Lecture 6: Clustering

Xenia Miscouridou

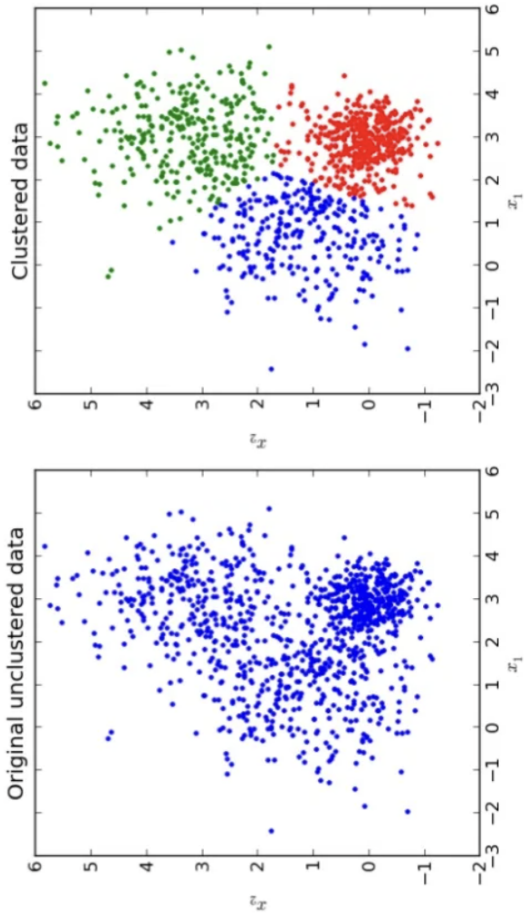
Imperial College London

Machine Learning Summer School
July 2022

Outline

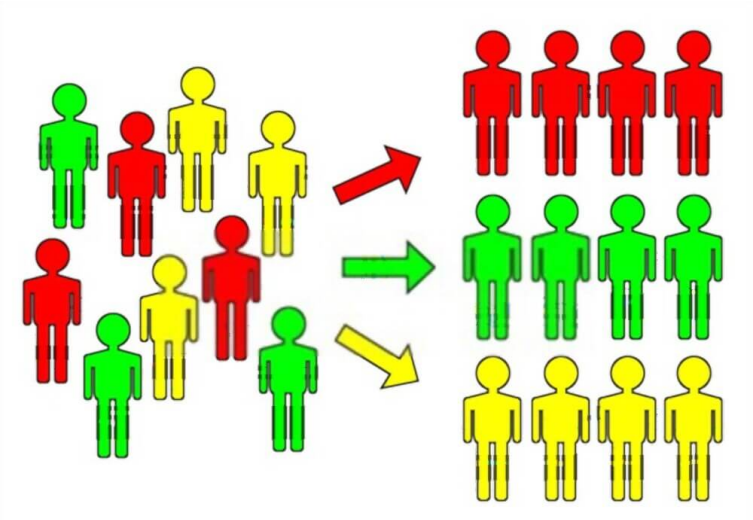
What is clustering?

- The organisation of unlabeled data into similarity groups called clusters
- A cluster is a collection of data items which are similar between them, and dissimilar to data items in other clusters
- Organising data into clusters such that there is
  - high intra-cluster similarity
  - low inter-cluster similarity
- Why do we want to do that?
- How do we do that?



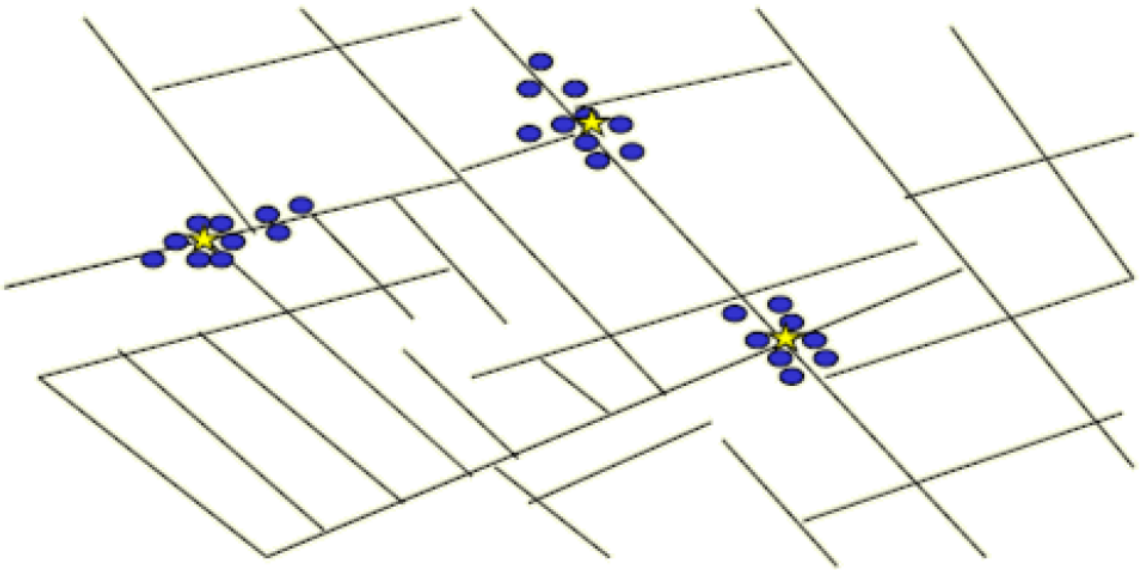
Customer segmentation

- One of the most important marketing tools, because it can help a business better understand its target audience
- This is because it groups customers based on common characteristics
- These groups can be used to build an overview of customers



Historical Example on Cholera in London

- John Snow, a London physician plotted cholera deaths on a map during an outbreak in 1850s
- Locations indicated that cases were clustered around certain intersections where there were polluted walls exposing both the problem and the solution



From: Nina Mishra HP Labs

Outline

Why do we care?

These clusters

- can provide clear interpretable meaning in many applications,
- can be used to label the data and train classifiers,
- can be used to detect departures from some expected characteristics of the data

This is in contrast to dimensionality reduction, the other form of unsupervised learning that we considered previously, which allows us to discover low-dimensional projections that

- can be used to visualise the data,
- represent features or input vectors that replace the original variables and can be used to more efficiently train a classifier on labelled data

Why do we care?

These clusters

- can provide clear interpretable meaning in many applications,
- can be used to label the data and train classifiers,
- can be used to detect departures from some expected characteristics of the data

This is in contrast to dimensionality reduction, the other form of unsupervised learning that we considered previously, which allows us to discover low-dimensional projections that

- can be used to visualise the data,
- represent features or input vectors that replace the original variables and can be used to more efficiently train a classifier on labelled data

What do we need in order to design a clustering method?

1. Similarity measure:

- Define a measure to help us find a partition such that data in the same group are more similar compared to data points in other groups
- For two data points x_1 and x_2 of dim D denote the similarity measure by $s(x_1, x_2)$ and a distance measure $d(x_1, x_2)$

- A common distance measure is the Euclidean distance

$$d(x_1, x_2) = \sqrt{\sum_{k=1}^{D} \left(x_1^{(k)} - x_2^{(k)}\right)^2}$$

What do we need in order to design a clustering method?

1. Similarity measure:

- Define a measure to help us find a partition such that data in the same group are more similar compared to data points in other groups
- For two data points x_1 and x_2 of dim D denote the similarity measure by $s(x_1, x_2)$ and a distance measure $d(x_1, x_2)$

- A common distance measure is the Euclidean distance

$$d(x_1, x_2) = \sqrt{\sum_{k=1}^{D} \left(x_1^{(k)} - x_2^{(k)}\right)^2}$$

What do we need in order to design a clustering method?

1. Similarity measure:

- Define a measure to help us find a partition such that data in the same group are more similar compared to data points in other groups
- For two data points x_1 and x_2 of dim D denote the similarity measure by $s(x_1, x_2)$ and a distance measure $d(x_1, x_2)$

- A common distance measure is the Euclidean distance

$$d(x_1, x_2) = \sqrt{\sum_{k=1}^{D} \left(x_1^{(k)} - x_2^{(k)} \right)^2}$$

What do we need in order to design a clustering method?

2. Criterion function to evaluate the clustering

- Intra-cluster cohesion (compactness)
- Inter-cluster separation (isolation)

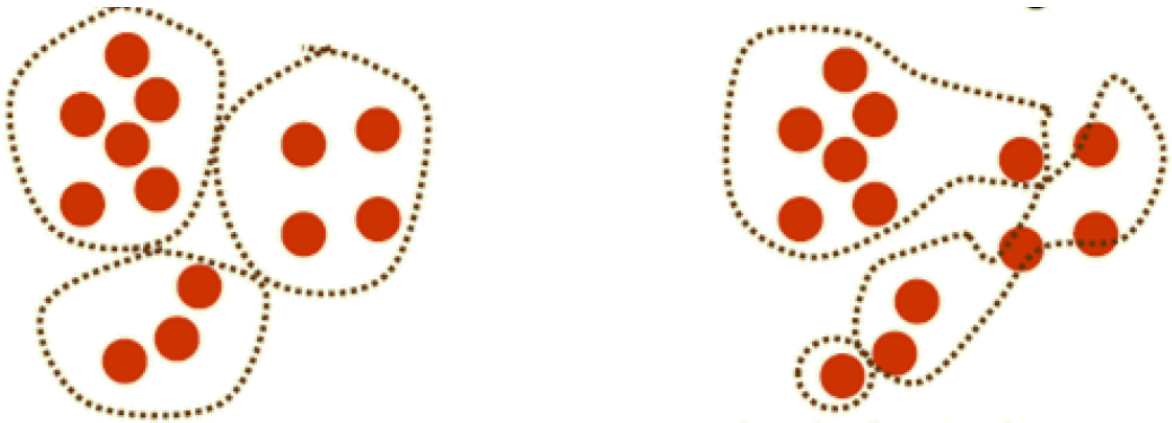


Figure: Good vs Bad clustering

3. An algorithm to compute the clustering that is scalable, interpretable and does not rely on domain knowledge

What do we need in order to design a clustering method?

