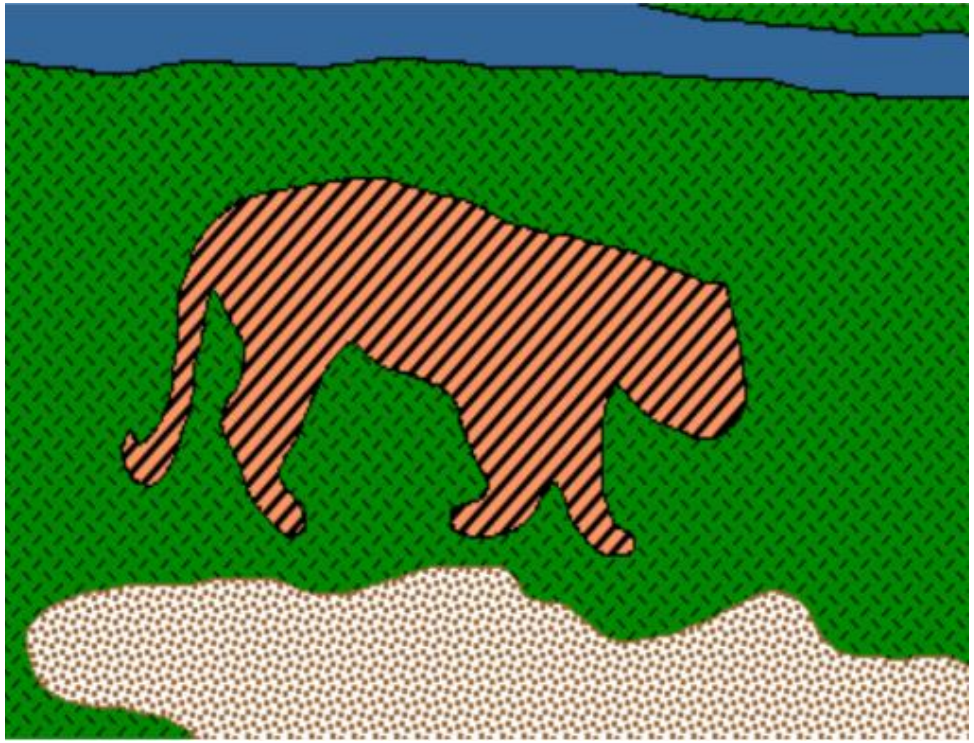
- Social network analysis
  - the discovery of clusters or communities, target marketing schemes, etc.
- Market segmentation
- Search result grouping
- Medical imaging
- Image segmentation and image concept extraction
- Anomaly detection
- …

Clustering image pixels

Image segmentation

Goal: identify groups of pixels that are similar and meaningfully connected

?



Discuss about data points, feature types for this clustering example

Clustering image pixels

Aim: detecting and extracting <u>interest regions</u> from an image

Identify groups of pixels that are similar and meaningfully connected

a

How?

b

Data (only colour)

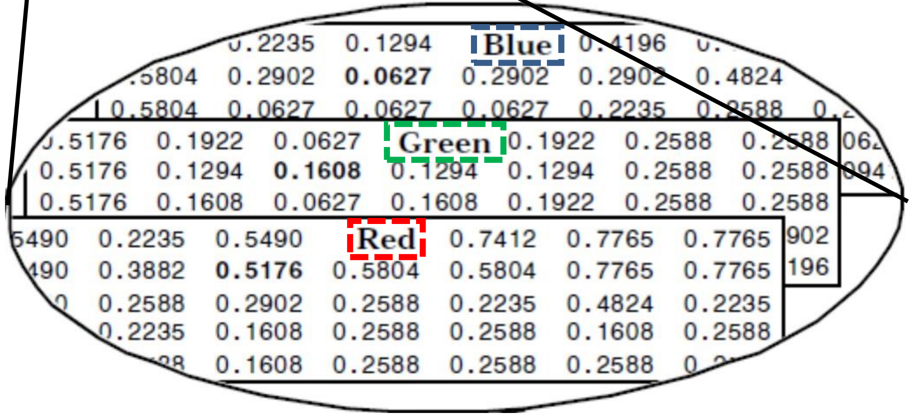


Figure 1.3 | **a**, A colour image including an interest region (i.e., butterfly). **b**, Interest region extracted by a computer program

