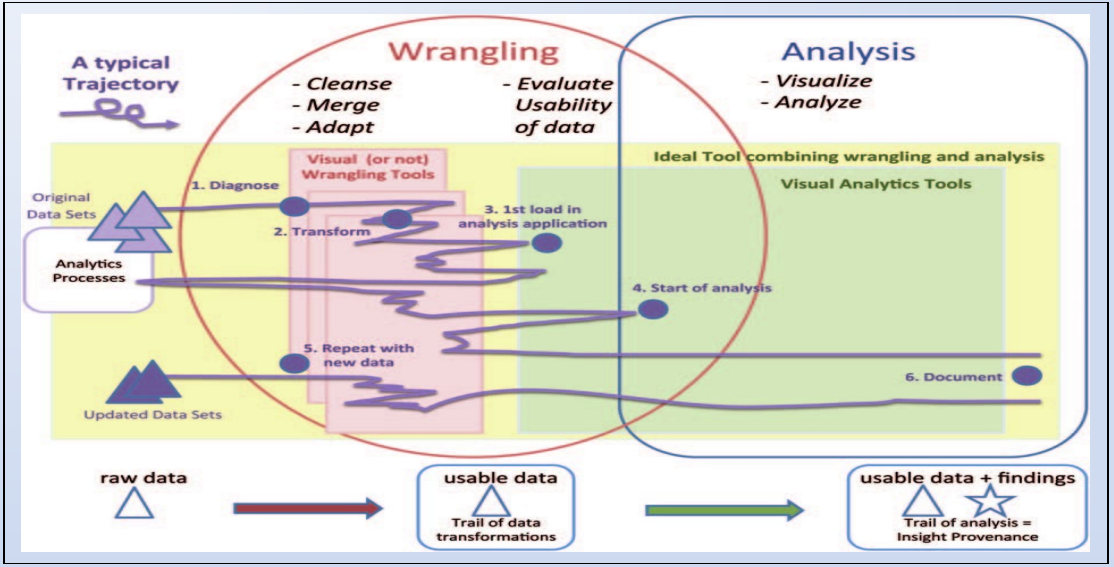
COMP20008 Elements of Data Processing

Clustering

Semester 1, 2020

Contact: uwe.aickelin@unimelb.edu.au

Where are we now?



Outline

- Clustering algorithms
  - K-means
  - Visualisation of clustering tendency (VAT)
  - Hierarchical clustering

Data in higher dimensions

- Difficult to visualise

- How can we determine what the significant groups / segments / communities are?
  - Can understand the data better
  - Apply separate interventions to each group (e.g. marketing campaign)

What is Cluster Analysis?

We will be looking at two classic clustering algorithms

- K-means
- Hierarchical clustering

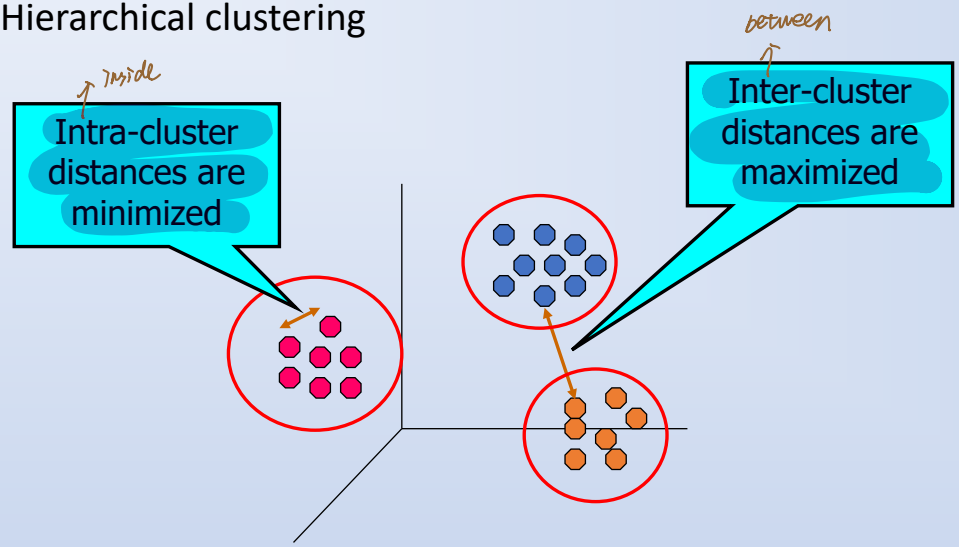


Figure from Tan, Steinbach and Kumar 2004

Quality: What Is a Good Clustering?

- A good clustering method will produce high quality clusters
  - Objects within same cluster are close together
  - Objects in different clusters are far apart
- Clustering is a major task in data analysis and visualisation, useful not just for outlier detection, e.g.
  - Market segmentation
  - Search engine result presentation
  - Personality types

Cambridge Analytica

Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach

Whistleblower describes how firm linked to former Trump adviser Steve Bannon compiled user data to target American voters

- Infer people's personality from their Facebook likes (correlating likes with an online personality test) – psychographics
  - *"Kosinski proved that on the basis of an average of 68 Facebook "likes" by a user, it was possible to predict their skin color (with 95 percent accuracy), their sexual orientation (88 percent accuracy), and their affiliation to the Democratic or Republican party (85 percent). But it didn't stop there. Intelligence, religious affiliation, as well as alcohol, cigarette and drug use, could all be determined."*

Cambridge Analytica

- *“before long, he was able to evaluate a person better than the average work colleague, merely on the basis of ten Facebook "likes”. Seventy "likes" were enough to outdo what a person's friends knew, 150 what their parents knew, and 300 "likes" what their partner knew. More "likes" could even surpass what a person thought they knew about themselves.”*

- Might then target clusters (segments) of people with certain personality traits, with appropriate political messages
  - *"For a highly neurotic and conscientious audience the threat of a burglary—and the insurance policy of a gun.”*

- Quotes taken from following article

https://motherboard.vice.com/en_us/article/how-our-likes-helped-trump-win

Pre-requisite: Distance functions

how similar two data objects are

- Clustering methods are typically distance based. Represent each object / instance as a vector (a row in the data) and then compute Euclidean distance between pairs of vectors
- Commonly normalise (scale) each attribute into range [0,1] via a pre-processing step before computing distances

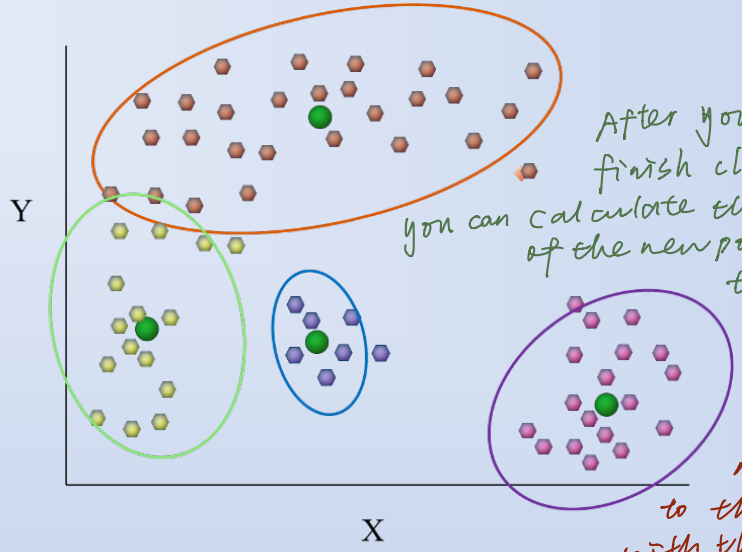
$$\text{Given } X = (x_1, x_2, x_3, \ldots, x_n) \text{ and } Y = (y_1, y_2, y_3, \ldots, y_n)$$

$$D(X, Y) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 + \ldots (x_n - y_n)^2}$$

clustering doesn't give you classification

How to obtain the clusters?