2. Criterion function to evaluate the clustering

- Intra-cluster cohesion (compactness)
- Inter-cluster separation (isolation)

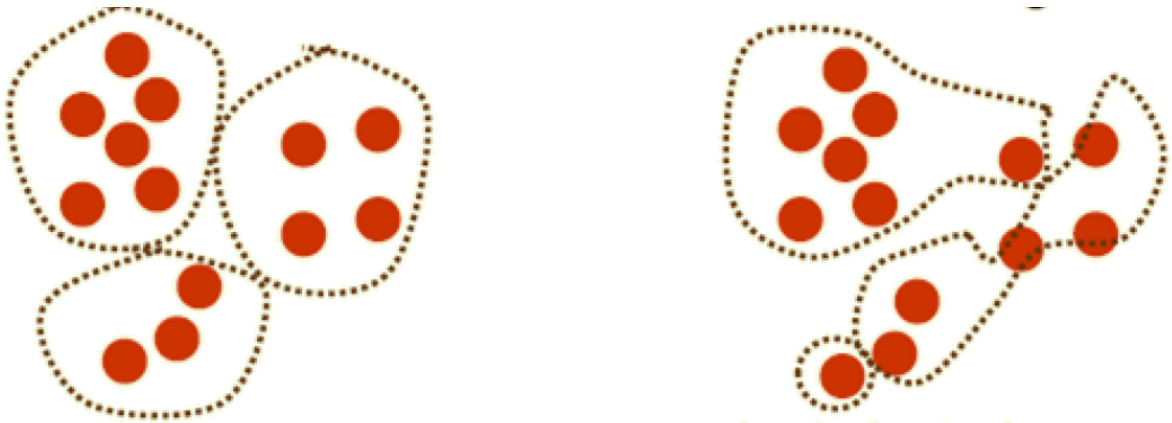
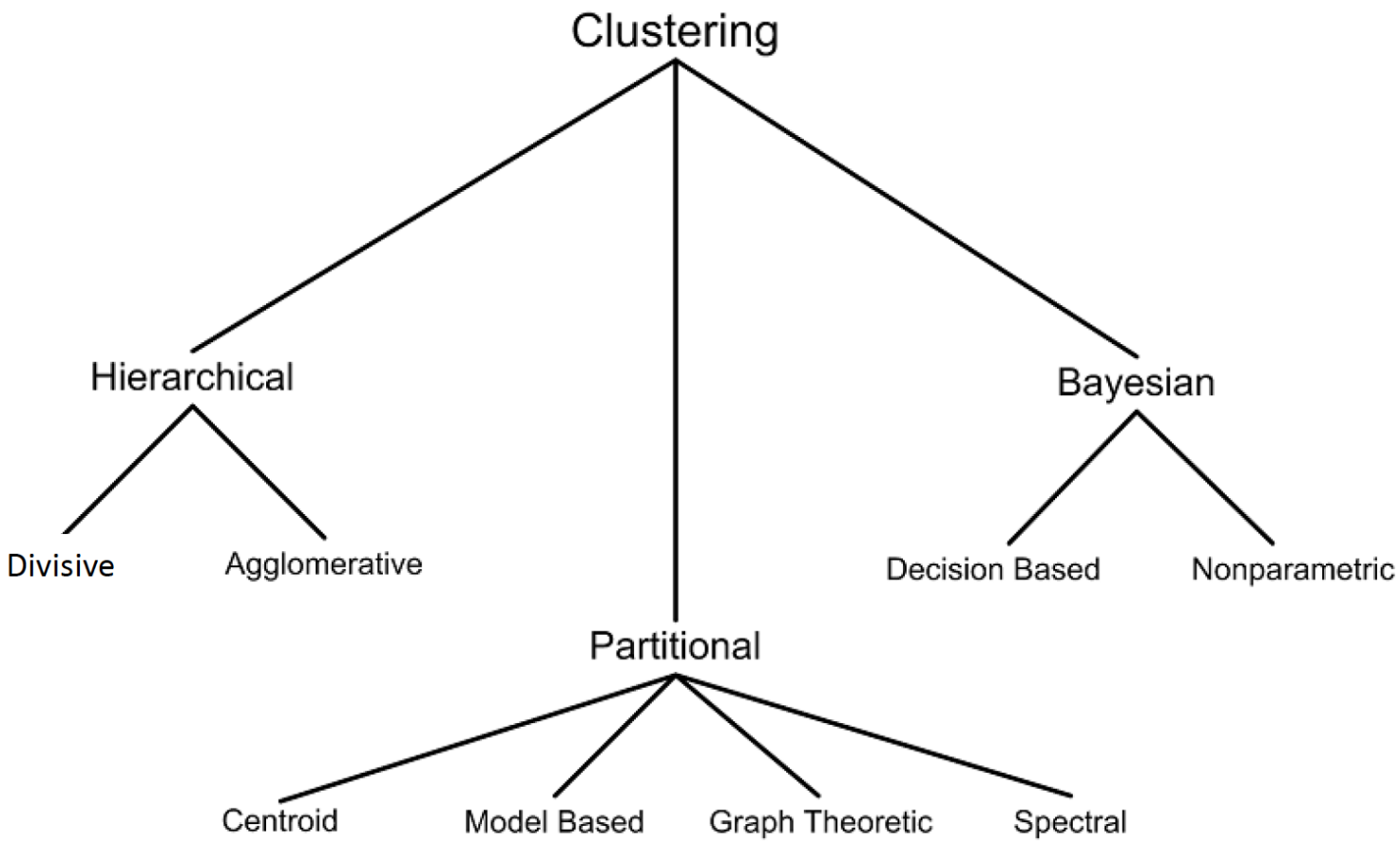


Figure: Good vs Bad clustering

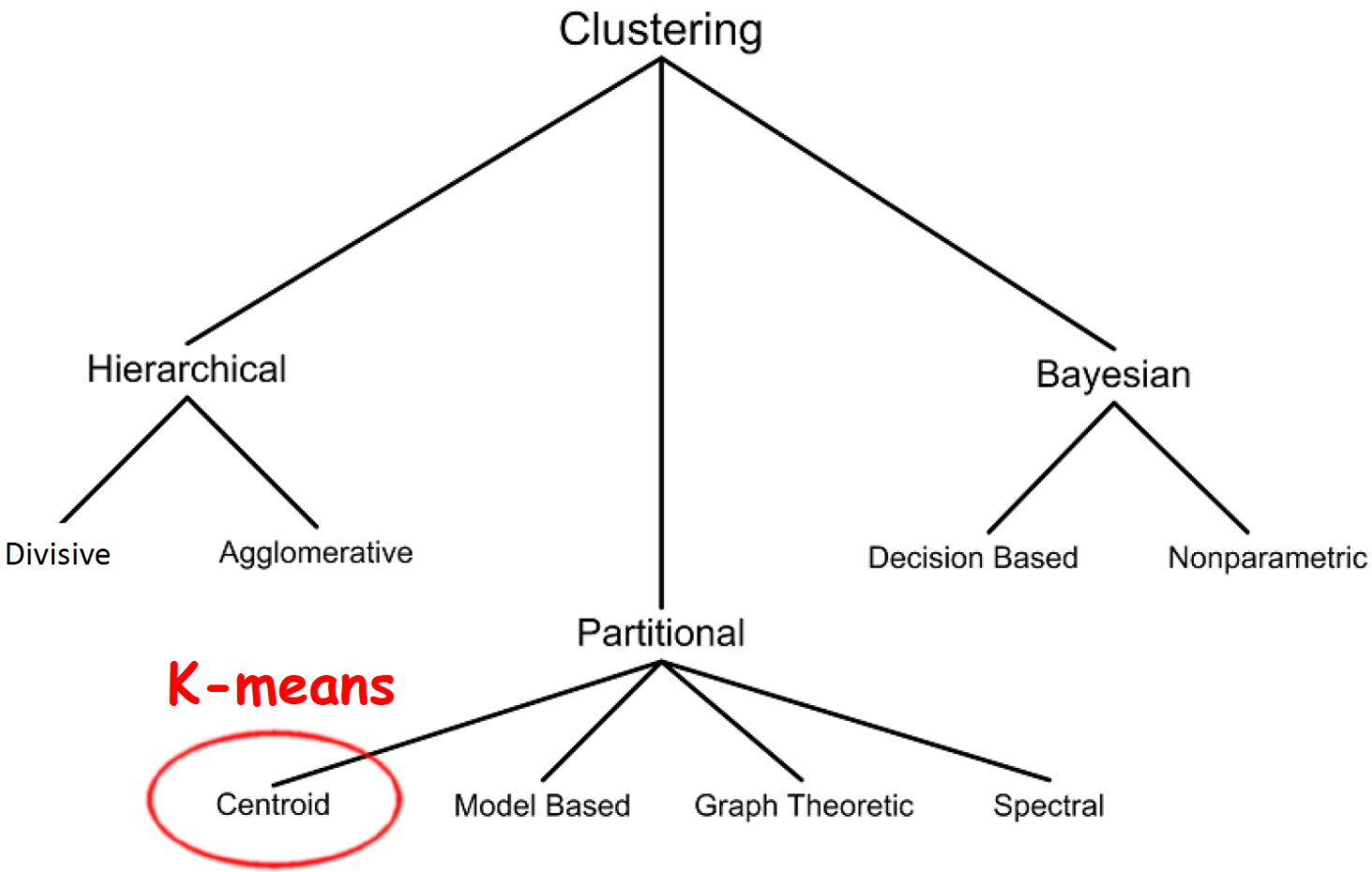
3. An algorithm to compute the clustering that is scalable, interpretable and does not rely on domain knowledge

Outline

Algorithms


Clustering
Hierarchical
Bayesian
Divisive
Agglomerative
Decision Based
Nonparametric
Partitional
Centroid
Model Based
Graph Theoretic
Spectral

Algorithm K-means



Algorithm K-means

- Setup Assume a collection of N data points in some D-dim space
- Objective Group similar data points together and discover underlying patterns by looking for a fixed number of clusters K in a dataset
- Assumption The number of clusters K is a hyperparameter that must be selected by the user
- How Given K the algorithm searches for K centroids, that are centres of each cluster
  Each data point is allocated to a cluster according to its distance from each centroid

Algorithm K-means

- Setup Assume a collection of N data points in some D-dim space
- Objective Group similar data points together and discover underlying patterns by looking for a fixed number of clusters K in a dataset
- Assumption The number of clusters K is a hyperparameter that must be selected by the user
- How Given K the algorithm searches for K centroids, that are centres of each cluster
  Each data point is allocated to a cluster according to its distance from each centroid

Algorithm K-means

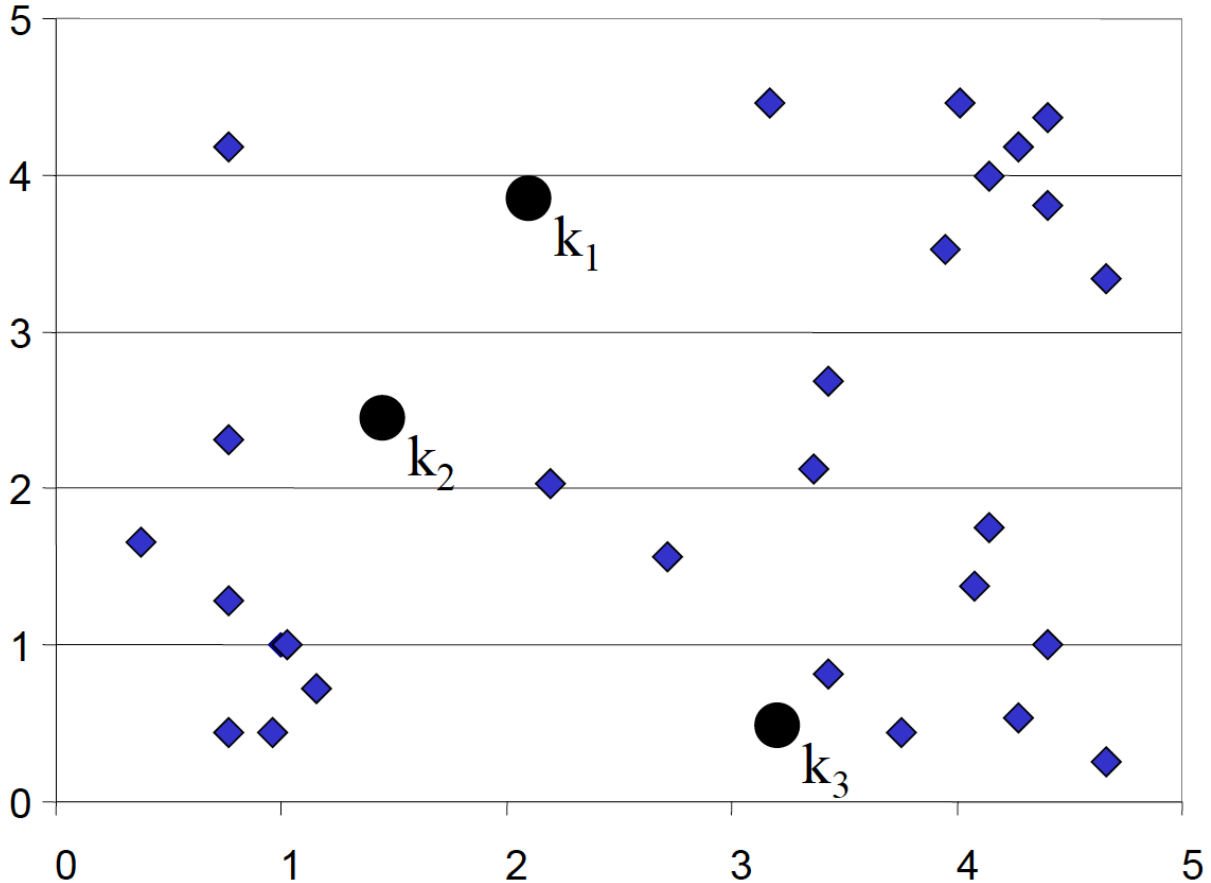
- Setup Assume a collection of N data points in some D-dim space
- Objective Group similar data points together and discover underlying patterns by looking for a fixed number of clusters K in a dataset
- Assumption The number of clusters K is a hyperparameter that must be selected by the user
- How Given K the algorithm searches for K centroids, that are centres of each cluster
  Each data point is allocated to a cluster according to its distance from each centroid

