COMP20008 Elements of Data Processing

Semester 2 2018

Lecture 7: Clustering and Clustering Visualisation

Announcements

- Project Phase 1 was released on Monday 13th August
- Consultation sessions about Project Phase 1
  - Fri 17/08/2018 Room 09.02 Doug McDonell 9.30am-10:30am
  - Thu 23/08/2018 Room 07.02 Doug McDonell 11am-12pm
  - Wed 29/08/2018 Room 07.02 Doug McDonell 10:30am-11:30am

Outline

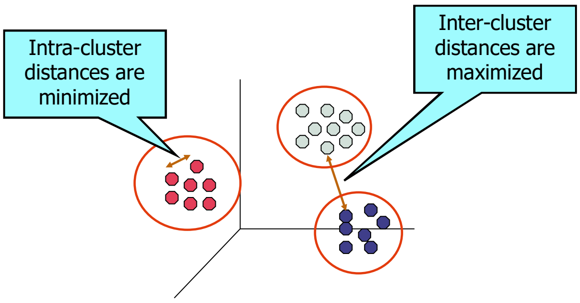
- Complete section of basic visualisations
- Clustering algorithms
  - K-means
  - Visualisation of clustering tendency (VAT)
- Next class
  - Hierarchical clustering (next class)

Data in higher dimensions

- For datasets with more than 4 dimensions (features)
  - Difficult to visualise
- How can we determine what the significant groups/segments/communities are?
  - If we have this information
    - Can understand the data better
    - Apply separate interventions to each group (e.g. marketing campaign)

What is Cluster Analysis?

- Figure below from Tan, Steinbach and Kumar 2004
- We will be looking at two classic clustering algorithms
  - K-means
  - Hierarchical clustering



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.
  - Market segmentation
  - Image analysis
  - Search engine result presentation
  - Personality type
  - ….

Pre-requisite: Distance functions

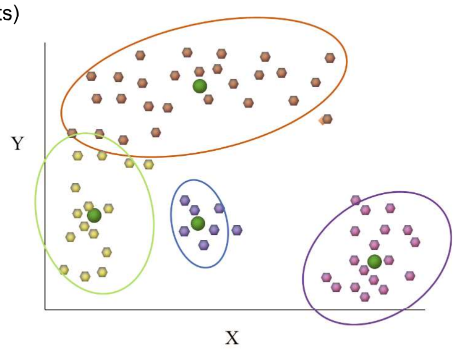
- Clustering methods are typically distance based. Represent each object/instance as a vector (a row in the data, with values for each different feature) and then can compute Euclidean distance between pairs of vectors.
- Commonly normalise (scale) each attribute into range [0,1] via a pre-processing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

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

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

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)



THE UNIVERSITY OF MELBOURNE

K-Means: The best known clustering method

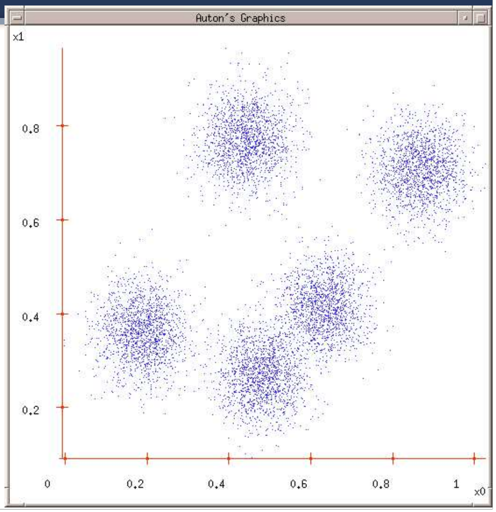
- Given parameter *k*, the *k-means* algorithm is implemented in four steps:
  1. Select *k* seed points as the initial cluster centres
  2. Assign each object to the cluster with the nearest seed point
  3. Compute new seed points as the centroids of the clusters of the current partitioning (the centroid is the center, i.e., ***mean point***, of the cluster)
  4. Go back to Step 2, stop when the assignment does not change