- Need to assign each object to exactly one cluster
- Each cluster can be summarised by its centroid (the average of all its objects)



After you have finish clustering, you can calculate the distance of the new point to the existing centroids.

assign the new point to the cluster with the nearest centroid

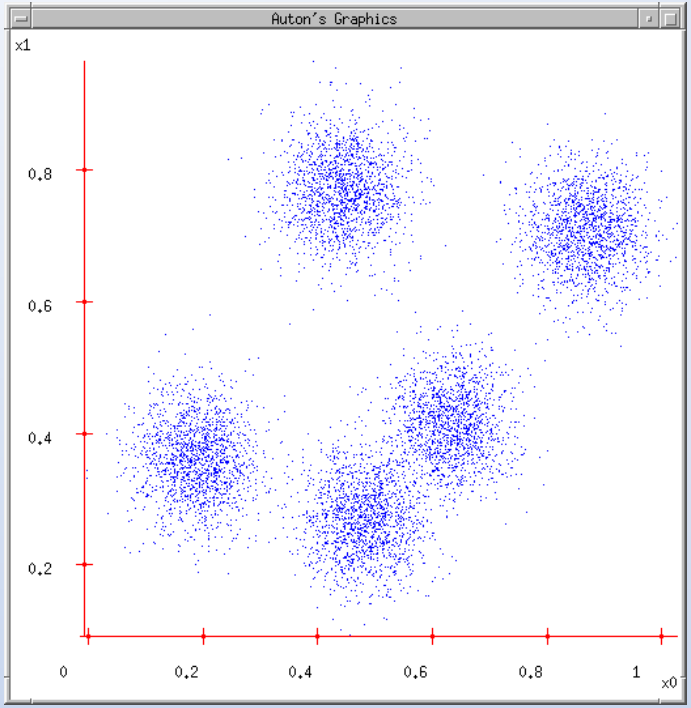
K-Means Clustering

K-Means: The best-known clustering method

Given parameter *k*, the *k-means* algorithm is implemented in four steps:

1. Randomly select *k* seed points as the initial cluster centroids
2. Assign each object to the cluster with the nearest centroid
3. Compute the centroids of the clusters of the current partitioning (the centroid is the center, i.e., ***mean point***, of the cluster)
4. Go back to Step 2, stop when the assignment does not change

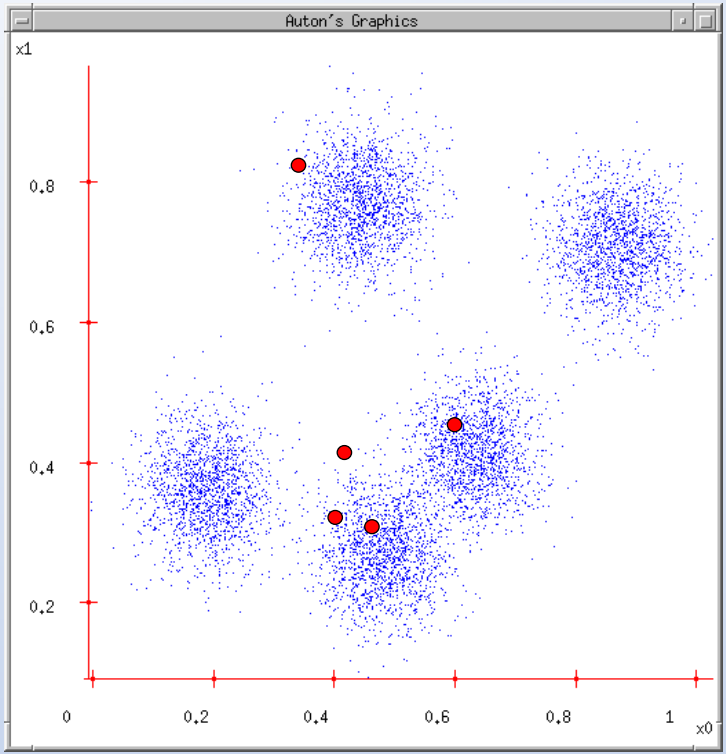
K-means - Example



1. Ask user how many clusters they would like *(e.g. K=5)*

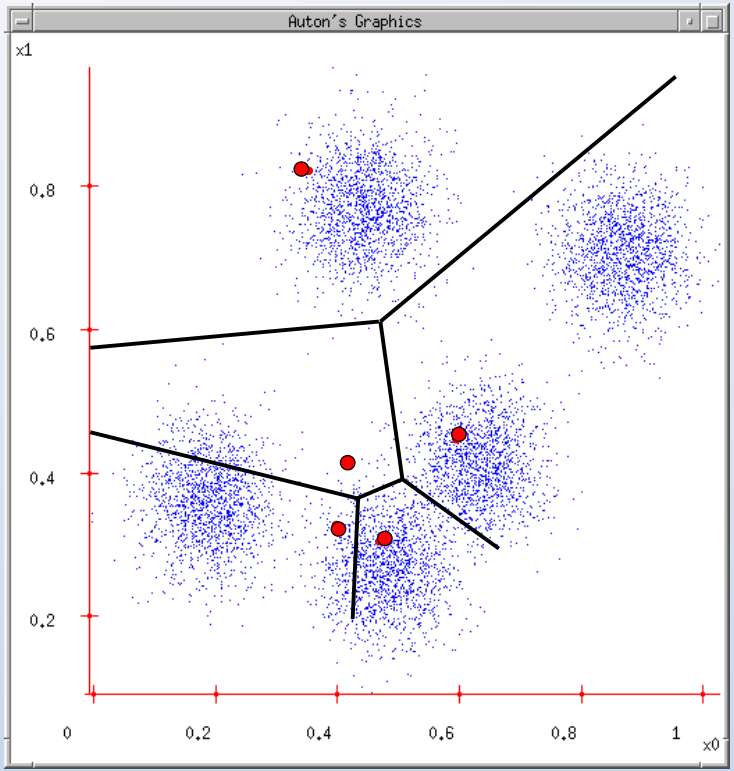
(Example from Andrew Moore
http://www.autonlab.org/tutorials/kmeans11.pdf)

K-means - Example



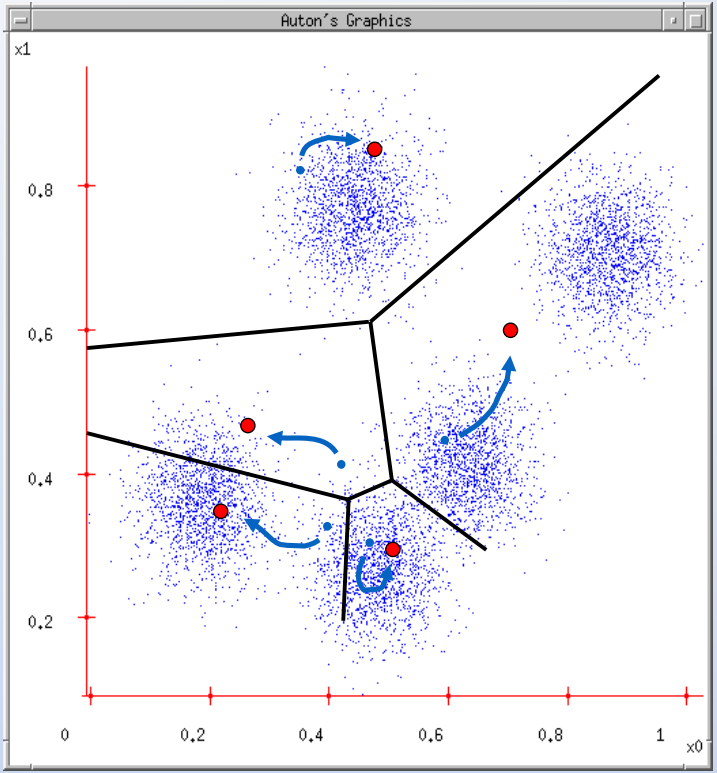
1. Ask user how many clusters they would like *(e.g. K=5)*
2. Randomly pick K cluster centroids

K-means - Example



1. Ask user how many clusters they would like *(e.g. K=5)*
2. Randomly pick K cluster centroids
3. For each data point find out which centroid it is closest to. (Thus each centroid "owns" a set of datapoints)

K-means - Example



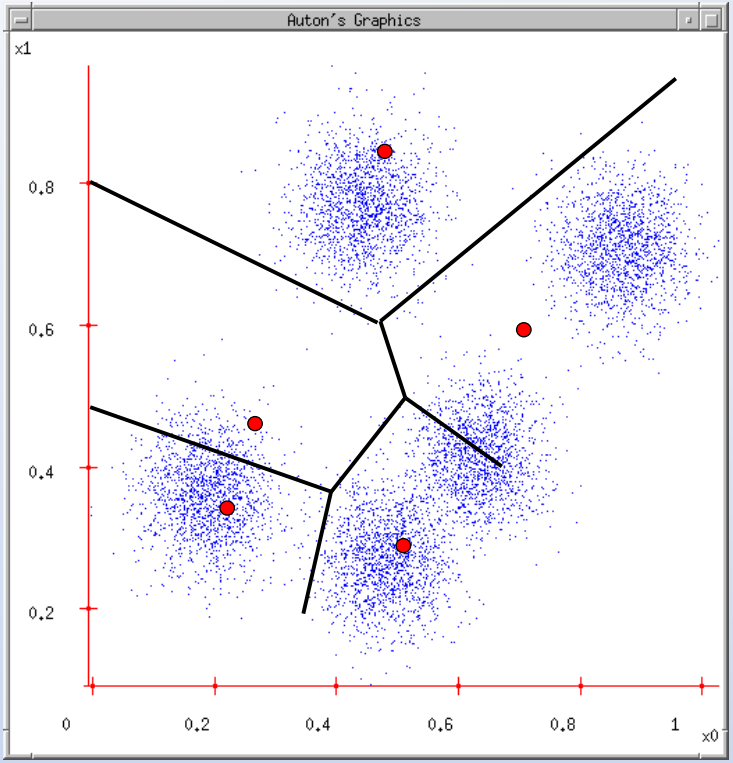
1. Ask user how many clusters they would like *(e.g. K=5)*
2. Randomly pick K cluster centroids
3. For each data point find out which centroid it is closest to. (Thus each centroid "owns" a cluster of datapoints)
4. Each cluster finds the new centroid of the points in it

not a real point
is the point you calculate
(you randomly pick the 1st centroid).