Algorithm K-means

- Setup Assume a collection of N data points in some D-dim space
- Objective Group similar data points together and discover underlying patterns by looking for a fixed number of clusters K in a dataset
- Assumption The number of clusters K is a hyperparameter that must be selected by the user
- How Given K the algorithm searches for K centroids, that are centres of each cluster
  Each data point is allocated to a cluster according to its distance from each centroid

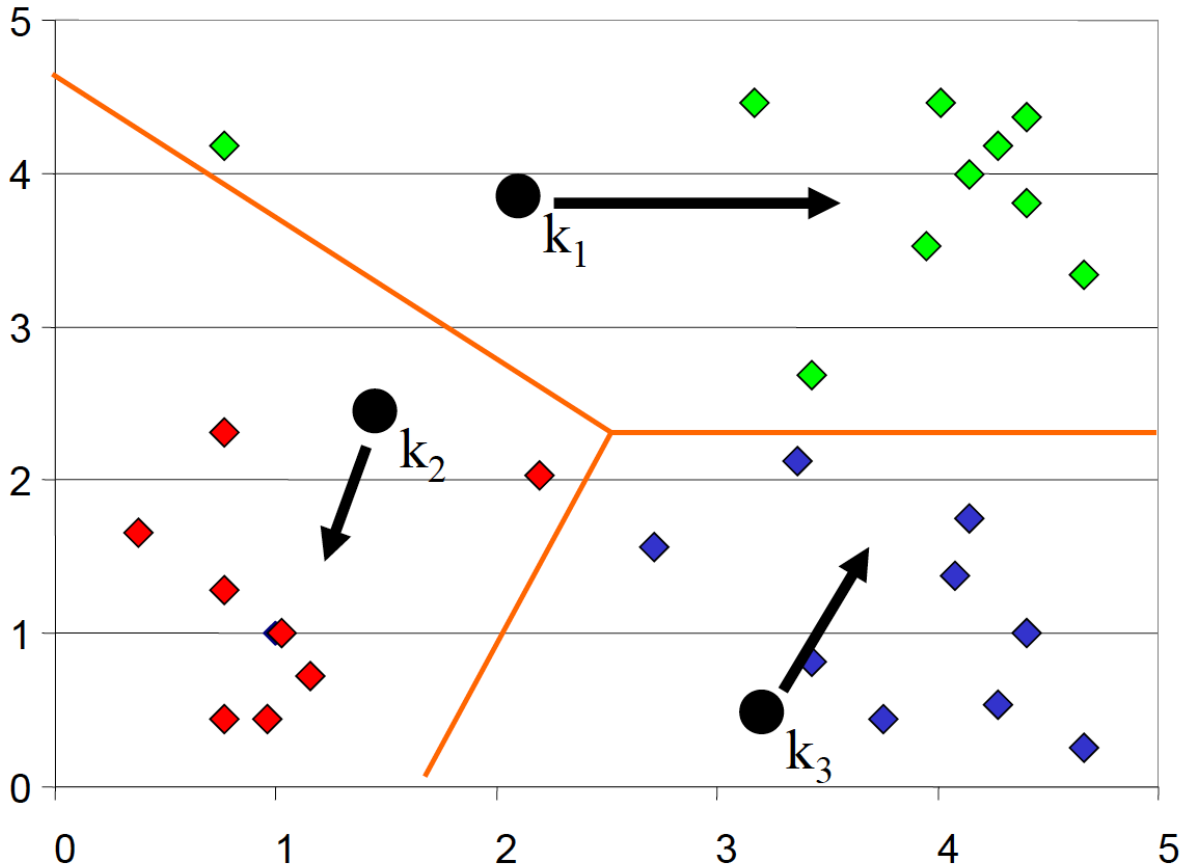
K-means clustering: initialisation

Decide ***K*** and initialise the ***K*** centres randomly in the space



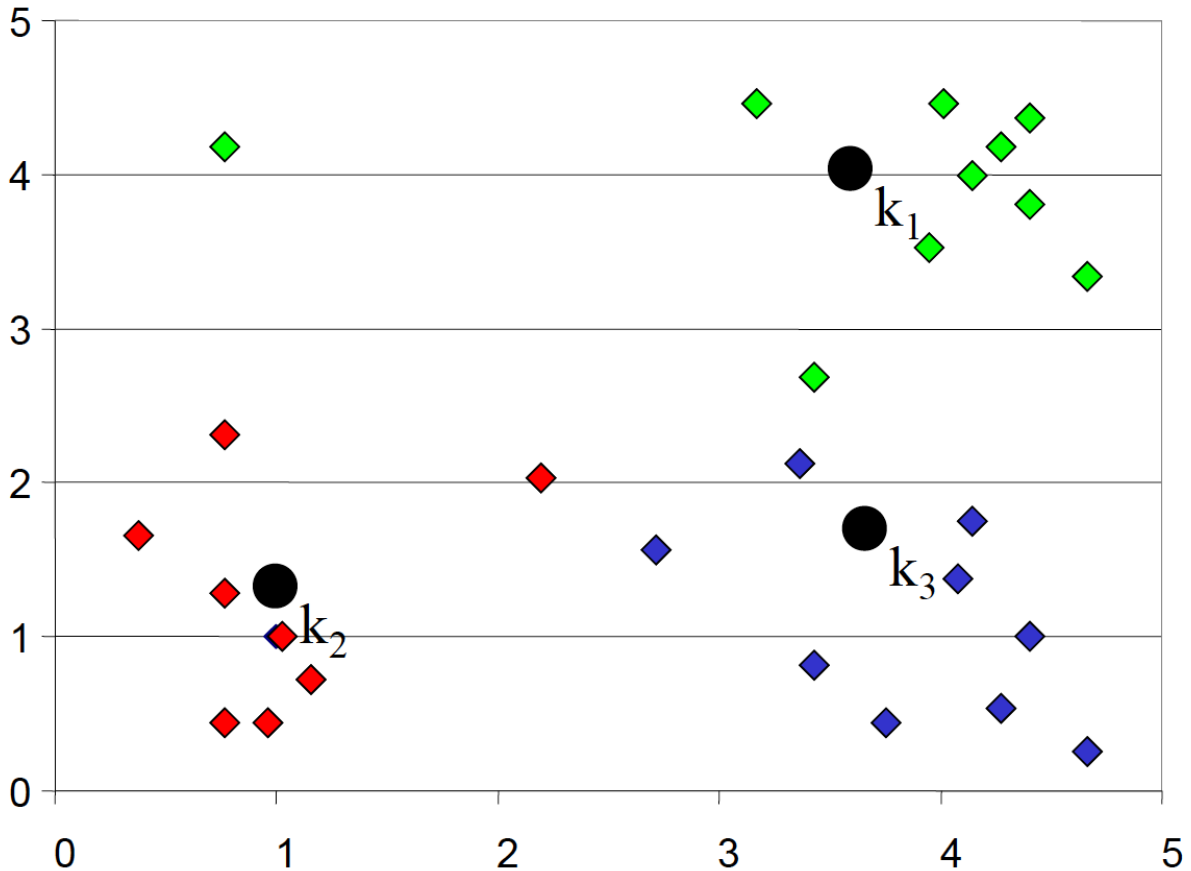
K-means clustering: iteration 1

- Run over all the datapoints to find their nearest centre and therefore assign them to the corresponding cluster
- Move the centers to the mean of its members