THE UNIVERSITY OF MELBOURNE

K-means

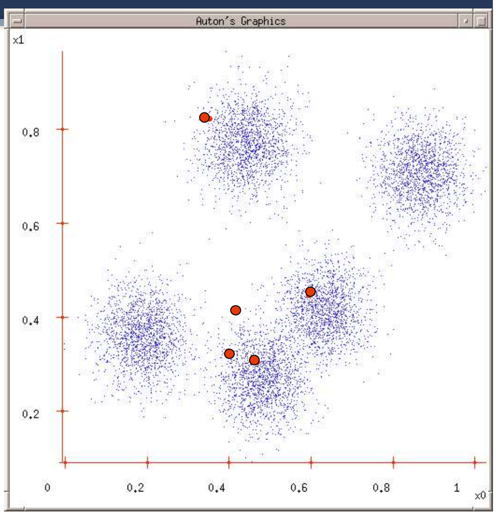
1. Ask user how many clusters they' d like. *(e.g. K=5)*

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



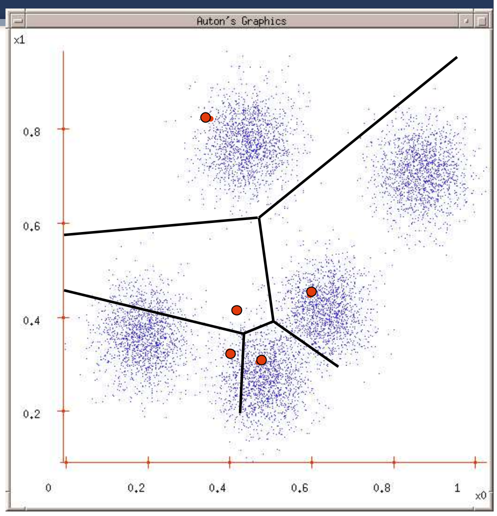
THE UNIVERSITY OF MELBOURNE

1. Ask user how many clusters they' d like. *(e.g. K=5)*
2. Randomly guess K cluster Center locations

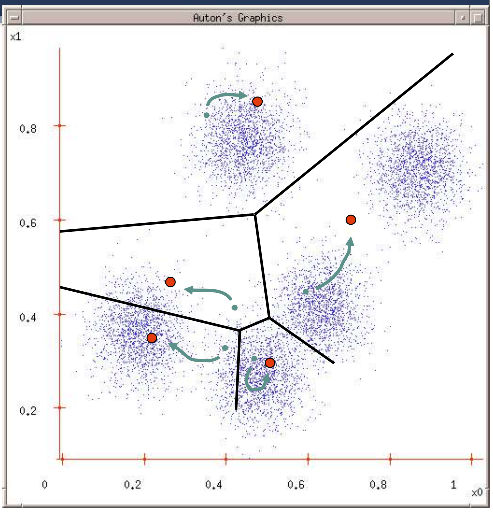


THE UNIVERSITY OF MELBOURNE

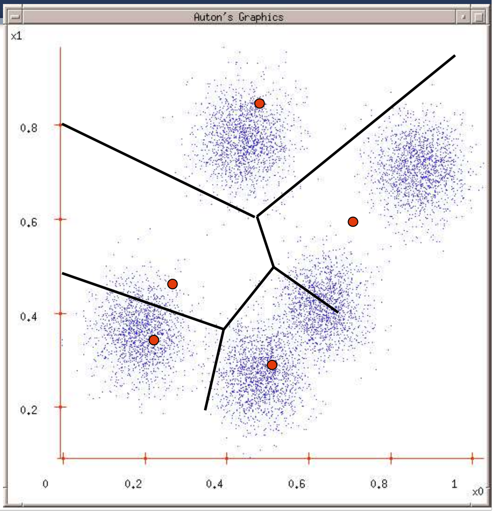
1. Ask user how many clusters they' d like. *(e.g. K=5)*
2. Randomly guess K cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



1. Ask user how many clusters they' d like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it' s closest to.
4. Each Center finds the centroid of the points it owns

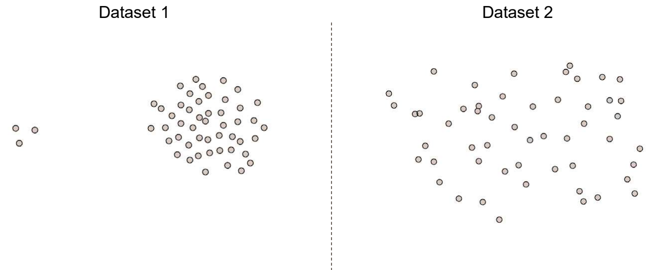


1. Ask user how many clusters they' d like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it' s closest to.
4. Each Center finds the centroid of the points it owns
5. New Centers => new boundaries
6. Repeat until no change