always the average of all the points in the region

step 4 is based on the region, calculate new average

K-means - Example



1. Ask user how many clusters they would like *(e.g. K=5)*
2. Randomly pick K cluster centroids
3. For each data point find out which centroid it is closest to. (Thus each centroid "owns" a cluster of datapoints)
4. Each cluster finds the new centroid of the points in it
5. New centroids => new boundaries
6. Repeat 3-5 until no further changes

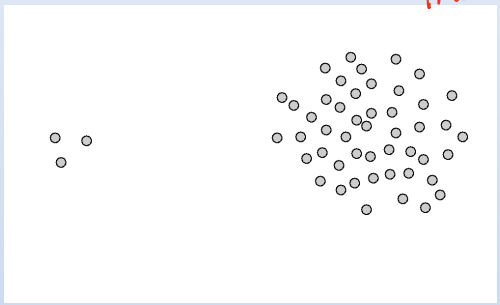
Understanding the Algorithm

For which dataset does k-means require a fewer number of iterations?

k= 2

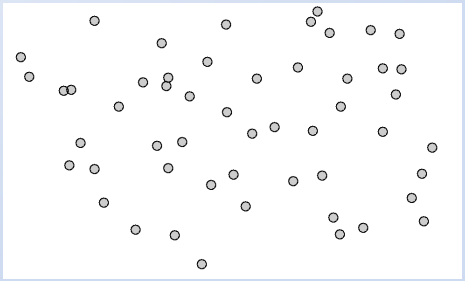
will be solved quickly → data point in general are much closed to each other → less likely to get change from boundary

Dataset 1



including new datapoint will change the boundaries → points will more likely to belong to different clusters as time goes on

Dataset 2



K-means: Further details

- Typically choose the initial seed points randomly
  - Different runs of the algorithm will produce different results
- Closeness measured by Euclidean distance
  - Can also use other distance functions (e.g. correlation)
- Algorithm can be shown to converge (to a local optimum)
  - Typically does not require many iterations

K-Means Interactive demos

- http://syskall.com/kmeans.js/
- http://tech.nitoyon.com/en/blog/2013/11/07/k-means/
- https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

An application:
Cluster-based outlier detection

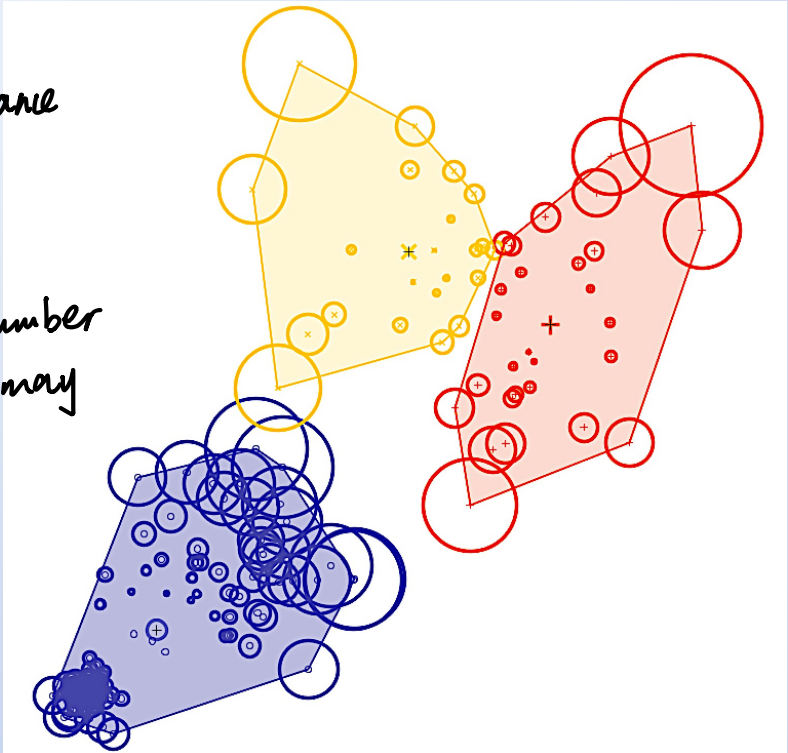
- An outlier is expected to be far away from any groups of normal objects

- Each object is associated with exactly one cluster and its outlier score is equal to the distance from its cluster centre
  - Score = size of circle in following example

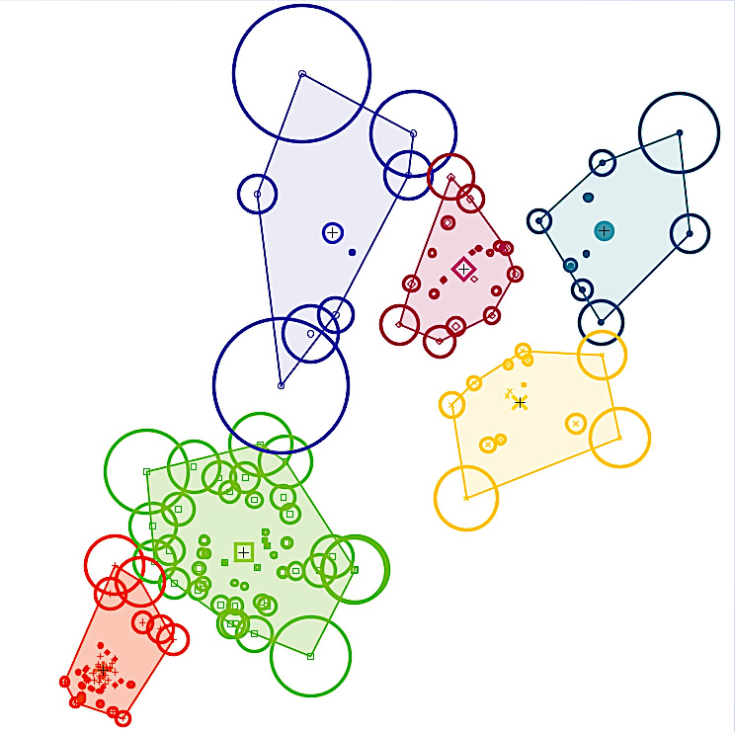
Clustering based outlier detection
3 clusters

size of circle is the distance to the cluster center

we could do the ontlier detection just on the number but the visualisation may help us see better



Clustering based outlier detection
6 clusters



Clustering based outlier detection
9 clusters

Visual Assessment of Clustering Tendency (VAT)

Visual Assessment of Clustering Tendency (VAT)

- How many clusters are in the data? How big are they? What is the likely membership of objects in each cluster?
  - The *k* parameter for k-means
- One solution: *visually determine the clustering structure by inspecting a heat map*
  - Represent datasets in an n*n image format
  - Applicable for many different types of object data

What is a (dis)similarity matrix?

Object id	Feature1	Feature2	Feature3
1	5	10	15
2	10	5	10
3	20	20	20

Compute all pairwise distances between objects. This gives a dissimilarity matrix.

number ↑

difference ↑

distance is calculated by Euclidean distance

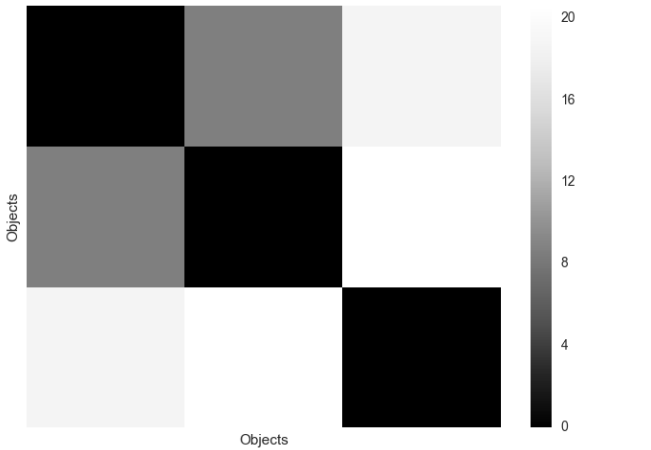
$$\sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Object	1	2	3
1	0	8.7	18.7
2	8.7	0	20.6
3	18.7	20.6	0

Visualising a dissimilarity matrix

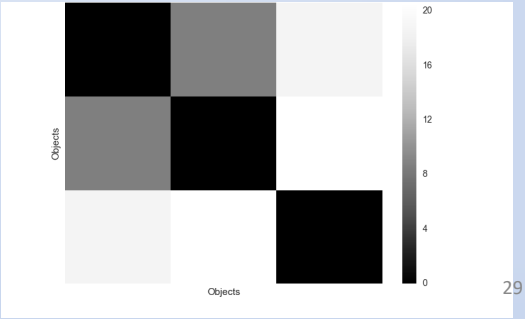
We can visualise a dissimilarity matrix as a *heat map*, where the colour of each cell indicates that cell's value