https://www.sciencedirect.com/science/article/pii/S0031320313001179

QUEEN'S UNIVERSITY BELFAST

SCHOOL OF ELECTRONICS, ELECTRICAL ENGNIEERING AND COMPUTER SCIENCE

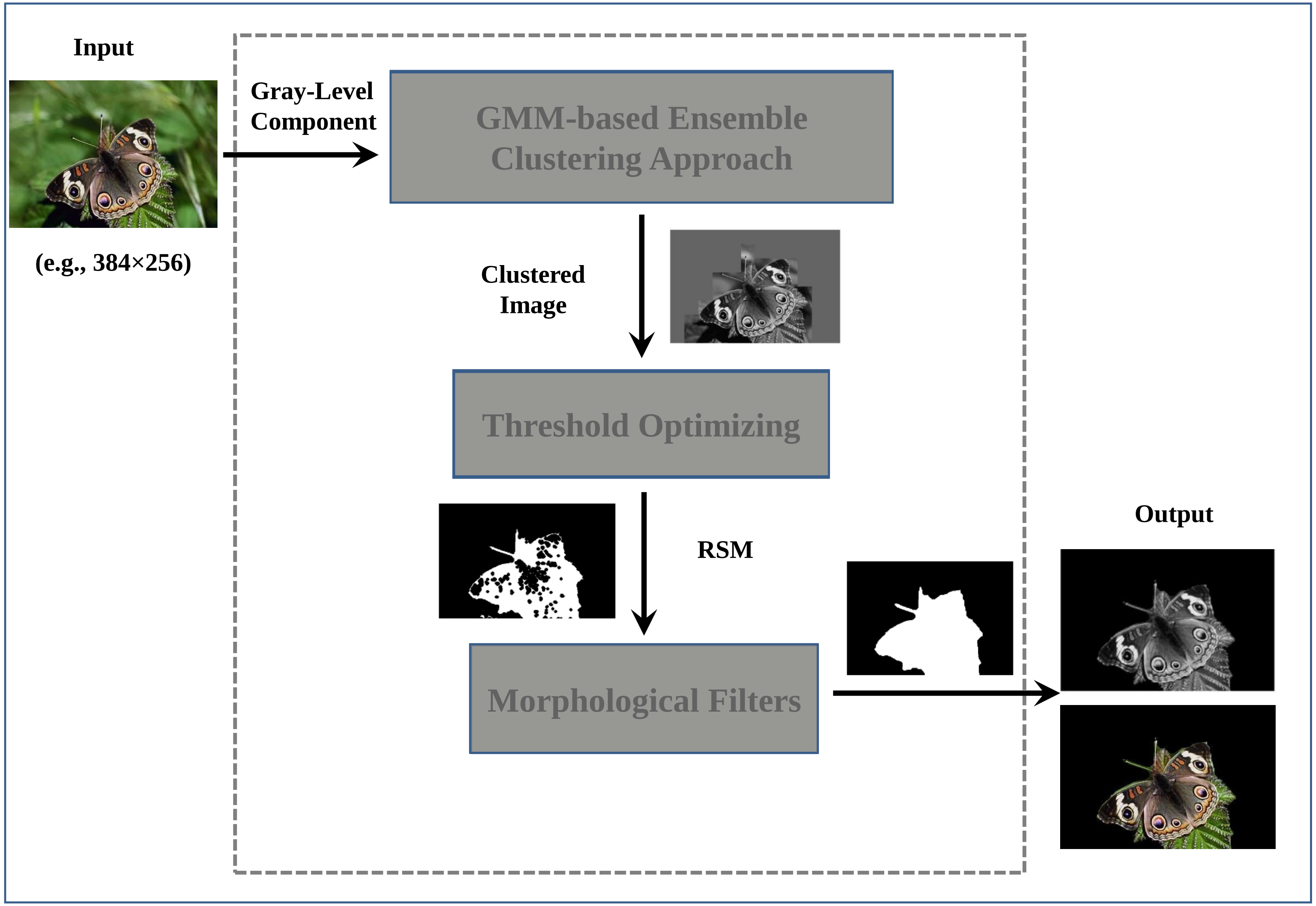
Clustering image pixels

Region-of-interest extraction from images

G. Rafiee, S.S. Dlay, W.L. Woo
Region-of-interest extraction in low depth of field images using ensemble clustering and difference of Gaussian approaches
Pattern Recognition., 46 (10) (2013), pp. 2685-2699