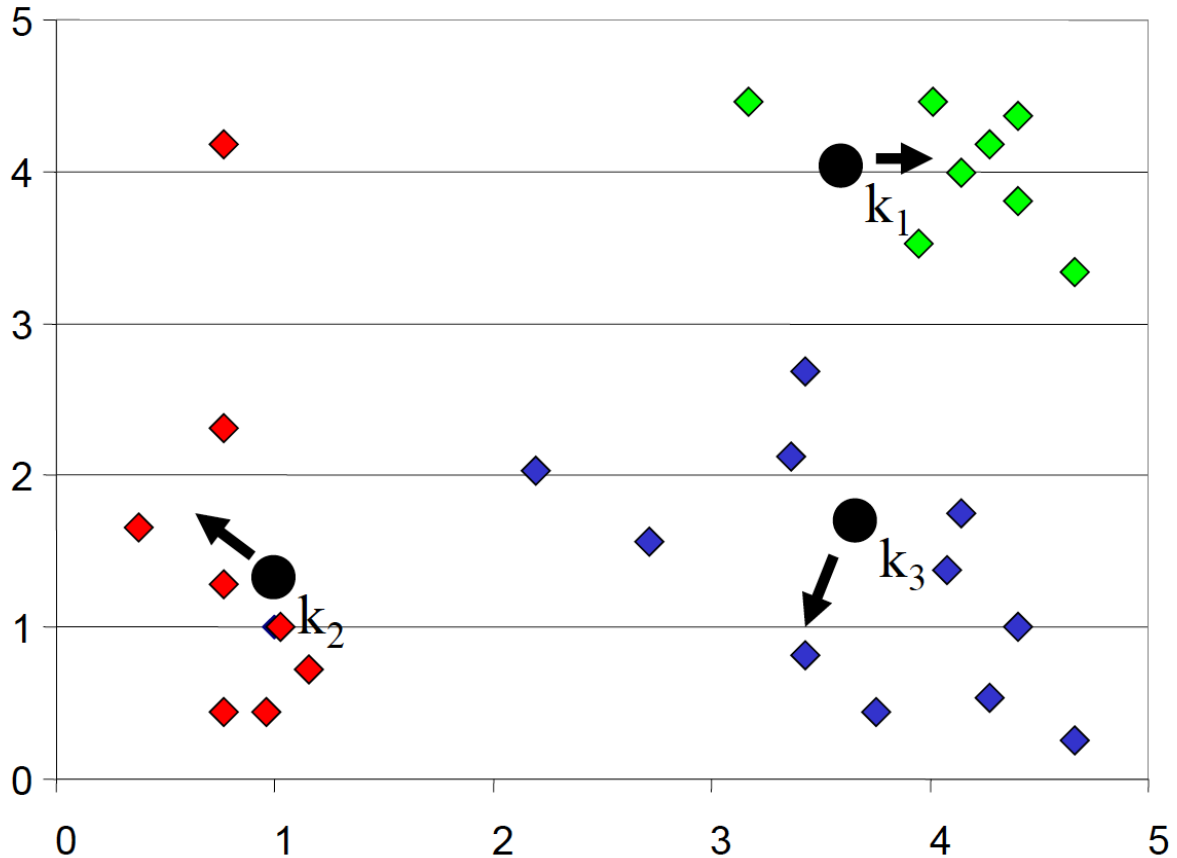
K-means clustering: iteration 2

After moving centers reassign objects



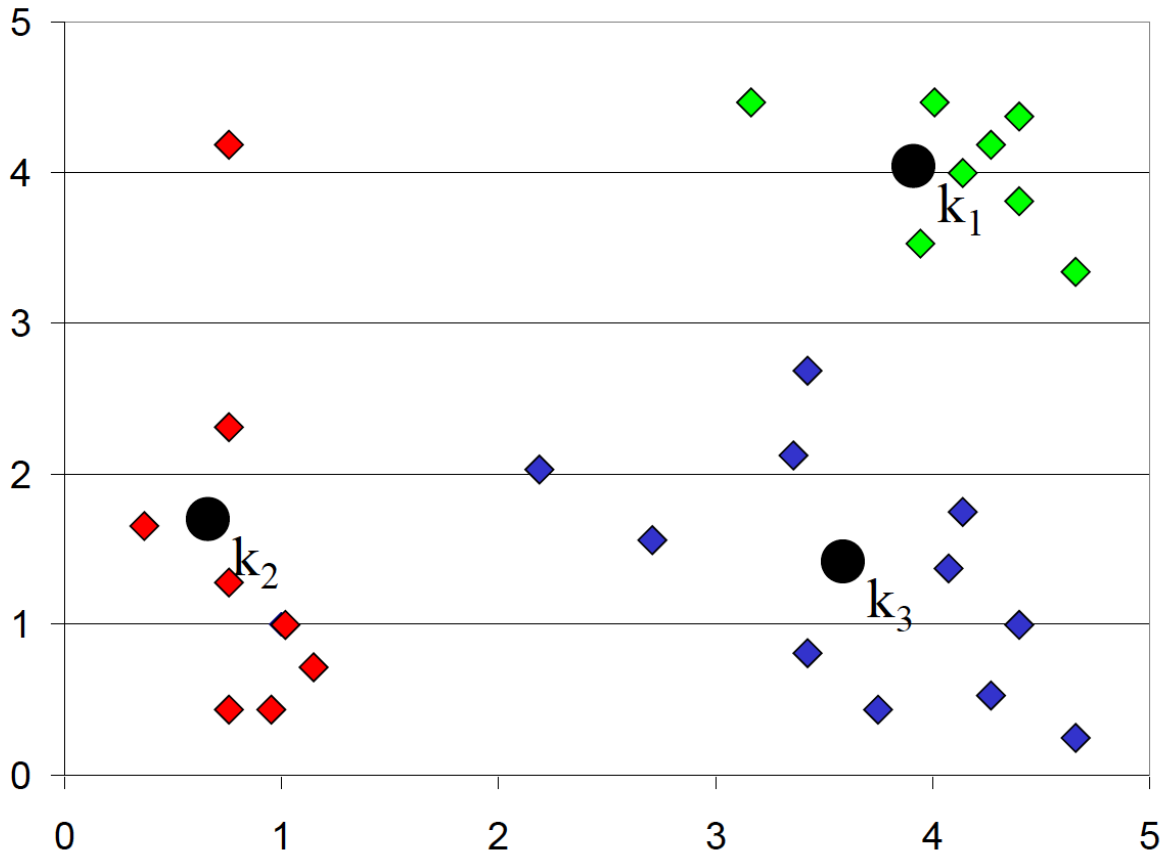
K-means clustering: iteration 2

After moving centers reassign objects to the nearest centers
Move a center to the mean of its new members



K-means clustering: finished

Reassign and move the centers, until no objects changed membership



K-means Algorithm

Given K, the K-means algorithm works as follows:

1. Choose K (random) data points to be the initial centroids, cluster centers
2. Assign each data point to the closest centroid

   Use a distance function as discussed earlier!
3. Re-compute the centroid susing the current cluster memberships

   Average/median of class members!
4. Repeat steps **2** and **3** until none of the N objects changed membership in the last iteration

K-means in image analysis

Objective: partition the image into regions, each of which has a reasonably homogeneous visual appearance

After clustering, each data point is replaced by its cluster mean

For a given K the algorithm represents the image using only K colours

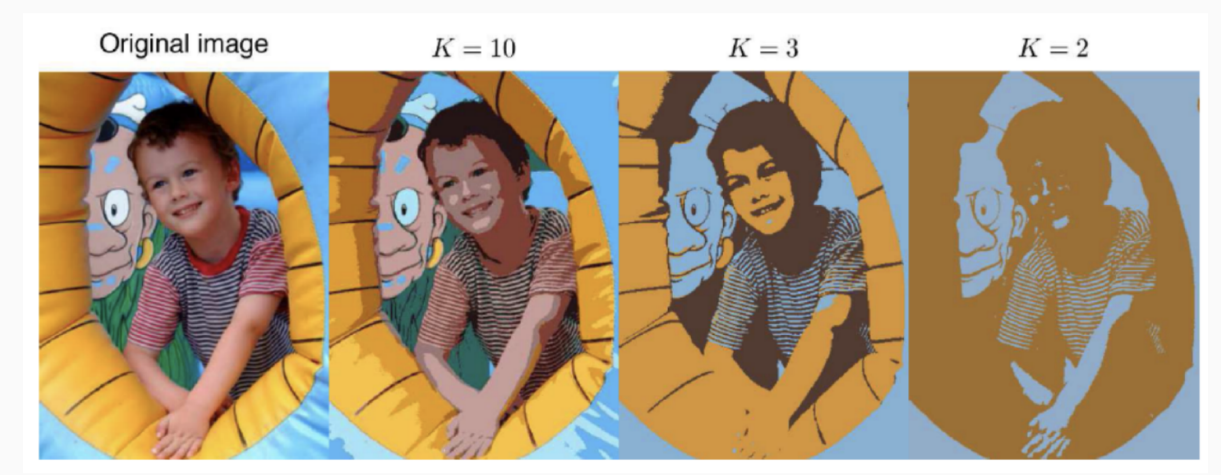


Figure: Bishop, C.M., 2009, Pattern Recognition and Machine Learning (Chapter 9), Springer, USA

K-means in image analysis

Objective: partition the image into regions, each of which has a reasonably homogeneous visual appearance
After clustering, each data point is replaced by its cluster mean
For a given K the algorithm represents the image using only K colours

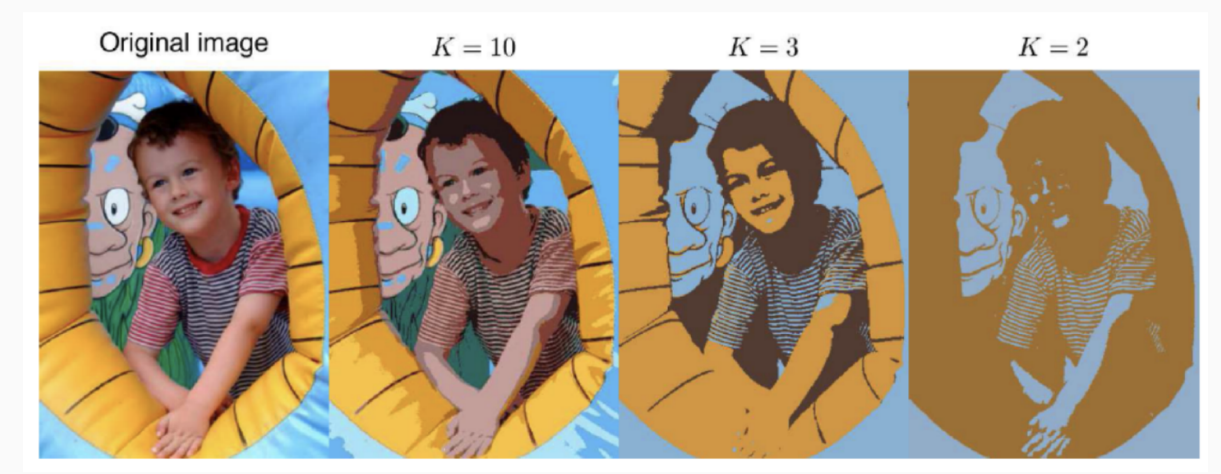


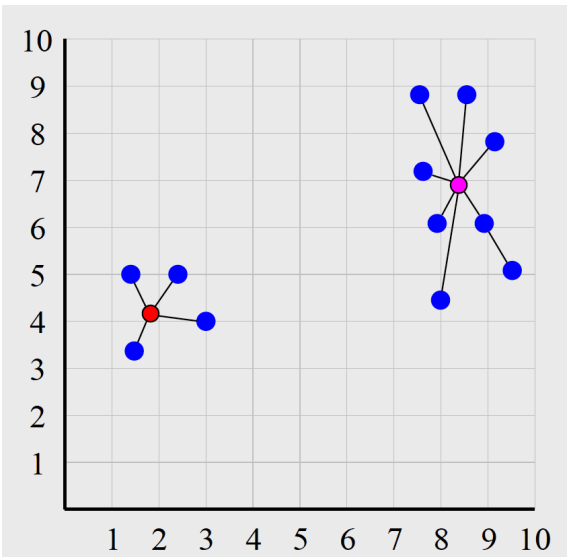
Figure: Bishop, C.M., 2009, Pattern Recognition and Machine Learning (Chapter 9), Springer, USA

Why does K-means actually work?

- It ensures high intra-cluster similarity
- K-means optimises
  - the average distance to members of the same cluster

$$\sum_{k=1}^{K} \frac{1}{n_k} \sum_{i=1}^{n_k} \sum_{j=1}^{n_k} |x_{ki} - x_{kj}|^2$$

  - which is twice the total distance to the centers, also called squared error $\sum_{k=1}^{K} \sum_{i=1}^{n_k} ||x_{ki} - \mu_k||^2$



Overview of K-means

- Very popular!
- Strengths
  - simple and easy to implement
  - efficient $O(tKN)$ for N objects, K clusters and t iterations
- Weaknesses
  - need to specify K in advance
  - categorical data?
  - terminates at a local optimum
  - initalisation matters
  - sensitive to outliers or noisy data
  - not suitable to discover clusters with non-convex shapes

Overview of K-means

- Very popular!
- Strengths
  - simple and easy to implement
  - efficient $O(tKN)$ for N objects, K clusters and t iterations
- Weaknesses
  - need to specify K in advance
  - categorical data?
  - terminates at a local optimum
  - initalisation matters
  - sensitive to outliers or noisy data
  - not suitable to discover clusters with non-convex shapes

Overview of K-means

- Very popular!
- Strengths
  - simple and easy to implement
  - efficient $O(tKN)$ for N objects, K clusters and t iterations
- Weaknesses
  - need to specify K in advance
  - categorical data?
  - terminates at a local optimum
  - initalisation matters
  - sensitive to outliers or noisy data
  - not suitable to discover clusters with non-convex shapes

Overview of K-means

- Very popular!
- Strengths
  - simple and easy to implement
  - efficient $O(tKN)$ for N objects, K clusters and t iterations
- Weaknesses
  - need to specify K in advance
  - categorical data?
  - terminates at a local optimum
  - initalisation matters
  - sensitive to outliers or noisy data
  - not suitable to discover clusters with non-convex shapes

Overview of K-means

- Very popular!
- Strengths
  - simple and easy to implement
  - efficient $O(tKN)$ for N objects, K clusters and t iterations
- Weaknesses
  - need to specify K in advance
  - categorical data?
  - terminates at a local optimum
  - initalisation matters
  - sensitive to outliers or noisy data
  - not suitable to discover clusters with non-convex shapes

Overview of K-means

- Very popular!
- Strengths
  - simple and easy to implement
  - efficient $O(tKN)$ for N objects, K clusters and t iterations
- Weaknesses
  - need to specify K in advance
  - categorical data?
  - terminates at a local optimum
  - initalisation matters
  - sensitive to outliers or noisy data
  - not suitable to discover clusters with non-convex shapes

Overview of K-means

- Very popular!
- Strengths
  - simple and easy to implement
  - efficient $O(tKN)$ for N objects, K clusters and t iterations
- Weaknesses
  - need to specify K in advance
  - categorical data?
  - terminates at a local optimum
  - initalisation matters
  - sensitive to outliers or noisy data
  - not suitable to discover clusters with non-convex shapes