Object	1	2	3
1	0	8.7	18.7
2	8.7	0	20.6
3	18.7	20.6	0

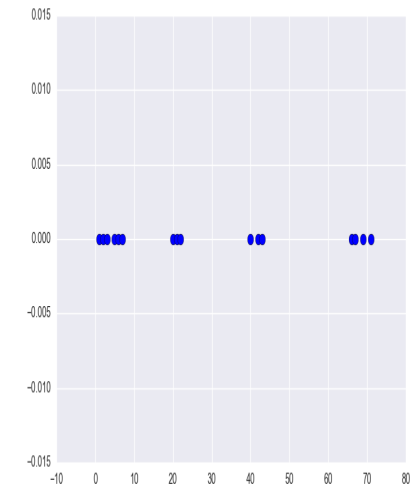


Properties of a dissimilarity matrix D

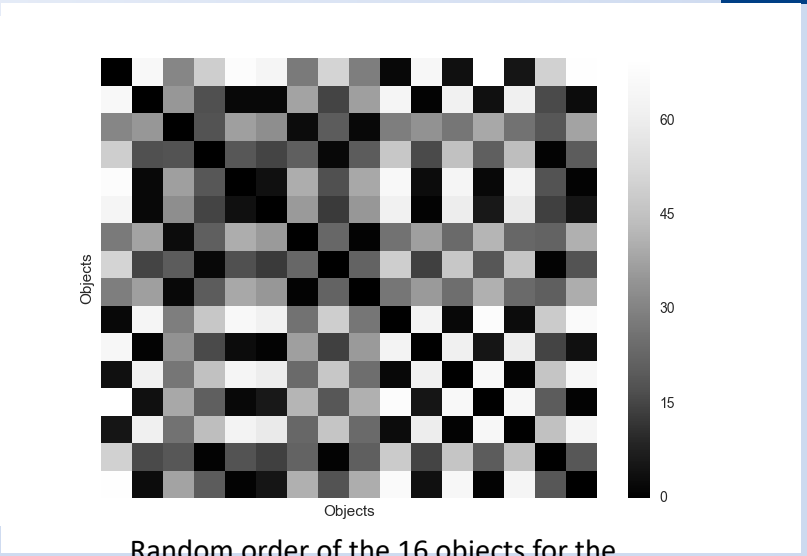
- The diagonal of D is all zeros
- D is symmetric about its leading diagonal
  - D(i,j)=D(j,i) for all i and j
  - Objects follow the same order along rows and columns
- Visualising the (raw) dissimilarity matrix may not reveal enough useful information
  - Further processing often needed



Reordering a Dissimilarity matrix

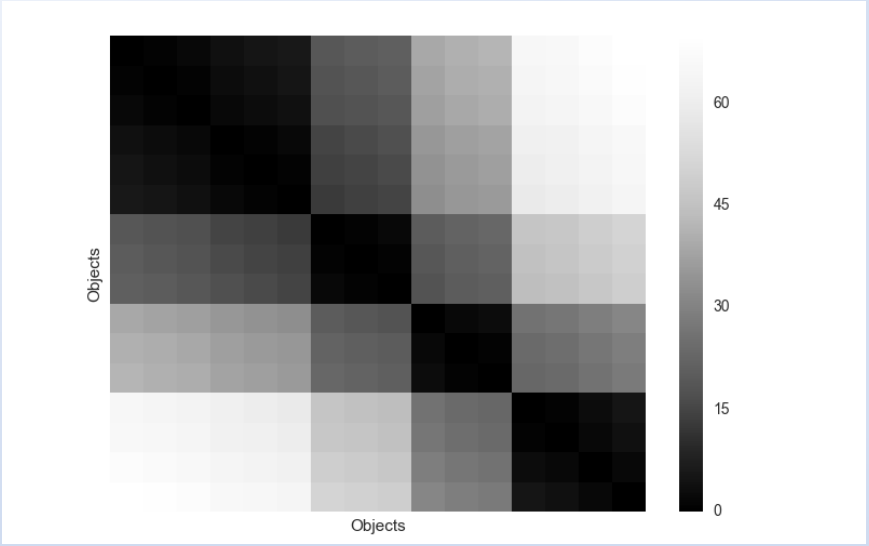


Example dataset with 16 objects



Random order of the 16 objects for the dissimilarity matrix

Reordering the matrix reveals the clusters



"count" the black squares in the VAT - it gives you a good guess about the closeted numbers

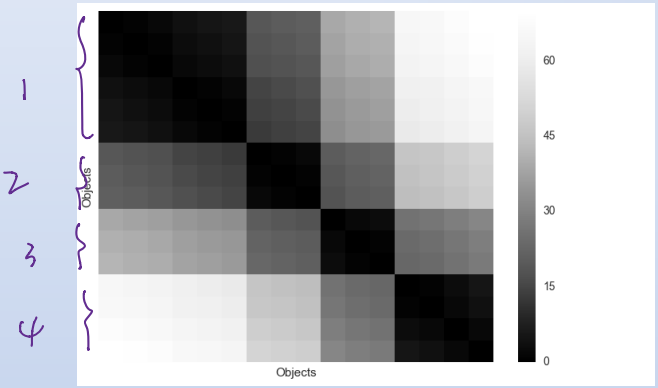
A better ordering of the 16 objects. Nearby objects in the ordering are similar to each other, producing large dark blocks. We can see four clusters along the diagonal

Good VAT images

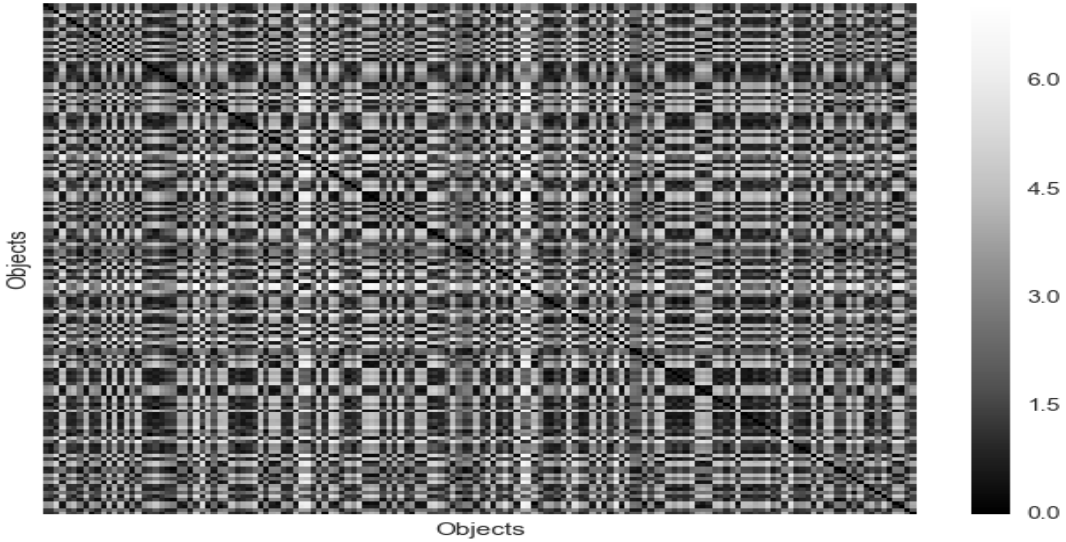
- A good VAT image suggests both the number of clusters and approximate members of each cluster
- A diagonal dark block appears in the VAT image only when a tight group exists in the data (low within-cluster dissimilarities)

dark → low dissimilarity

4 clusters

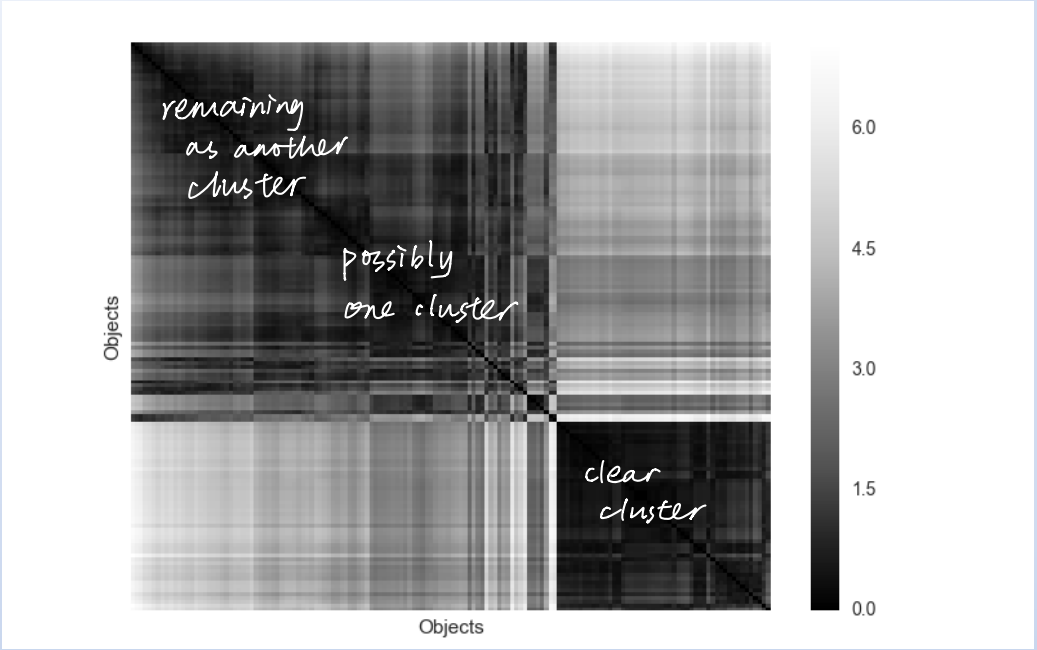


Another example: dissimilarity matrix for Iris dataset

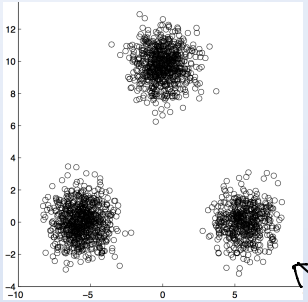


Random order for 150 objects: Where are the clusters???