Clustering image pixels

Region-of-interest extraction from images



G. Rafiee, S.S. Dlay, W.L. Woo
Region-of-interest extraction in low depth of field images using ensemble clustering and difference of Gaussian approaches
Pattern Recognition., 46 (10) (2013), pp. 2685-2699

Clustering image pixels

Region-of-interest extraction from images

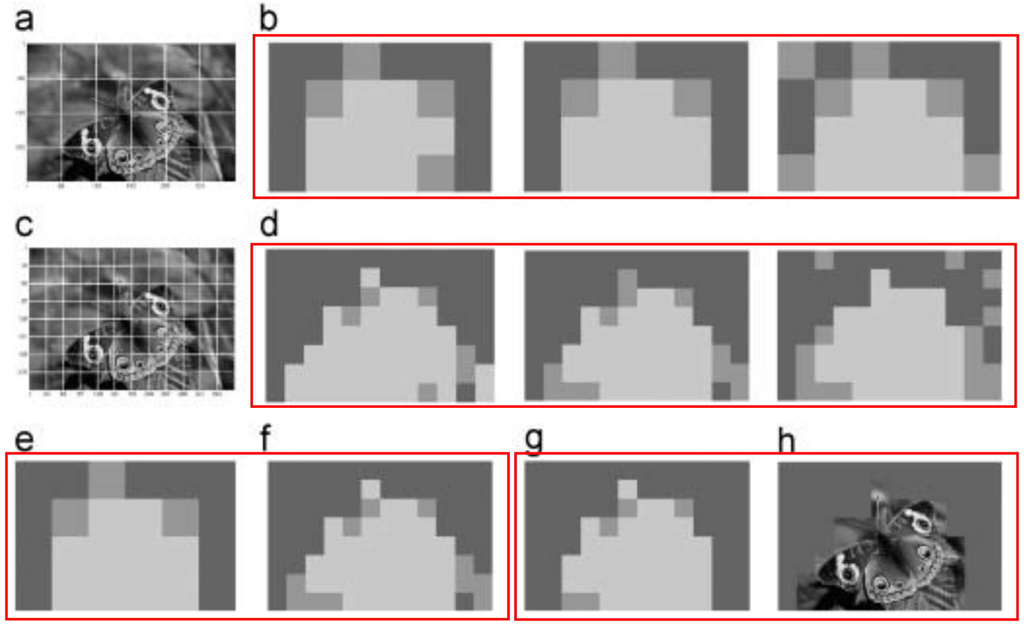


Illustration of different partitions and the fusion decision process. (a) and (c) Grayscale images with uniform partitioning at two consecutive levels, i.e., 64*64 and32*32. (b) and (d) Different partitions corresponding to different local optima at the first and second level, respectively. (e) and (f) Partitions after aggregating process in each level. (g) Final partition after combining (e) and (f). (h) Clustered image.