Understanding the Algorithm

For which dataset does k-means require less number of iterations?
k= 2



K-means: Further detail

- Typically choose the initial seed points randomly
  - Different runs of the algorithms will produce different results
- Closeness measured by Euclidean distance (Can also use other distance functions)
- Algorithm can be shown to converge (to a local optimum), typically doesn't require many iterations

THE UNIVERSITY OF MELBOURNE

K-Means Interactive demo

- http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html
- http://syskall.com/kmeans.js/

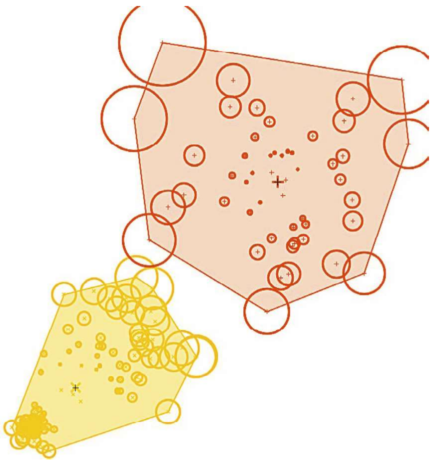
THE UNIVERSITY OF MELBOURNE

An application: Clustering 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.

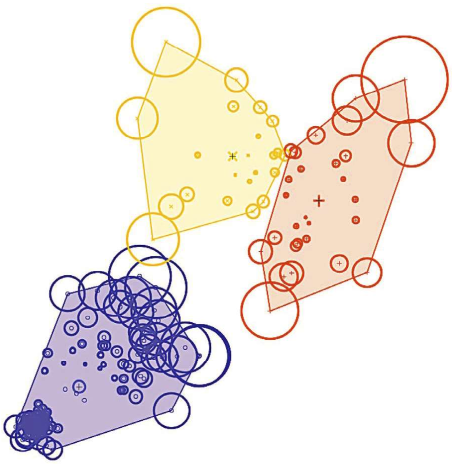
THE UNIVERSITY OF MELBOURNE

Clustering based outlier detection. 2 clusters



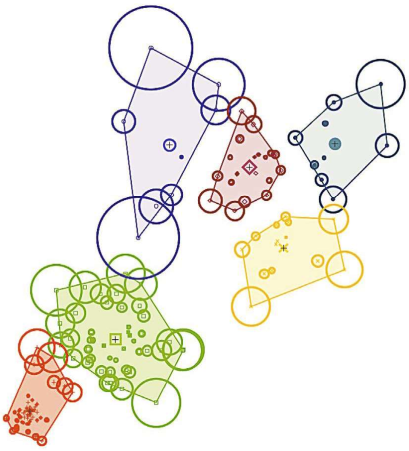
THE UNIVERSITY OF MELBOURNE

3 clusters



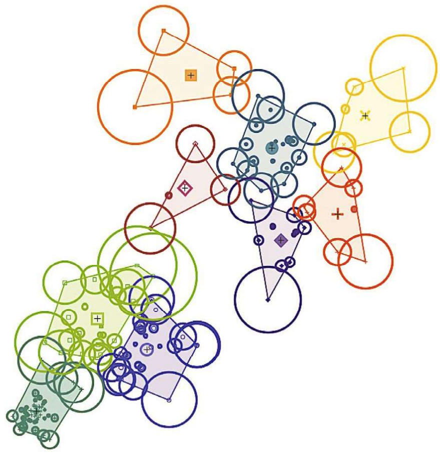
THE UNIVERSITY OF MELBOURNE

6 clusters



THE UNIVERSITY OF MELBOURNE

9 clusters



THE UNIVERSITY OF MELBOURNE

Visual Assessment for Clustering Tendency

- 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

THE UNIVERSITY OF MELBOURNE

Background: What is a dissimilarity matrix?