Overview of K-means

- Very popular!
- Strengths
  - simple and easy to implement
  - efficient $O(tKN)$ for N objects, K clusters and t iterations
- Weaknesses
  - need to specify K in advance
  - categorical data?
  - terminates at a local optimum
  - initalisation matters
  - sensitive to outliers or noisy data
  - not suitable to discover clusters with non-convex shapes

Overview of K-means

- Very popular!
- Strengths
  - simple and easy to implement
  - efficient $O(tKN)$ for N objects, K clusters and t iterations
- Weaknesses
  - need to specify K in advance
  - categorical data?
  - terminates at a local optimum
  - initalisation matters
  - sensitive to outliers or noisy data
  - not suitable to discover clusters with non-convex shapes

Overview of K-means

- Very popular!
- Strengths
  - simple and easy to implement
  - efficient $O(tKN)$ for N objects, K clusters and t iterations
- Weaknesses
  - need to specify K in advance
  - categorical data?
  - terminates at a local optimum
  - initalisation matters
  - sensitive to outliers or noisy data
  - not suitable to discover clusters with non-convex shapes

Overview of K-means

- Very popular!
- Strengths
  - simple and easy to implement
  - efficient $O(tKN)$ for N objects, K clusters and t iterations
- Weaknesses
  - need to specify K in advance
  - categorical data?
  - terminates at a local optimum
  - initalisation matters
  - sensitive to outliers or noisy data
  - not suitable to discover clusters with non-convex shapes

Outliers and initialisation

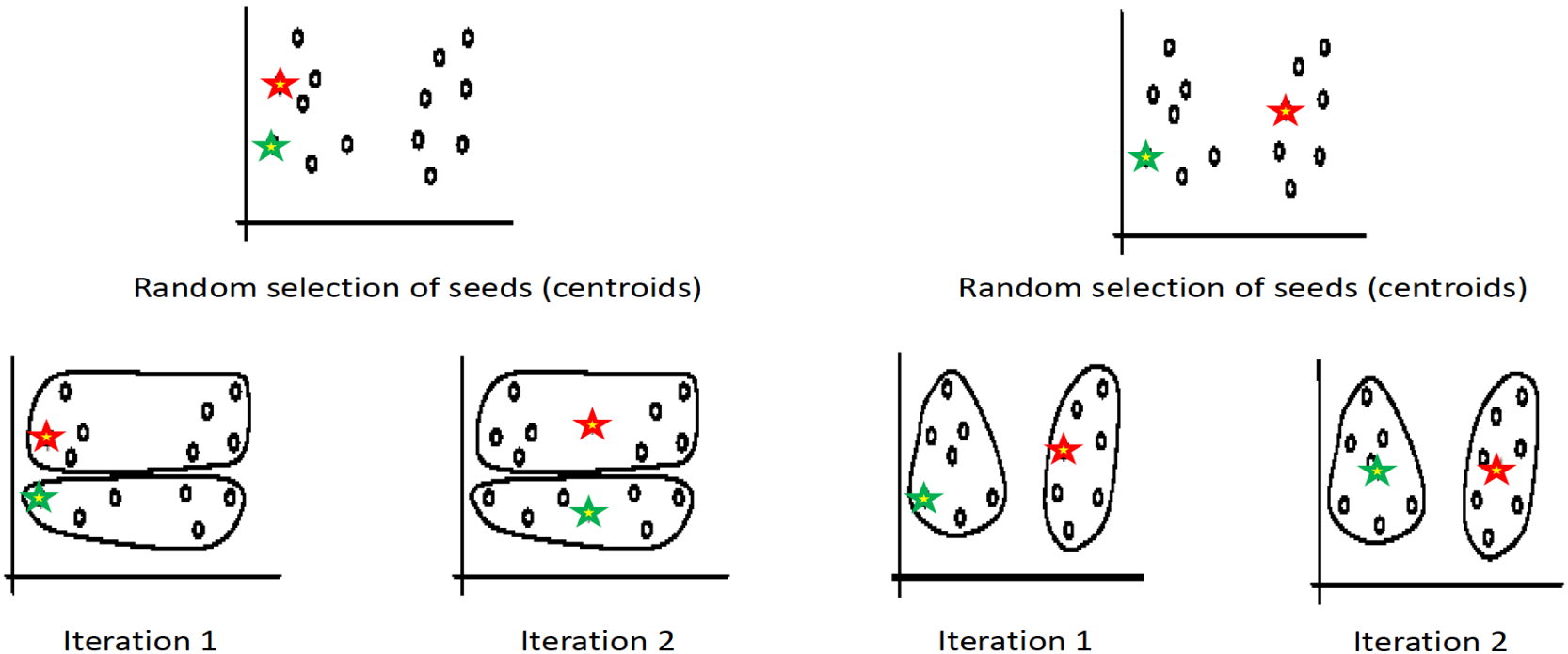
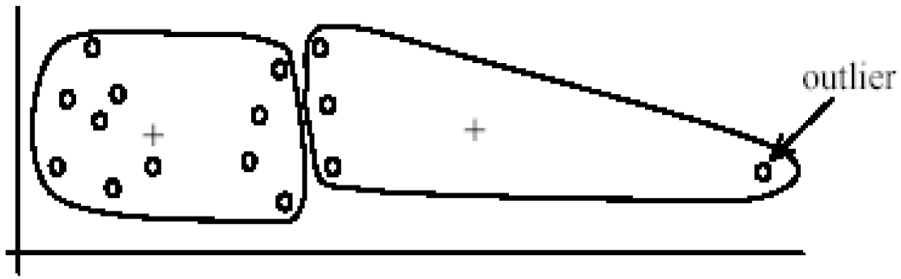
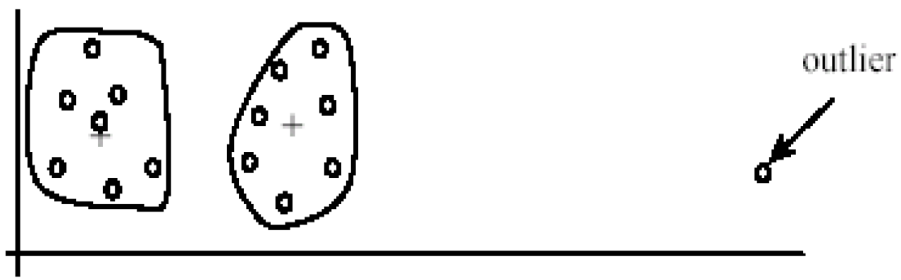


Figure: Sensitivity to initialisation

Outliers and K-means



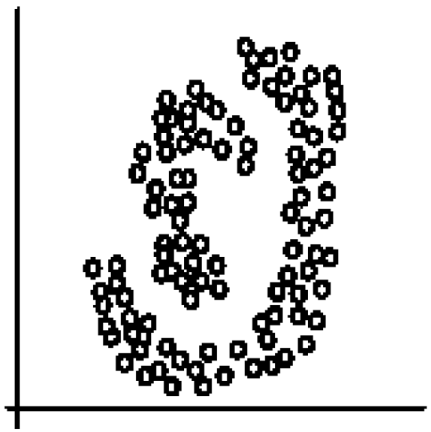
(A): Undesirable clusters



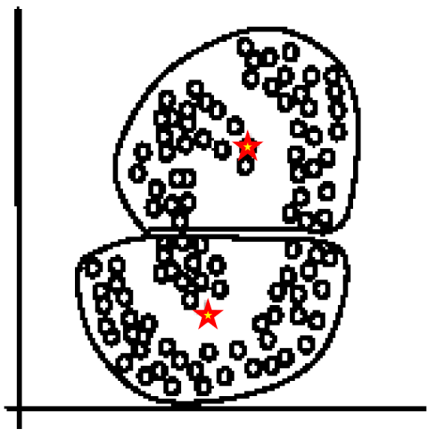
(B): Ideal clusters

- Can remove points that are very far from the centroids relatively to other points
- Perform random sampling by choosing a small subset of the data points

Outliers and non-convex shapes



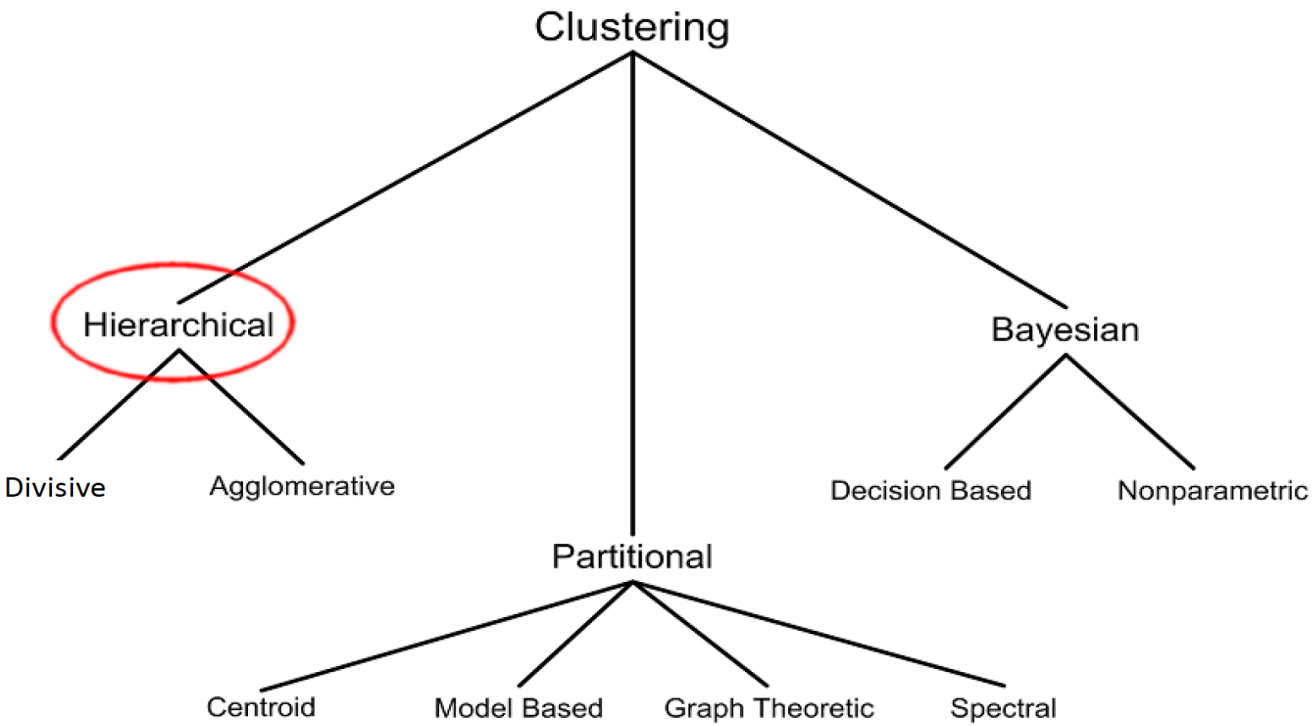
(A): Two natural clusters



(B): k-means clusters

The ***K***-means algorithm is not suitable for discovering clusters that are not hyper-ellipsoids (or hyper-spheres)