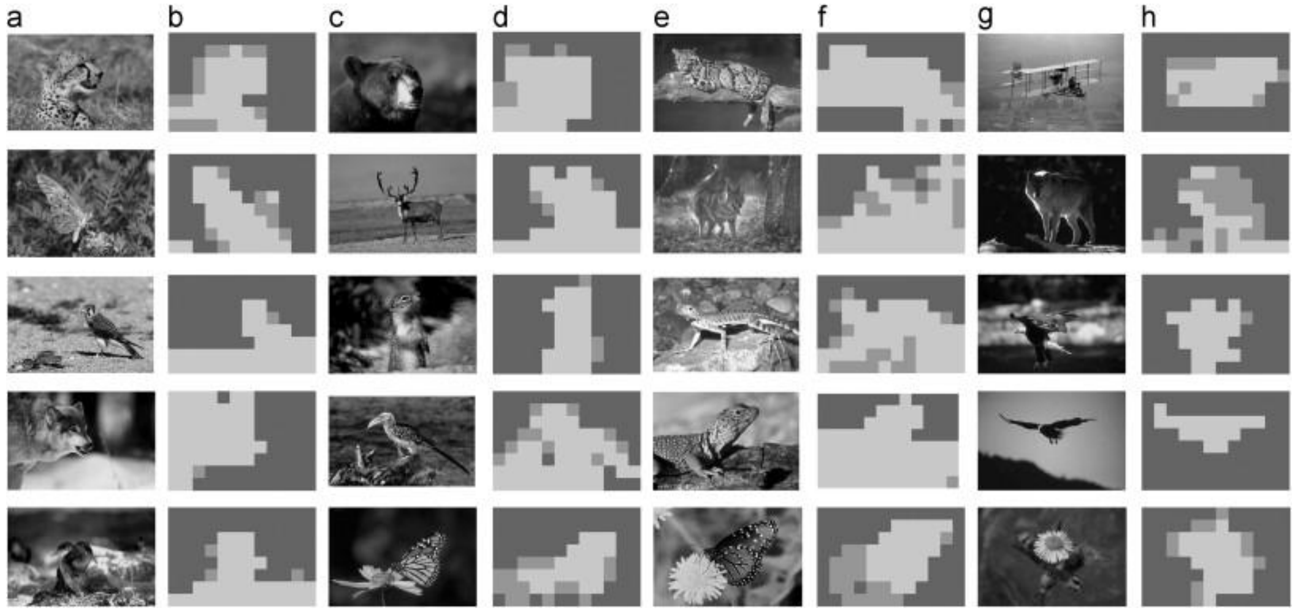
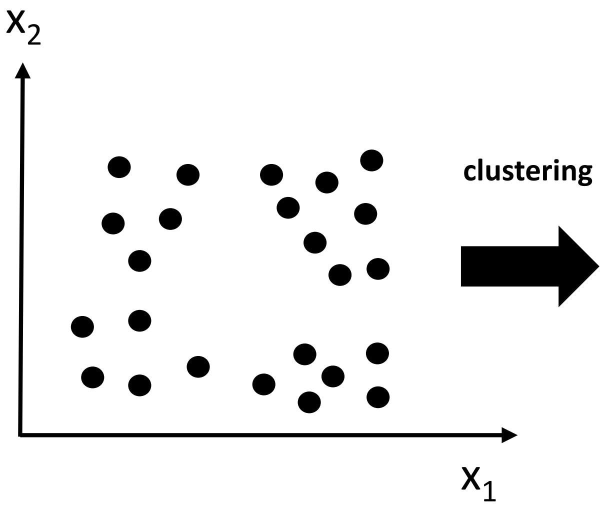


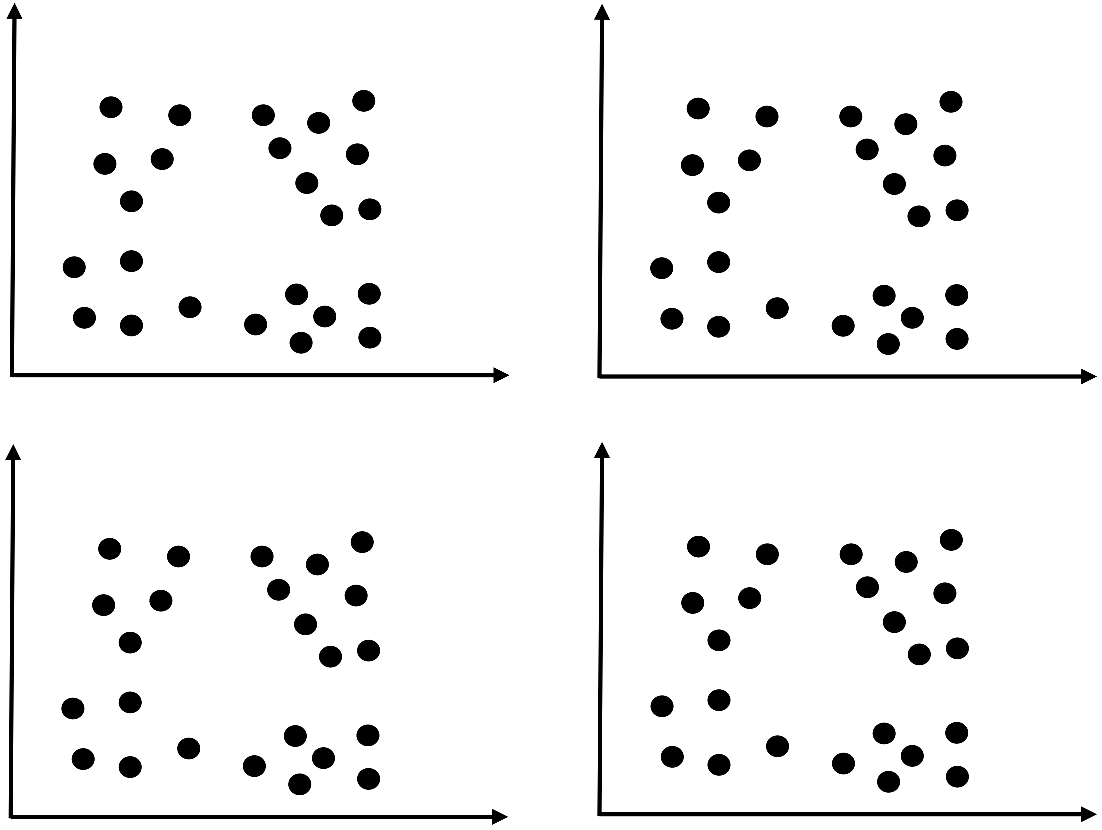
Illustration of final partitions for a number of images obtained from the algorithm withT_1=10. (a), (c), (e), and (g) Grayscale test images. (b), (d), (f), and (h) Final partitions after employing the combining process.