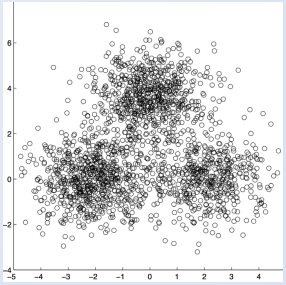
Dissimilarity matrix for Iris data (reordered)



Exercise - Match the datasets with the VAT images

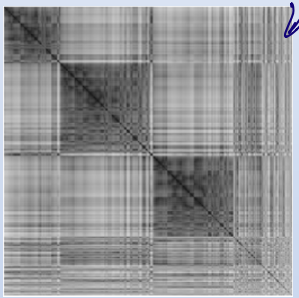


1

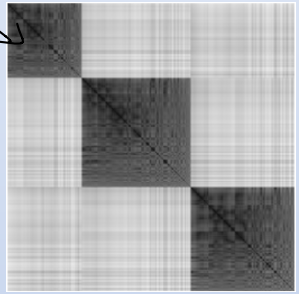


2

loose clusters



a

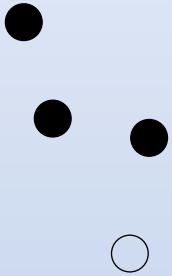


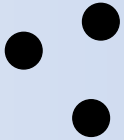
b

From: An algorithm for clustering Tendency Assessment, Hu, Y., Hathaway, R. WSEAS Transactions on Mathematics, 2008.

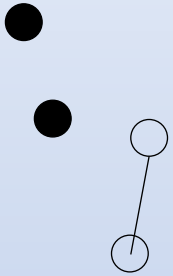
VAT ordering example

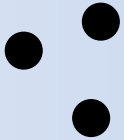
VAT ordering example



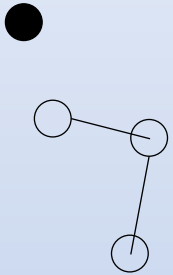


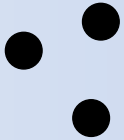
VAT ordering example



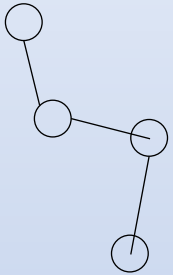


VAT ordering example

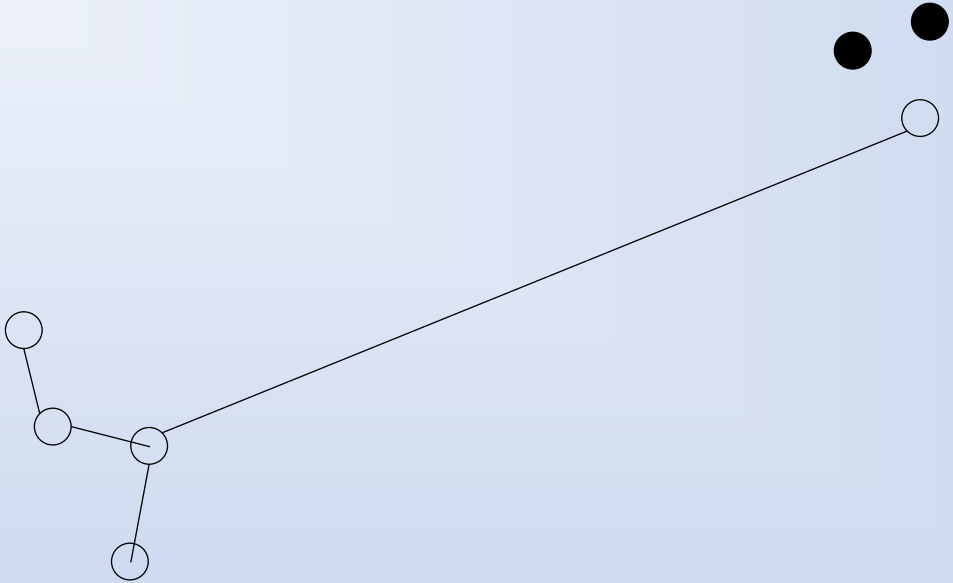




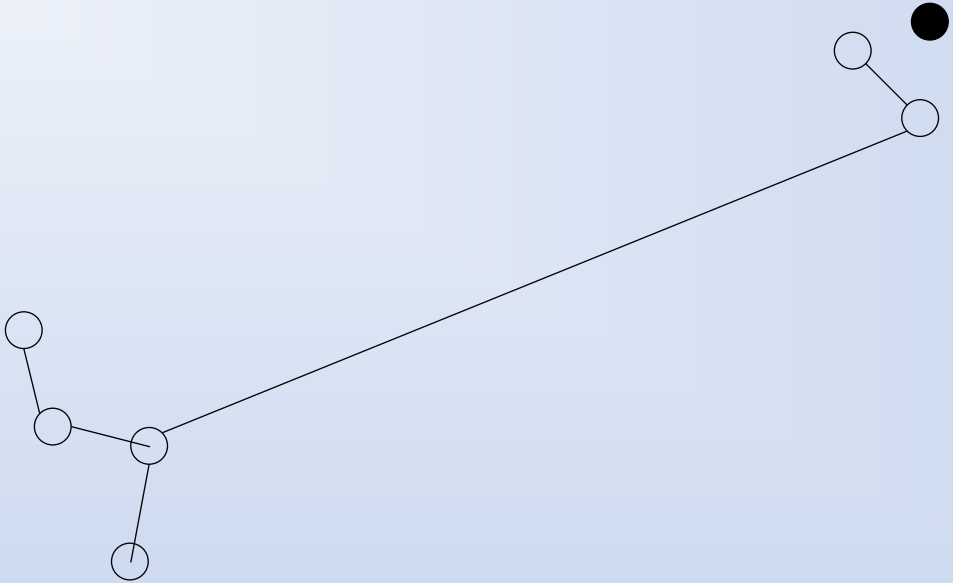
VAT ordering example



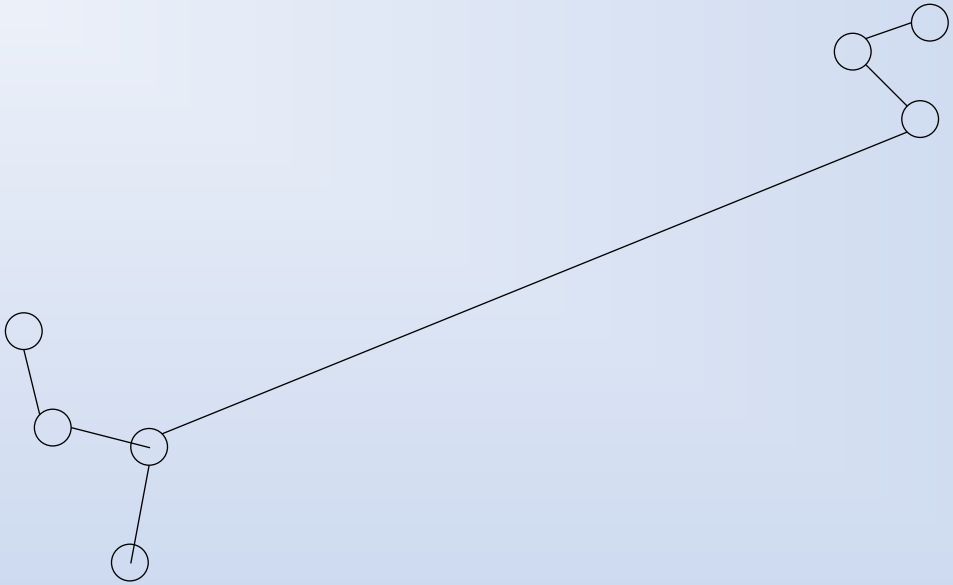
VAT ordering example



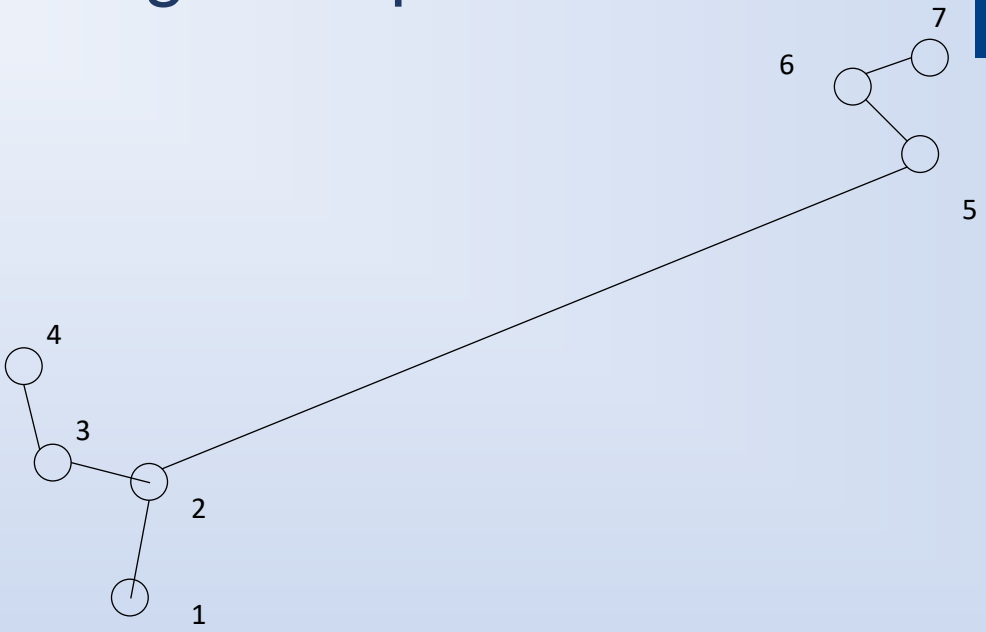
VAT ordering example



VAT ordering example



VAT ordering example



The VAT algorithm: Visual assessment for clustering tendency

(Bezdek and Hathaway 2002)