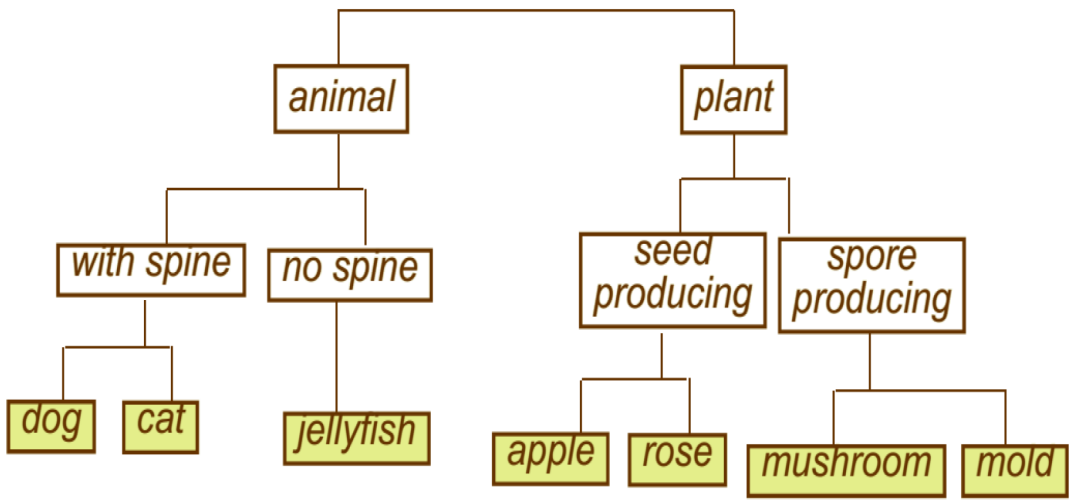
Hierarchical clustering



Outline

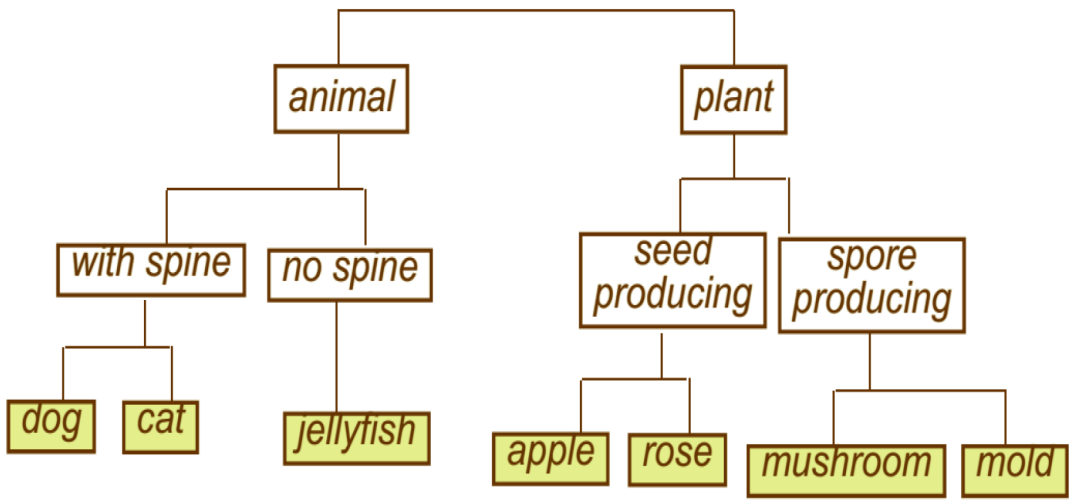
Hierarchical Clustering

- Produces a set of nested clusters organised as a hierarchical tree
- Can be visualised as a dendrogram: a tree like diagram that records the sequences of merges and splits
- Do not have to assume a particular number of clusters as you can cut the dendrogram wherever
- They may correspond to meaningful taxonomies



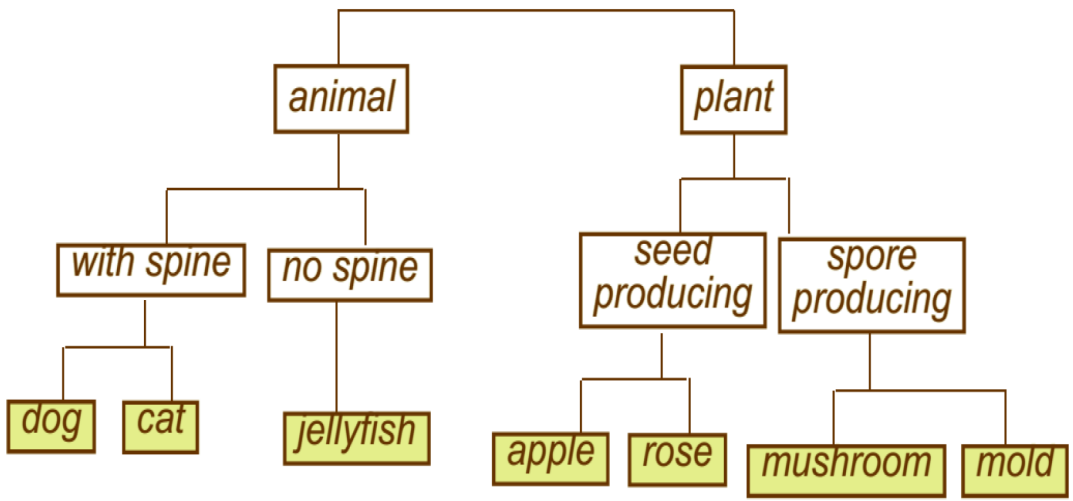
Hierarchical Clustering

- Produces a set of nested clusters organised as a hierarchical tree
- Can be visualised as a dendrogram: a tree like diagram that records the sequences of merges and splits
- Do not have to assume a particular number of clusters as you can cut the dendrogram wherever
- They may correspond to meaningful taxonomies



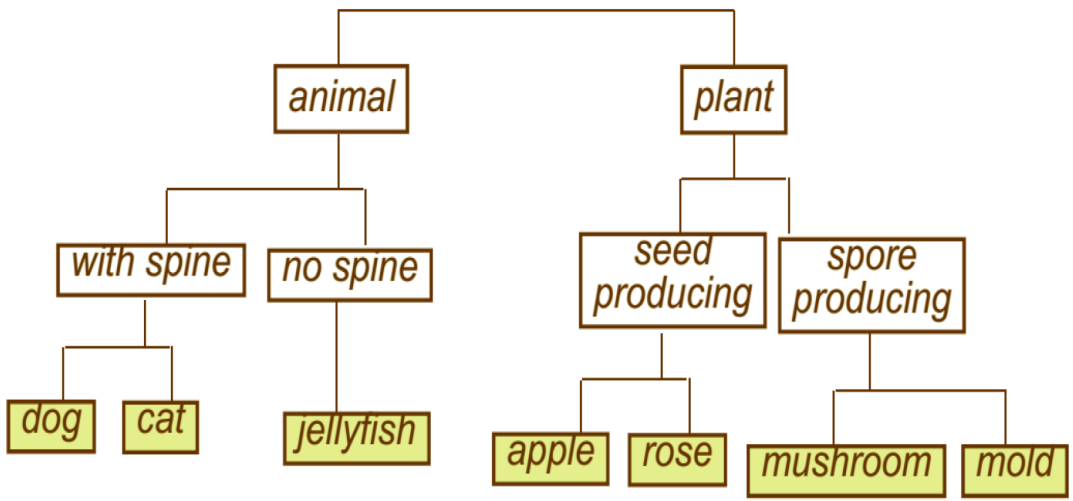
Hierarchical Clustering

- Produces a set of nested clusters organised as a hierarchical tree
- Can be visualised as a dendrogram: a tree like diagram that records the sequences of merges and splits
- Do not have to assume a particular number of clusters as you can cut the dendrogram wherever
- They may correspond to meaningful taxonomies



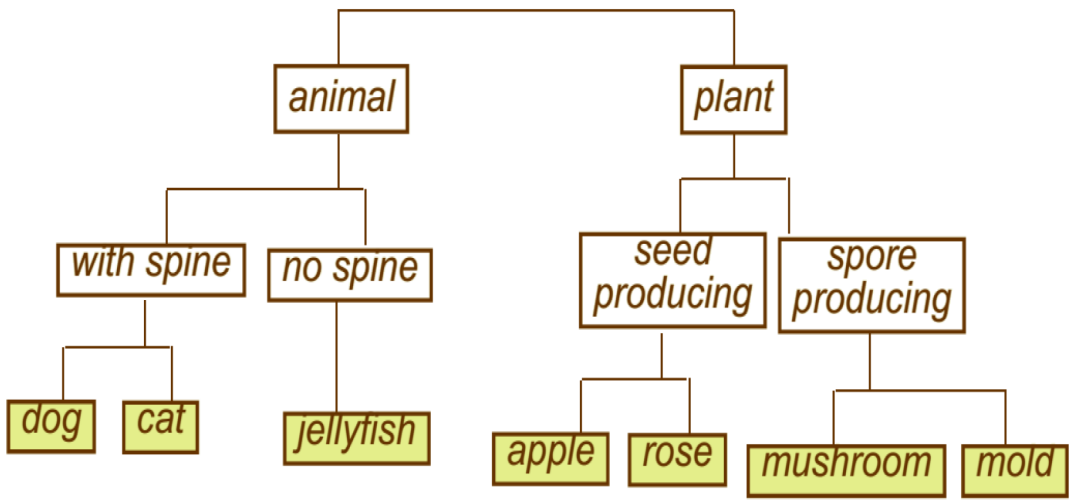
Hierarchical Clustering

- Produces a set of nested clusters organised as a hierarchical tree
- Can be visualised as a dendrogram: a tree like diagram that records the sequences of merges and splits
- Do not have to assume a particular number of clusters as you can cut the dendrogram wherever
- They may correspond to meaningful taxonomies



Hierarchical Clustering

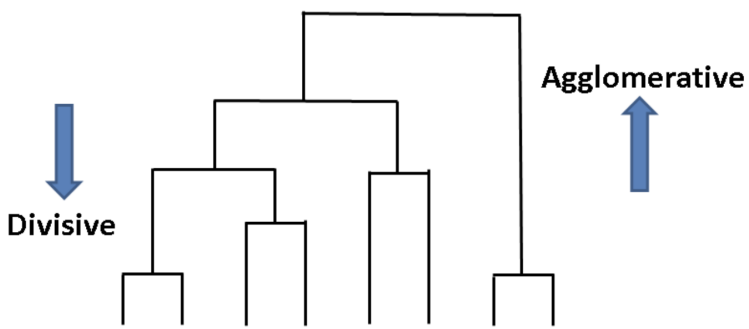
- Produces a set of nested clusters organised as a hierarchical tree
- Can be visualised as a dendrogram: a tree like diagram that records the sequences of merges and splits
- Do not have to assume a particular number of clusters as you can cut the dendrogram wherever
- They may correspond to meaningful taxonomies



Two types of hierarchical clustering

Hierarchical clustering builds a hierarchy of clusters and has two types

- Divisive or top down
  - One cluster containing all the observations
  - Recursively split one of the clusters at each level into two new clusters
- Agglomerative or bottom up
  - Each observation is first assigned to its own cluster
  - Then pairs of clusters are merged recursively until all the observations are grouped in one cluster

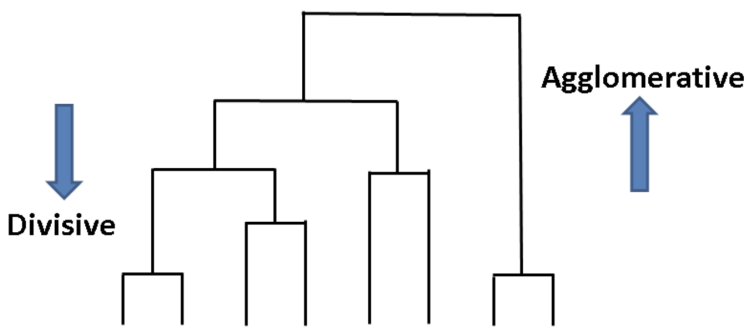


Use a distance matrix to decide whether to merge or split a cluster at a time

Two types of hierarchical clustering

Hierarchical clustering builds a hierarchy of clusters and has two types

- Divisive or top down
  - One cluster containing all the observations
  - Recursively split one of the clusters at each level into two new clusters
- Agglomerative or bottom up
  - Each observation is first assigned to its own cluster
  - Then pairs of clusters are merged recursively until all the observations are grouped in one cluster