https://www.sciencedirect.com/science/article/pii/S0031320313001179#f0040

Number of clusters

Clustering concept





And more …

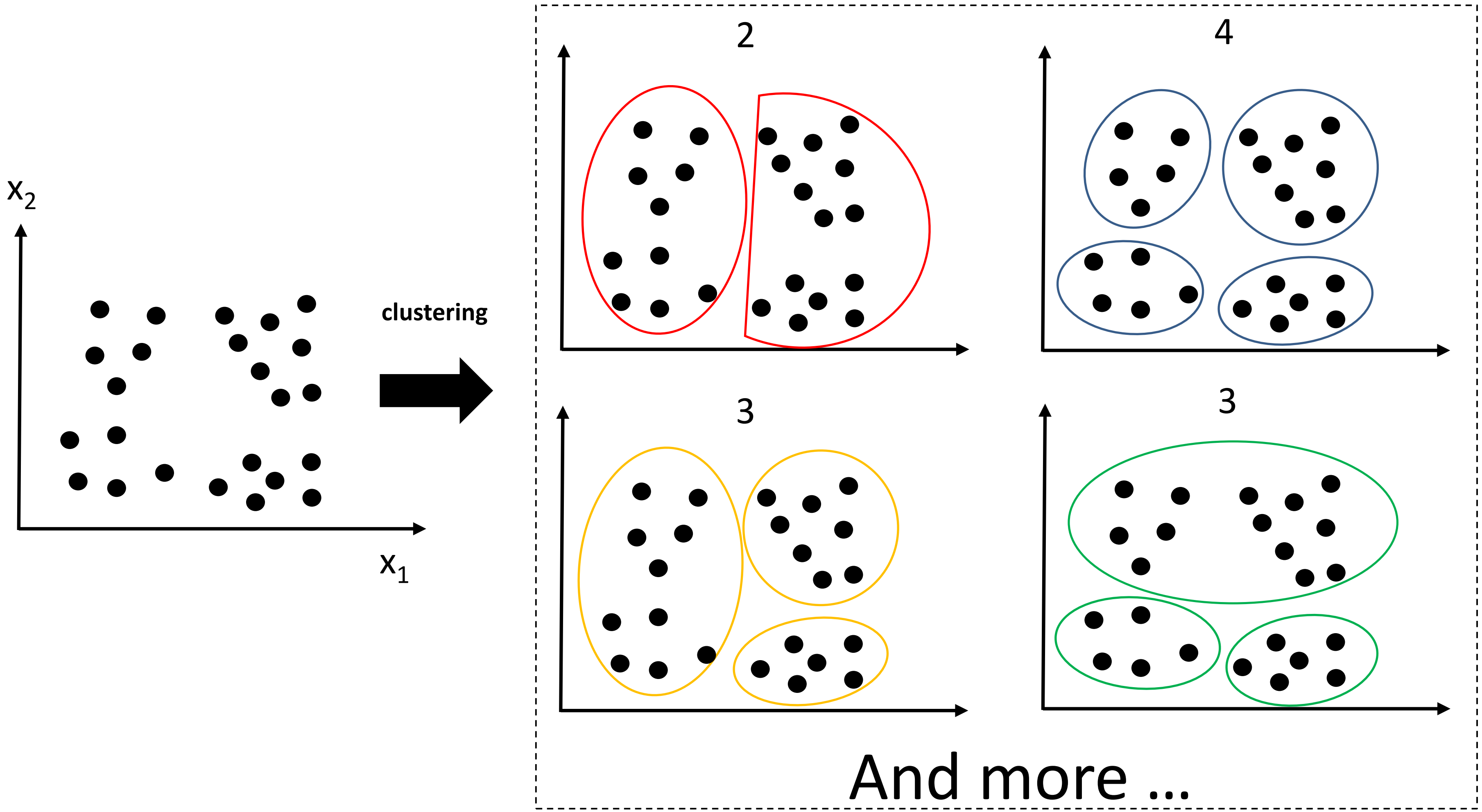
In case of applying an appropriate clustering method (with well-adjusted parameters/initialisation, bootstrapping and cross-validation techniques), we could have distinct groups (with possibly different number of clusters) but they might be meaningless!