Given an N*N dissimilarity matrix **D**

Let K={1,…N}, I=J={} ### {} is a set (collection of objects)

Pick the two least similar objects o_a and o_b from **D**

P(1)=a; I={a}; J=K-{a}

For r = 2, …., N

Select (i,j): pair of most similar objects o_i and o_j from **D**

Such that i ∈ I, j ∈ J ### '∈' means 'member of'

P(r) = j; I = I ∪ {j}; J=J - {j}; ### '∪' means 'union' i.e. add two collections together

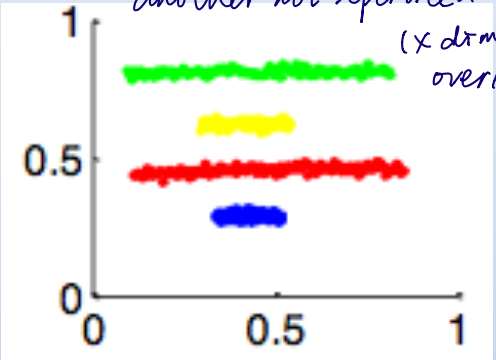
- Obtain reordered dissimilarity matrix D* from permutation (ordering) P
  - E.g. P(1)=2, P(2)=5, P(3)=7
  - The first object in the ordering is 2, the second is 5, the third is 7 ….
- VAT algorithm python code – you will practice in workshops

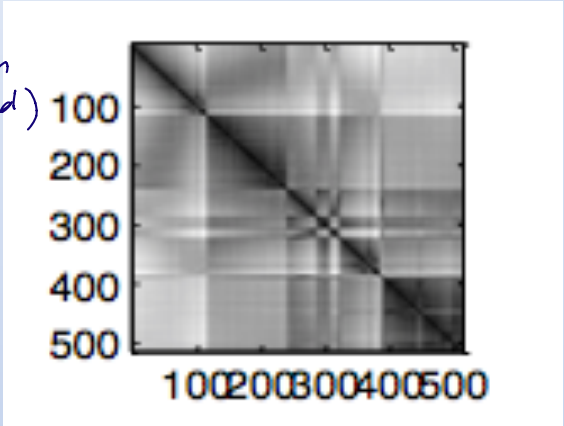
The VAT algorithm - problems

VAT algorithm is not effective in every situation

- For complex shaped datasets (either significant overlap or irregular geometries between different clusters), the quality of the VAT image may significantly degrade

one dimension clearly 4 clusters
another not seperated
(x dimension overlapped)

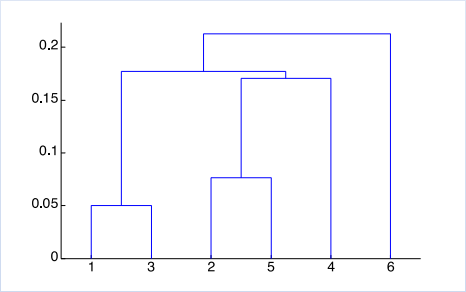


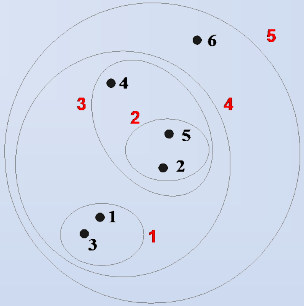


Hierarchical Clustering

Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges (y-axis is distance)





Strengths of Hierarchical Clustering

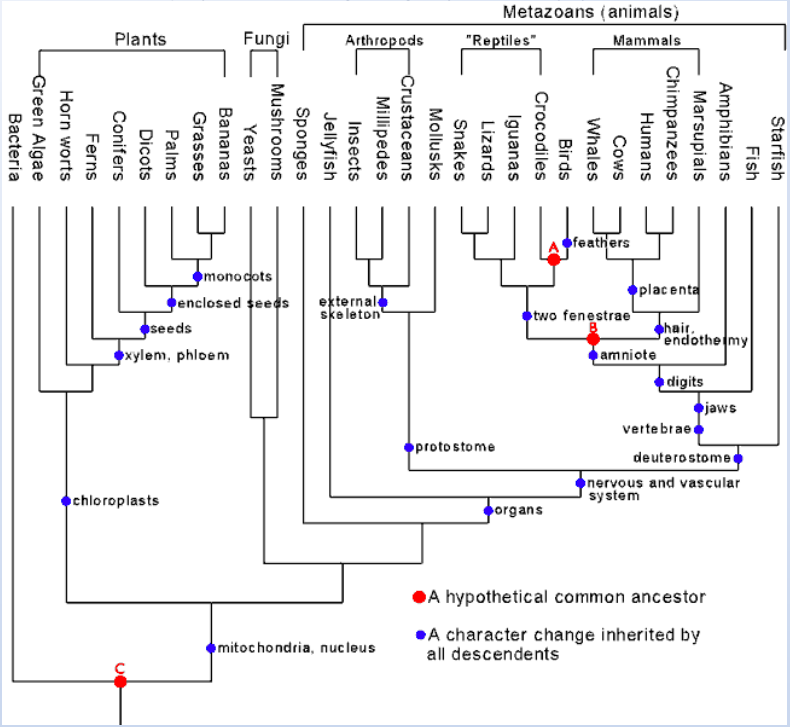
- No assumption of any particular number of clusters
  - Any desired number of clusters can be obtained by 'cutting' the dendrogram at the appropriate level

分类学.

- They may correspond to meaningful taxonomies
  - Example in biological sciences (e.g., animal kingdom, phylogeny reconstruction, …)

Tree of Life

(http://www.talkorigins.org/faqs/comdesc/phylo.html)



Hierarchical Clustering

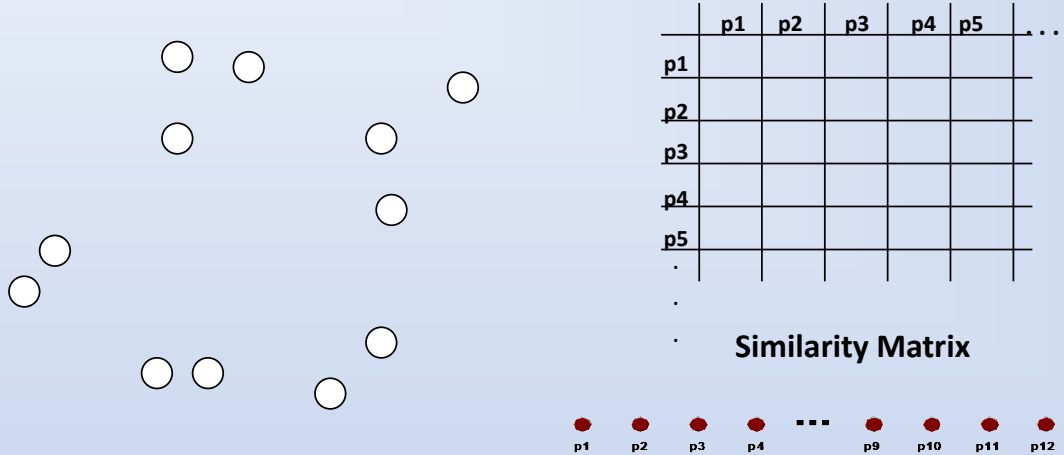
- Two main types of hierarchical clustering
  - Agglomerative: → collect and form into a mass or group
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a (dis)similarity or distance matrix
  - Merge or split one cluster at a time

Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward
  - Compute the similarity matrix
  - Let each data point be a cluster
  - **Repeat**
    - Merge the two closest clusters
    - Update the proximity matrix
    - **Until** only a single cluster remains
- Key operation is the computation of the similarity of two clusters
  - Different approaches to defining the distance between clusters distinguish the different algorithms