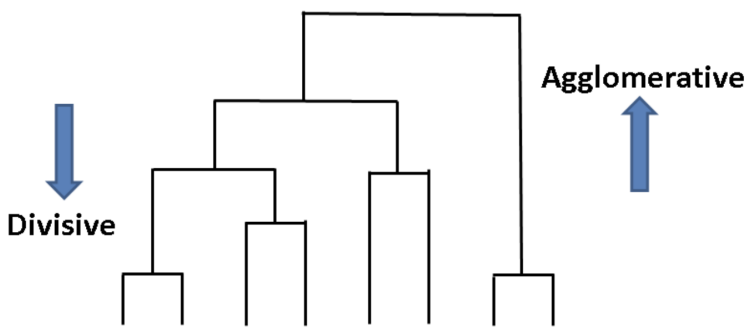


Use a distance matrix to decide whether to merge or split a cluster at a time

Two types of hierarchical clustering

Hierarchical clustering builds a hierarchy of clusters and has two types

- Divisive or top down
  - One cluster containing all the observations
  - Recursively split one of the clusters at each level into two new clusters
- Agglomerative or bottom up
  - Each observation is first assigned to its own cluster
  - Then pairs of clusters are merged recursively until all the observations are grouped in one cluster



Use a distance matrix to decide whether to merge or split a cluster at a time

Agglomerative Clustering Algorithm

The most popular hierarchical clustering technique and its idea is to ensure that nearby points end up in the same cluster

1. Start with a collection of singletons
   - Each cluster contains one data point
2. Repeat until only one cluster is left
   - Find a pair of clusters that is closest
   - Merge the two closest clusters into a new cluster
   - Remove the original clusters from the collection and add the new one

- Key operation is the computation of the proximity of two clusters
- Need to define a distance or proximity matrix
- There are different ways to define the distance between the clusters distinguishing the different algorithms

Agglomerative Clustering Algorithm

The most popular hierarchical clustering technique and its idea is to ensure that nearby points end up in the same cluster

1. Start with a collection of singletons
   - Each cluster contains one data point
2. Repeat until only one cluster is left
   - Find a pair of clusters that is closest
   - Merge the two closest clusters into a new cluster
   - Remove the original clusters from the collection and add the new one

- Key operation is the computation of the proximity of two clusters
- Need to define a distance or proximity matrix
- There are different ways to define the distance between the clusters distinguishing the different algorithms

Agglomerative Clustering Algorithm

The most popular hierarchical clustering technique and its idea is to ensure that nearby points end up in the same cluster

1. Start with a collection of singletons
   - Each cluster contains one data point
2. Repeat until only one cluster is left
   - Find a pair of clusters that is closest
   - Merge the two closest clusters into a new cluster
   - Remove the original clusters from the collection and add the new one

- Key operation is the computation of the proximity of two clusters
- Need to define a distance or proximity matrix
- There are different ways to define the distance between the clusters distinguishing the different algorithms

Agglomerative Clustering Algorithm

The most popular hierarchical clustering technique and its idea is to ensure that nearby points end up in the same cluster

1. Start with a collection of singletons
   - Each cluster contains one data point
2. Repeat until only one cluster is left
   - Find a pair of clusters that is closest
   - Merge the two closest clusters into a new cluster
   - Remove the original clusters from the collection and add the new one

- Key operation is the computation of the proximity of two clusters
- Need to define a distance or proximity matrix
- There are different ways to define the distance between the clusters distinguishing the different algorithms

Agglomerative Clustering Algorithm

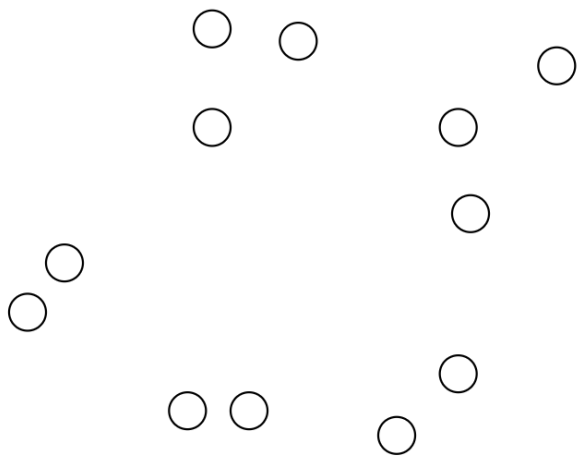
The most popular hierarchical clustering technique and its idea is to ensure that nearby points end up in the same cluster

1. Start with a collection of singletons
   - Each cluster contains one data point
2. Repeat until only one cluster is left
   - Find a pair of clusters that is closest
   - Merge the two closest clusters into a new cluster
   - Remove the original clusters from the collection and add the new one

- Key operation is the computation of the proximity of two clusters
- Need to define a distance or proximity matrix
- There are different ways to define the distance between the clusters distinguishing the different algorithms

Starting Step

Start with clusters of individual points and a proximity matrix

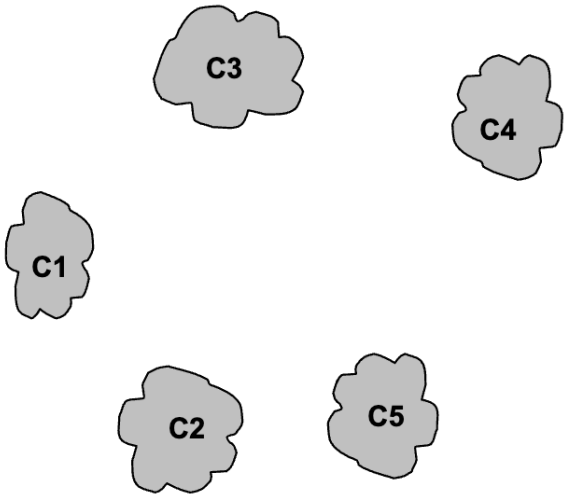


	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

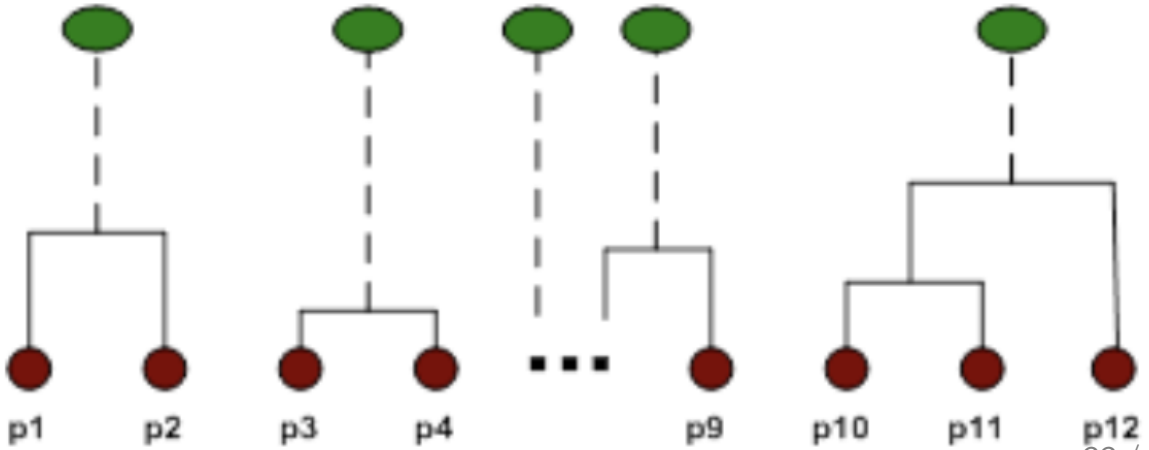
Intermediate Situation

After some merging steps we have some clusters



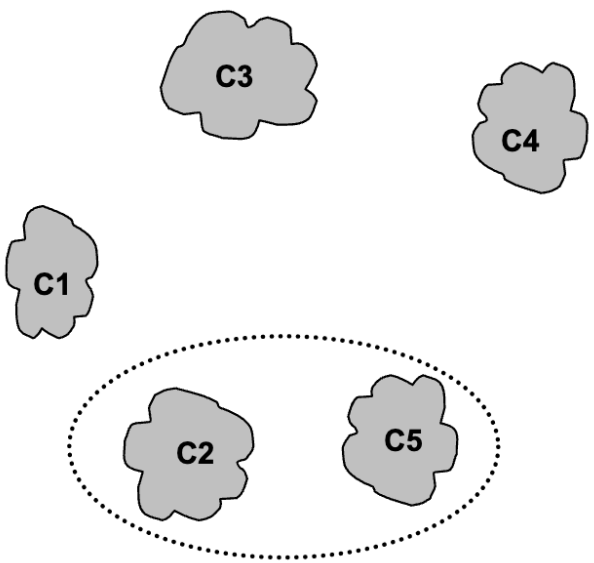
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



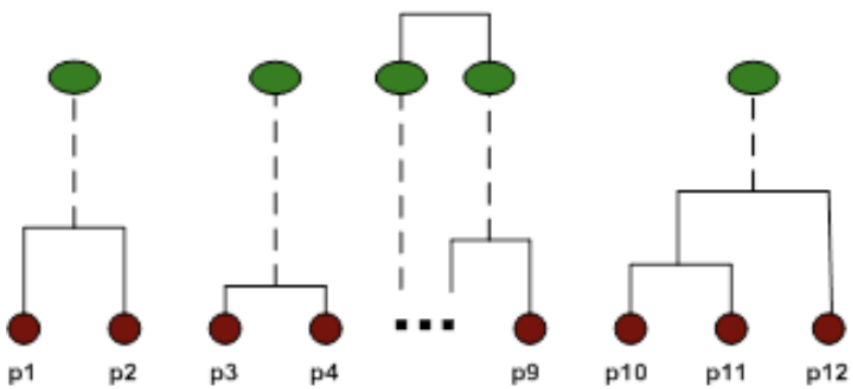
Intermediate Step

We want to merge the two closest clusters (C2 and C5) and update the proximity matrix



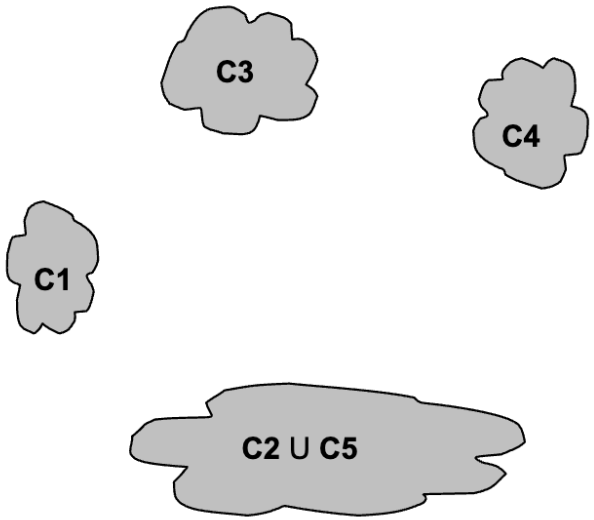
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Proximity Matrix



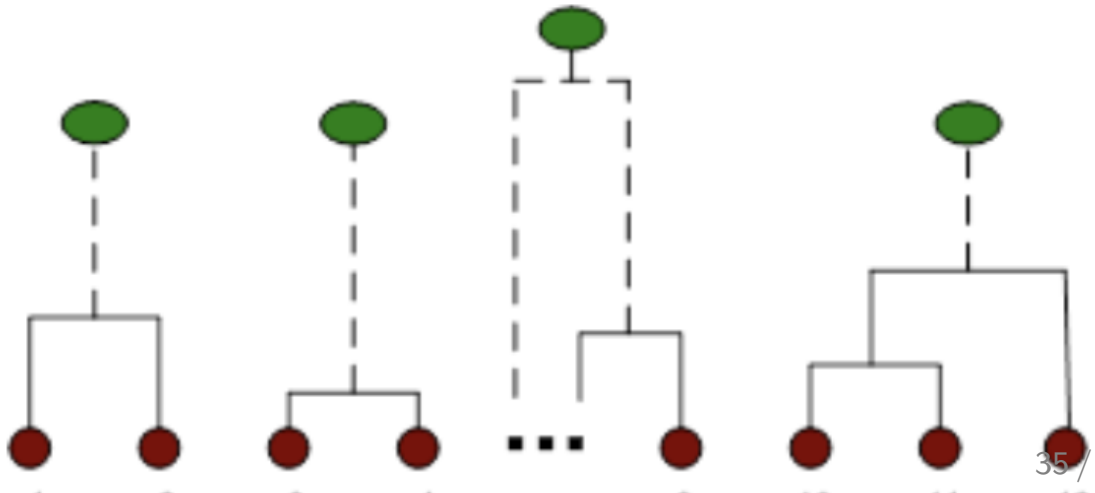
After merging

How do we actually fill in the proximity matrix?