QUEEN'S UNIVERSITY BELFAST

SCHOOL OF ELECTRONICS, ELECTRICAL ENGNIEERING AND COMPUTER SCIENCE

Number of clusters

Identifying the number of clusters is a very challenging task

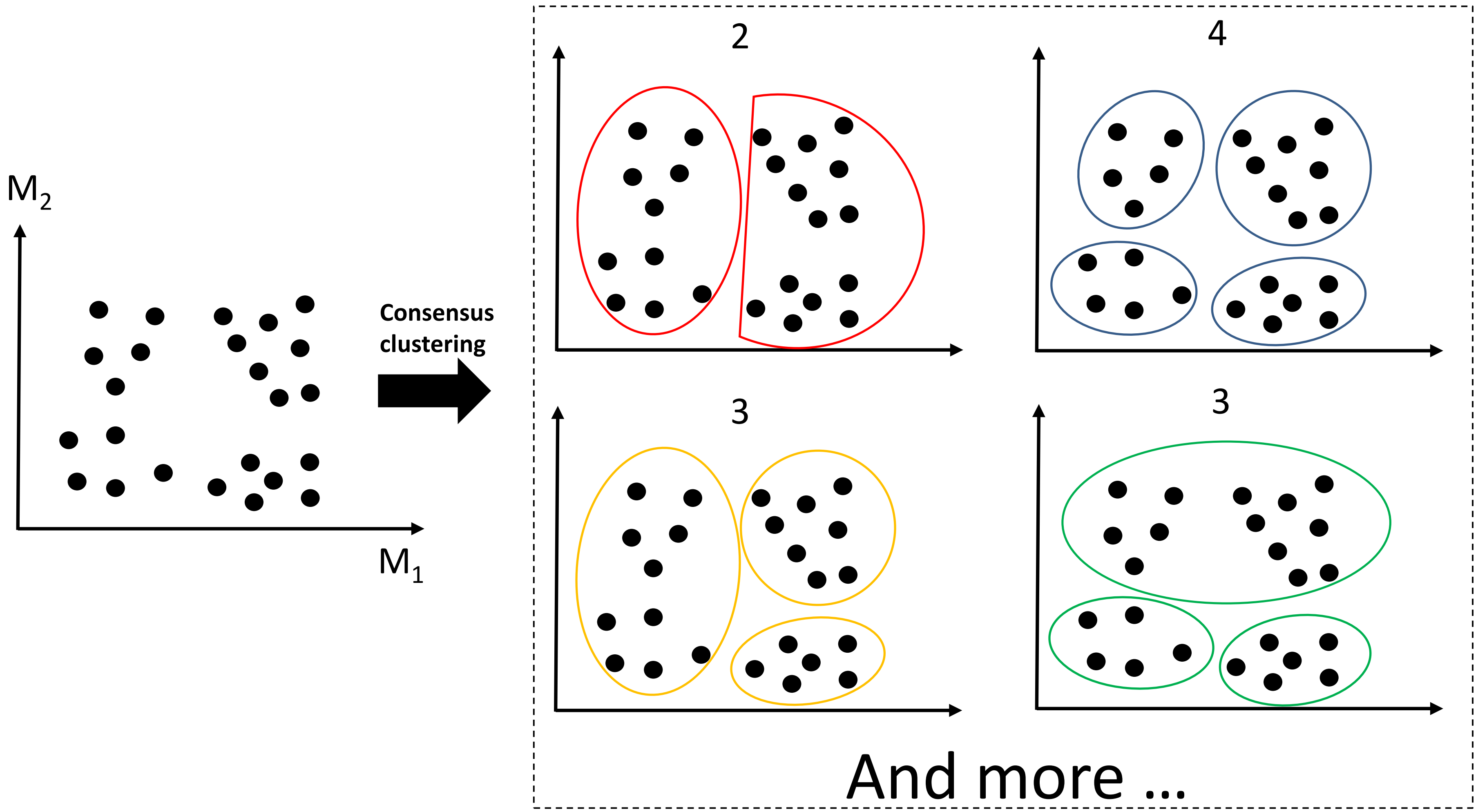


In case of applying an appropriate clustering method (with well-adjusted parameters/initialisation, bootstrapping and cross-validation techniques), we could have distinct groups (with possibly different number of clusters) but they might be meaningless!

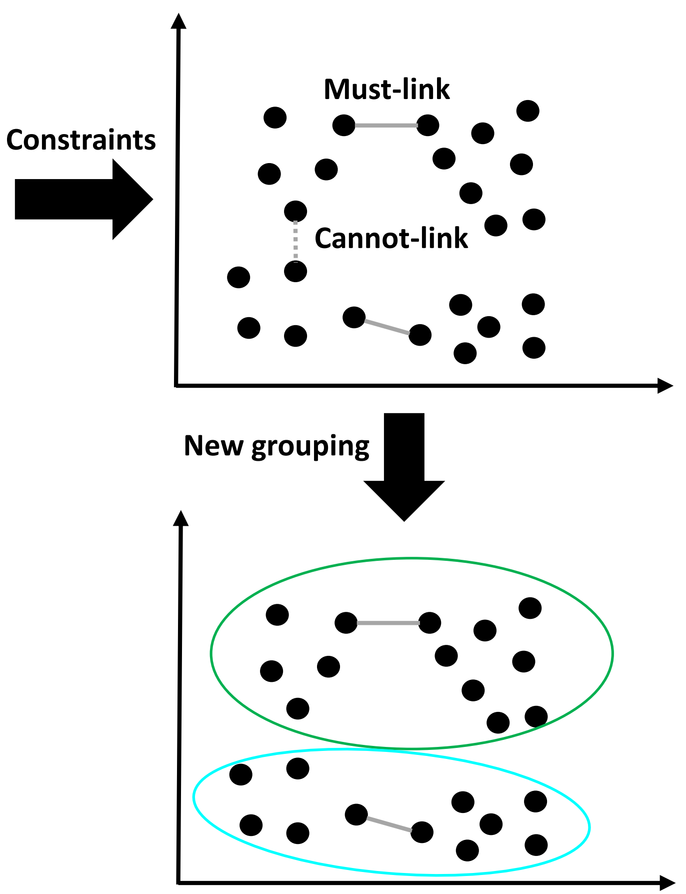
Prior knowledge – constrained clustering

QUEEN'S UNIVERSITY BELFAST
SCHOOL OF ELECTRONICS, ELECTRICAL ENGNIEERING AND COMPUTER SCIENCE

Sometimes prior knowledge could help in finding the correct number of clusters



A must-link constraints is the knowledge/information which we are aware of some samples' connection



In case of applying an appropriate consensus clustering method (with well-adjusted parameters/initialisation, bootstrapping and cross-validation techniques), we could have distinct groups (with possibly different number of clusters) but they might be biologically meaningless!

Must-link and cannot-link constraints are indicated by solid line and dashed line, respectively (e.g., 3 constraints).

K-means clustering

- k-means is one of the simplest unsupervised learning algorithms
- It classifies a given data set through a certain number of clusters (let's say k clusters)

K-means clustering

Basic Algorithm:

- Step 1: select k (number of clusters)
- Step 2: randomly select k initial cluster centers
- Step 3: calculate distance from each data point to each cluster center
  - What type of distance should we use? E.g., Euclidean distance
- Step 4: Assign each data point to the closest cluster
- Step 5: Recalculate the new cluster centre
- Repeat Step 3-5 until a final stop condition

K-means clustering

- Strengths
  - Simple and fast
  - Finds cluster centres that minimize conditional variance (good representation of data)
  - Easy to implement
- Weaknesses
  - Need to choose k
  - Sensitive to outliers
  - Prone to local minima and no guarantee of optimal solution (local optima)
    - Repeat with different starting values
  - Difficult to guess the correct “k”

Any Questions?