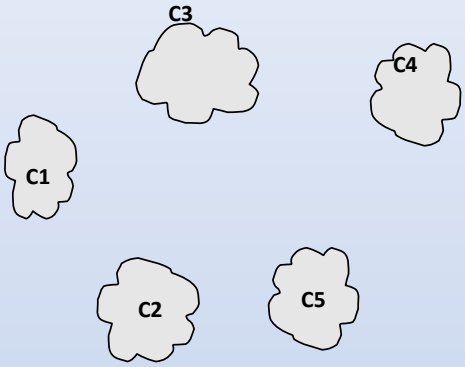
Agglomerative Clustering Example (1)

Start with clusters of individual points and a similarity matrix



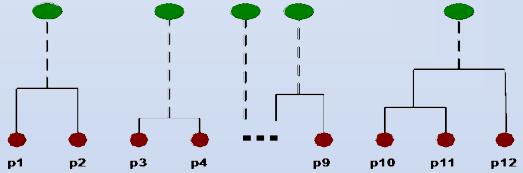
Agglomerative Clustering Example (2)

After some merging, we have clusters



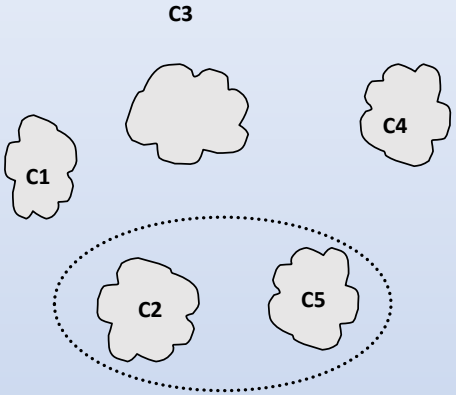
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Similarity Matrix



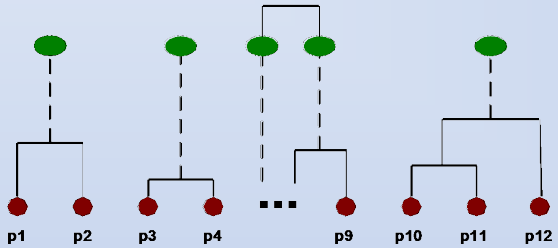
Agglomerative Clustering Example (3)

We merge the two closest clusters (C2 and C5) and update the similarity matrix



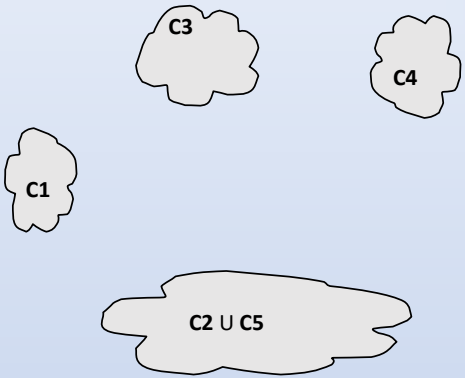
	C1	C2	C3	C4	C5
C1					
C2					
C3					
C4					
C5					

Similarity Matrix



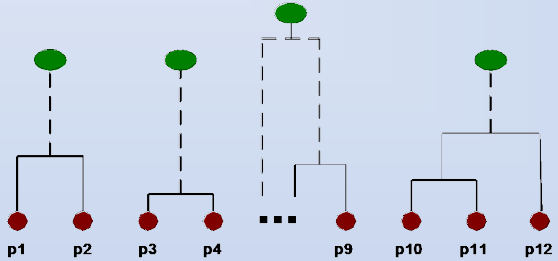
Agglomerative Clustering Example (4)

How do we update the similarity matrix?

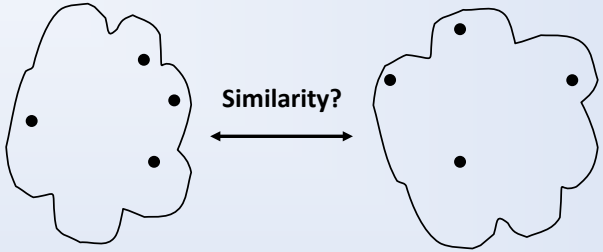


	C1	C2 U C5	C3	C4
C1		?		
C2 U C5	?	?	?	?
C3		?		
C4		?		

Similarity Matrix



Agglomerative Clustering Example (4)

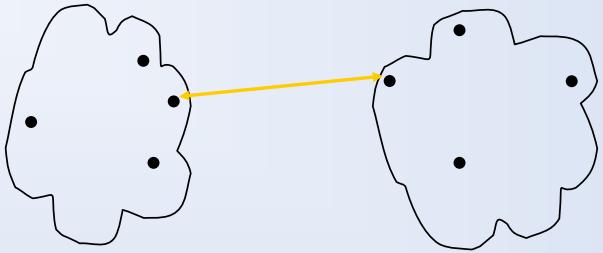


	p1	p2	p3	p4	p5	...
p1						
p2						
p3						
p4						
p5						

Similarity Matrix

- We define the similarity to be the minimum distance between the clusters, also known as single linkage
- Other choices also possible, e.g. maximum distance, also known as complete linkage

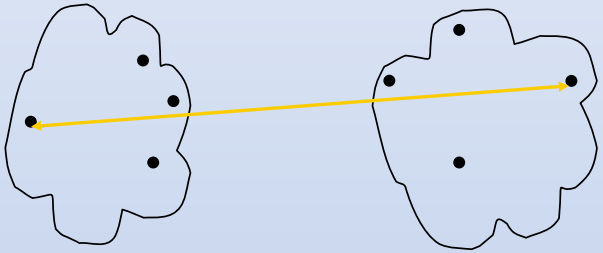
Agglomerative Clustering Example (5)



- MIN (Single Linkage)

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

Similarity Matrix

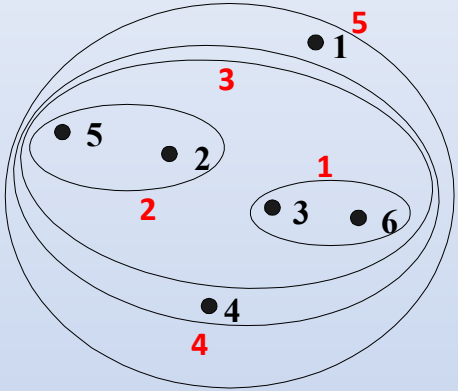


- MAX (Complete Linkage)

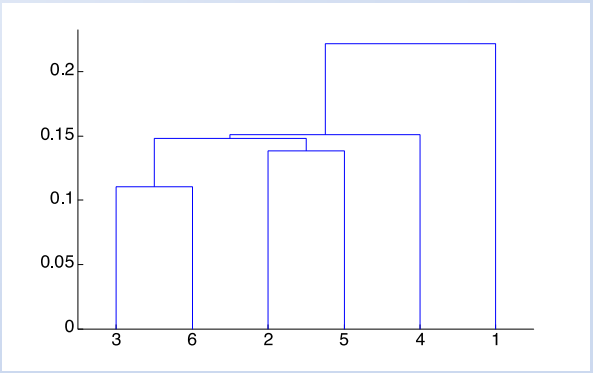
Agglomerative Clustering Example (6)

MIN (Single Linkage: the similarity of two clusters is based on the two most similar (closest) points in the different clusters

- Determined by one pair of points, i.e., by one link in the proximity graph.



Nested Clusters

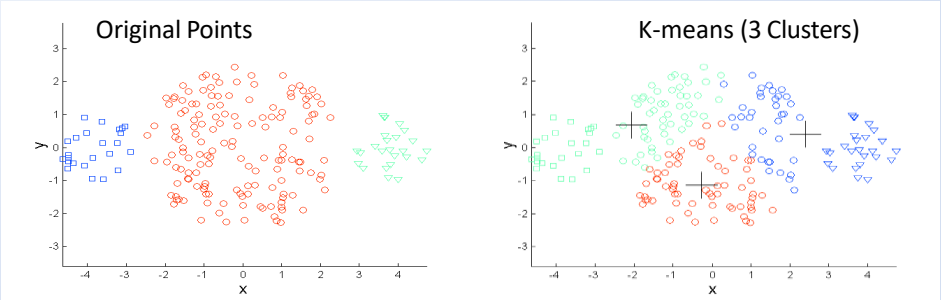


Dendrogram

Clustering - Reflections

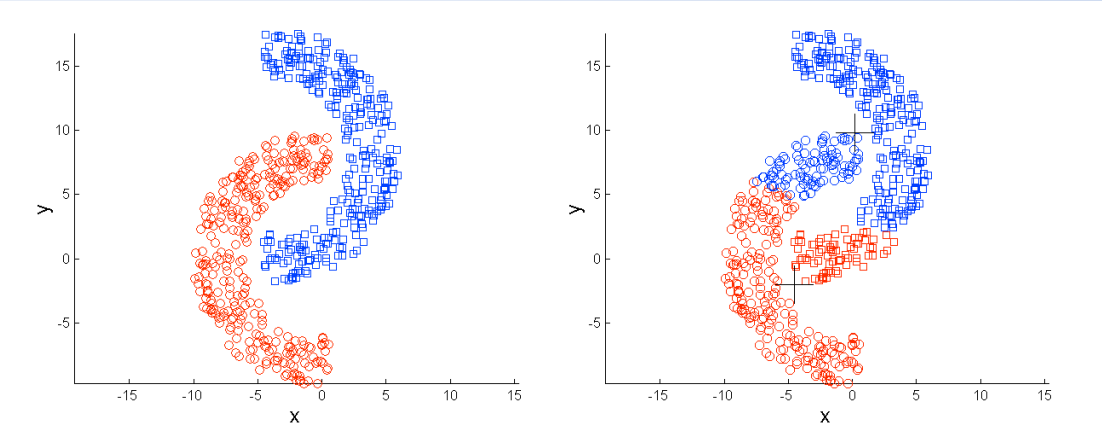
Clustering Reflections: K-means (1)