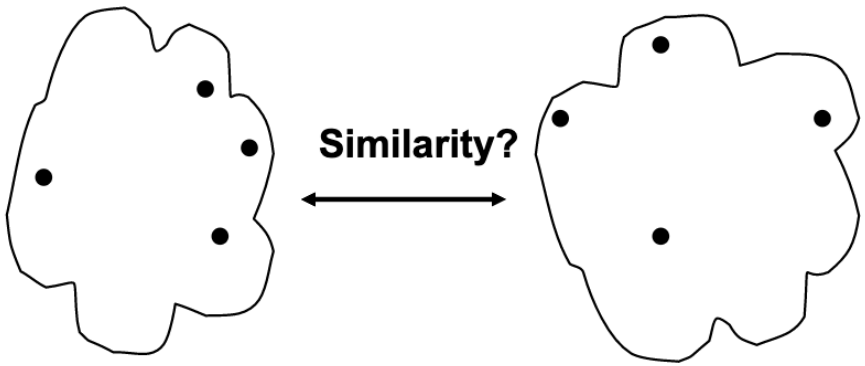


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Proximity Matrix

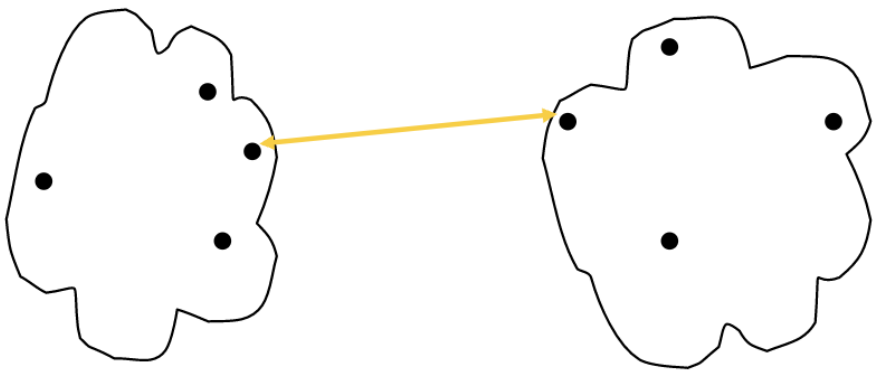


How to define inter-cluster similarity



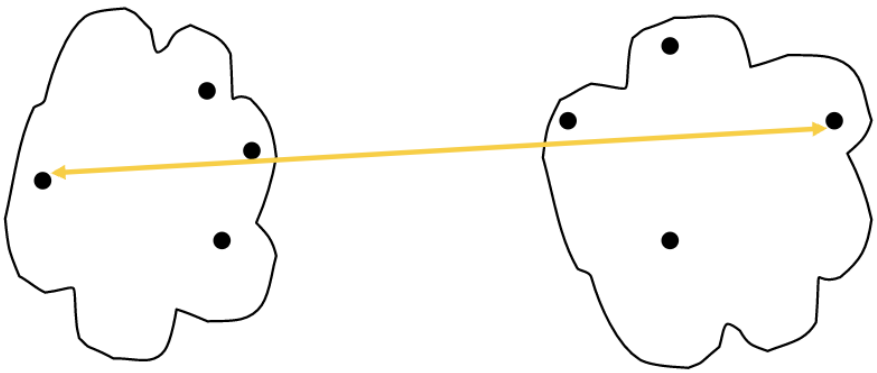
- MIN
- MAX
- Group Average
- Distance between centroids
- other methods driven by an objective function

How to define inter-cluster similarity



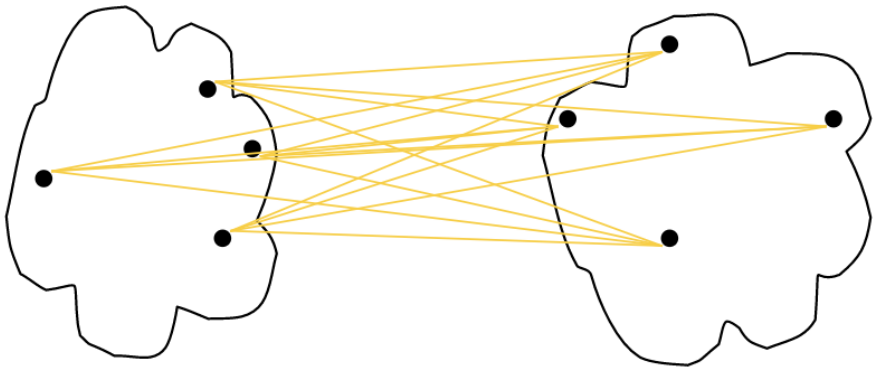
- MIN
- MAX
- Group Average
- Distance between centroids
- other methods driven by an objective function

How to define inter-cluster similarity



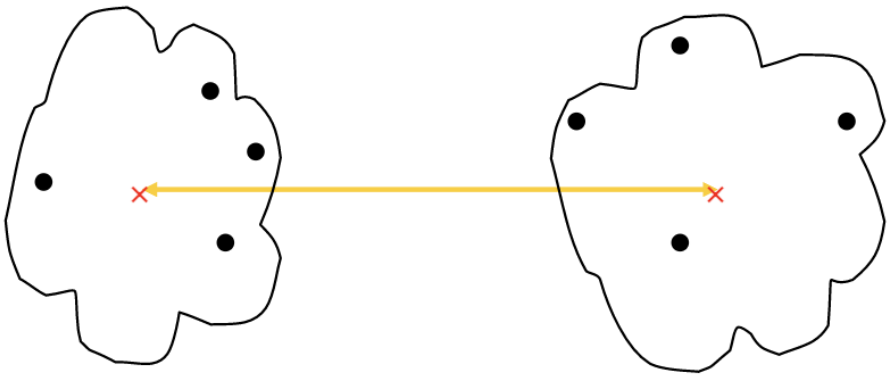
- MIN
- MAX
- Group Average
- Distance between centroids
- other methods driven by an objective function

How to define inter-cluster similarity



- MIN
- MAX
- Group Average
- Distance between centroids
- other methods driven by an objective function

How to define inter-cluster similarity



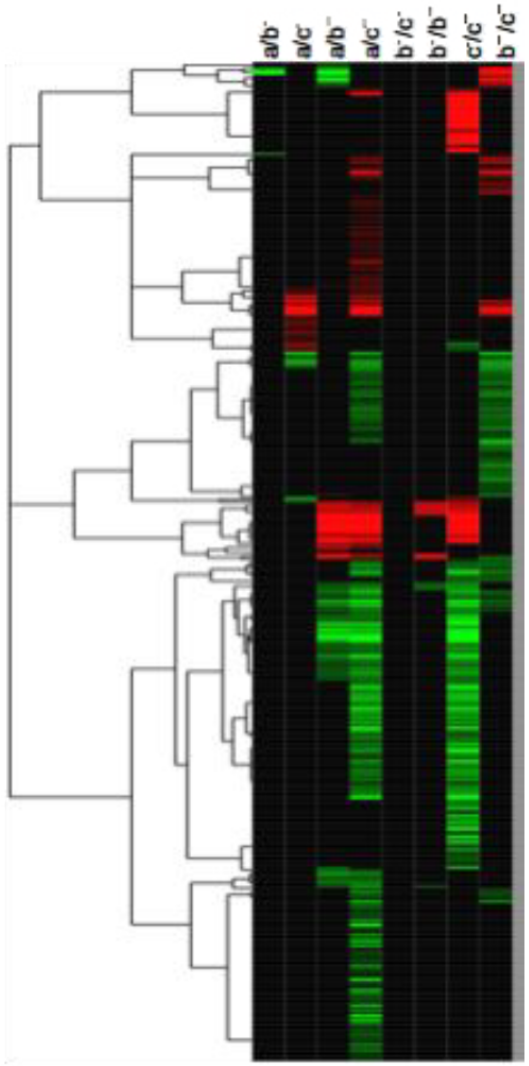
- MIN
- MAX
- Group Average
- Distance between centroids
- Other methods driven by an objective function

Weakness and strengths

- Usually the complexity is at least $O(N^2)$
- Once a decision is made to combine two clusters it cannot be undone
- The clusters are subjective
- No objective function is directly minimised
- Given the distance measure the results, strengths and weaknesses are different
  - sensitivity to noise and outliers
  - difficulty handling different sized clusters and shapes
  - breaking large clusters

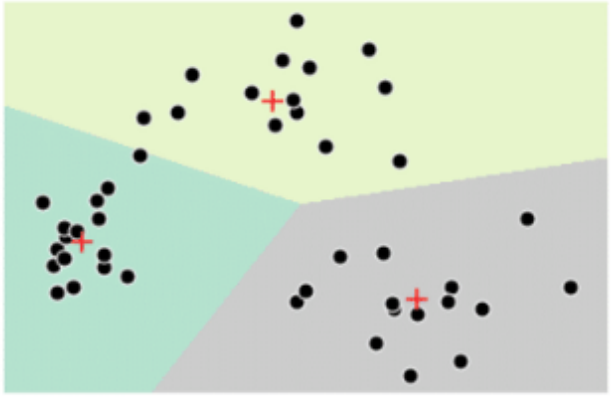
Example on gene segmentation

- Microarrays measure the activities of all genes in different conditions
- Clustering genes can help determine new functions for unknown genes
- One of early applications of clustering

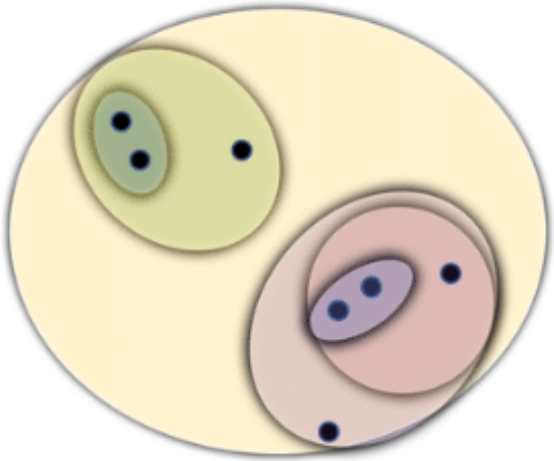


Comparison

	Hierarchical	**K-means**
Running time	naively, $O(N^3)$	fastest (each iteration is linear)
Assumptions	requires a similarity / distance measure	strong assumptions
Input parameters	none	K (number of clusters)
Clusters	subjective (only a tree is returned)	exactly K clusters



K-means clustering



Hierarchical clustering

Summary

- Clustering has a long history and still is in active research
- There are many more clustering algorithms: Density based, Neural networks based, Co-clustering...
- Clustering is hard to evaluate, but very useful in practice
- Clustering is highly application dependent (and to some extent subjective)
- You should know by now what clustering is and why it is important and be able to implement a clustering method depending on the context