Limitations of K-means – result quality varies based on size and density



Clustering Reflections: K-means (2)

Limitations of K-means – not strong for non-globular shapes

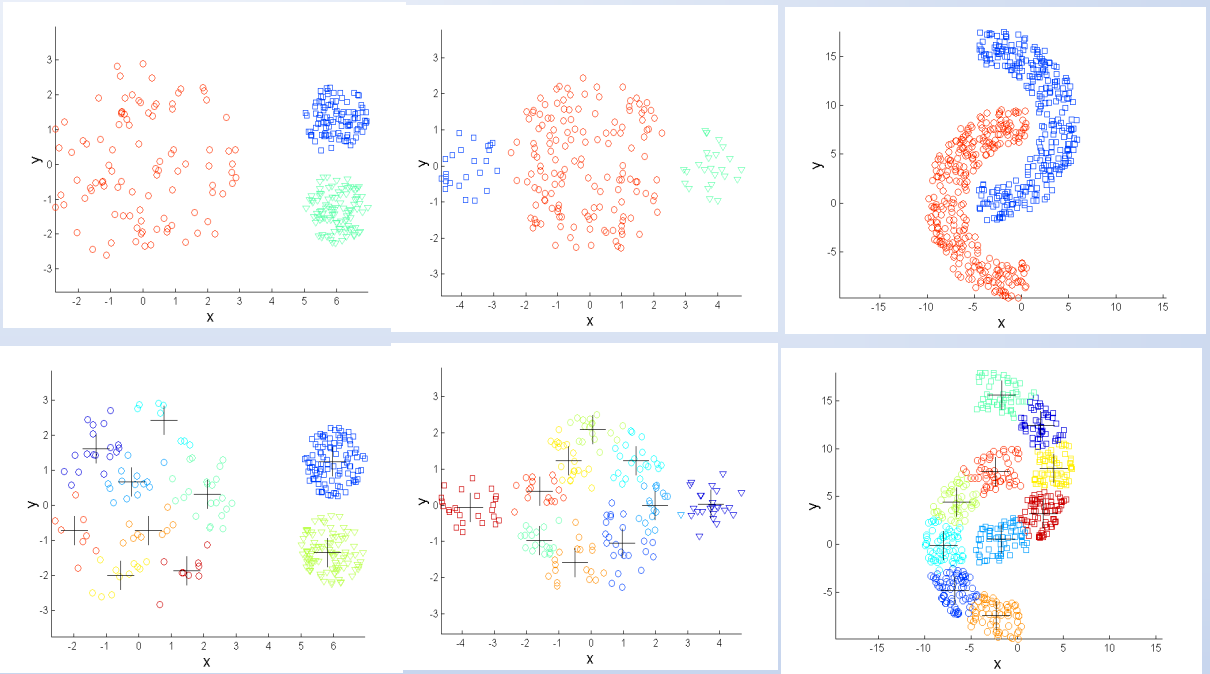


Original Points

K-means (2 Clusters)

Clustering Reflections: K-means (3)

An idea: Use many clusters, merge together

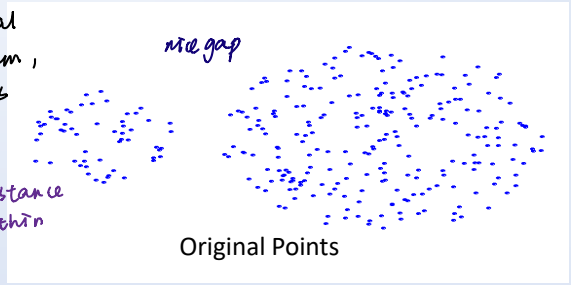


Clustering Reflections: hierarchical (1)

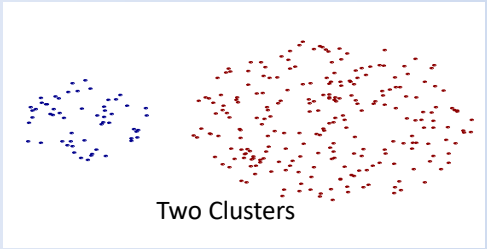
Strength of MIN: Single linkage: Can handle non-similar shapes

clear differential between two system, won't merge this two together

the minimum distance will always within the cluster

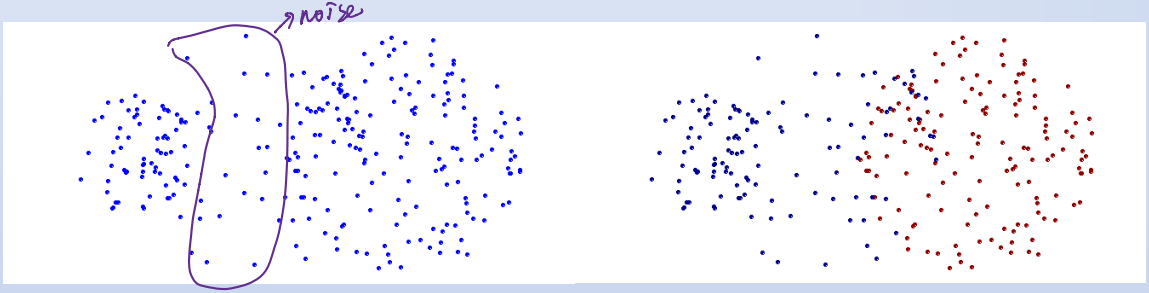


Original Points



Two Clusters

Limitations of MIN: Single linkage: Sensitive to noise and outliers

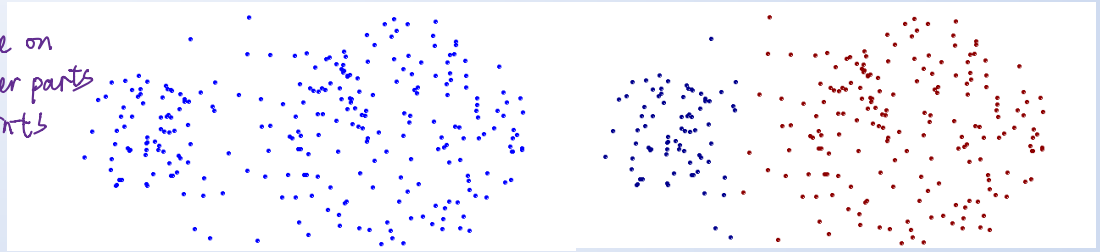


Original Points

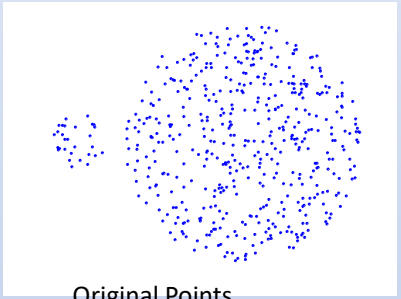
Two Clusters

Clustering Reflections: hierarchical (2)

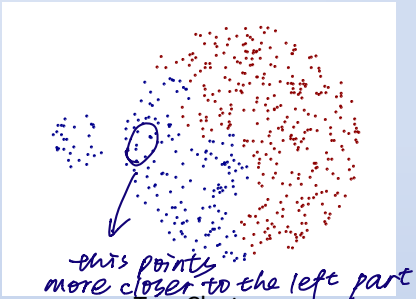
Strength of MAX: Complete linkage: Less susceptible to noise and outliers



Limitations of Max: Complete linkage: Tends to break large clusters



Original Points



Two Clusters

Clustering Reflections: hierarchical (3)

- Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimised, i.e. no 'relative' quality measure

good cluster = points within the clusters are close together and clusters are far away from each other

for any clustering method it is hard to understand if the result is good or not. it is very difficult to measure the quality numerically

for k-mean clustering, we could look at distance between centroids and say if the distance is large, the clusters are well seperated. However, for hierarchial methods, there is very little we can calculate to tell us if the quality is good.

Acknowledgements

- Materials are partially adopted from …
  - Previous COMP2008 slides including material produced by James Bailey, Pauline Lin, Chris Ewin, Uwe Aickelin and others
  - Data Mining Concepts and Techniques", Han et al, 2nd edition 2006.
  - "Introduction to Data Mining", Tan et al 2005.

① how do we choose between k mean and hierarchical ?

→ If you know or at least the data has a hierarchical structure eg classes, sub-classes. (eg. in biologyical ...).

→ Or after you have used k-means, you may realised that the data is hierarchical and then use other method.

→ One other reason to use hierarchical method is the flexible compression ie. you can reduce the data to 10, 8, 6, 4 clusters but for k-means, the k is fixed.