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

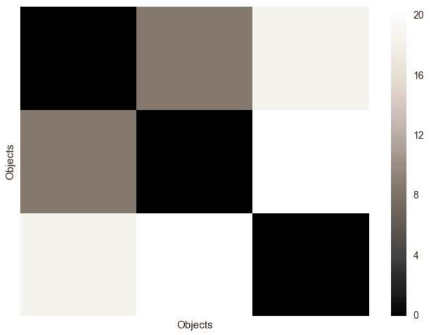
Compute all pairwise distances between objects. This gives a dissimilarity matrix.

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

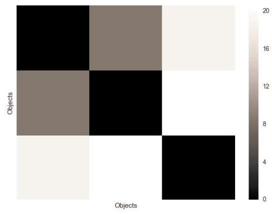
Object	**1**	**2**	**3**
1	0	8.7	18.7
2	8.7	0	20.6
3	18.7	20.6	0



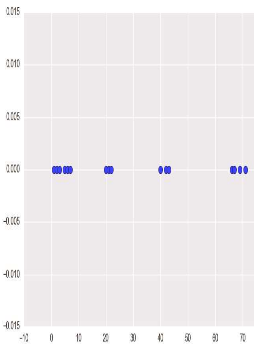
Properties of a dissimilarity matrix D

- 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
- In general, visualising the (raw) dissimilarity matrix may not reveal enough useful information
  - Further processing is needed

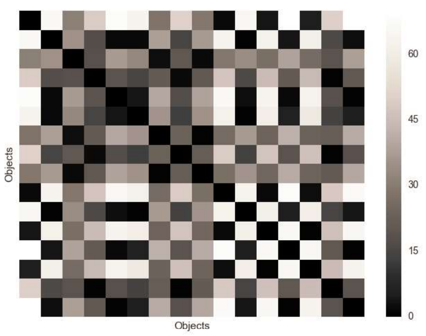
Object	**1**	**2**	**3**
1	0	8.7	18.7
2	8.7	0	20.6
3	18.7	20.6	0



Reordering a Dissimilarity matrix cont.

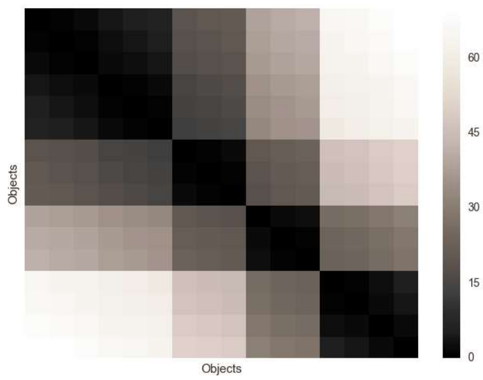


Example dataset with 16 objects



Random order of the 16 objects for the dissimiliarity matrix

Reordering the matrix reveals the clusters

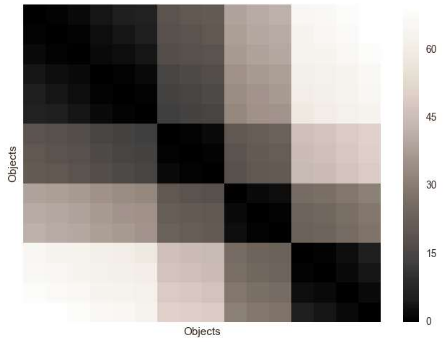


A better ordering of the 16 objects. Nearby objects in the ordering are similar to each other, producing large dark blocks. We can see four clusters along the diagonal.

THE UNIVERSITY OF MELBOURNE

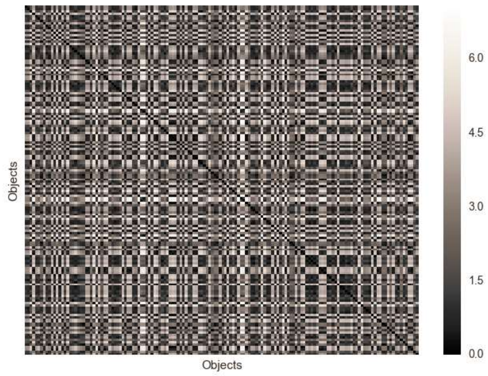
VAT images

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



THE UNIVERSITY OF MELBOURNE

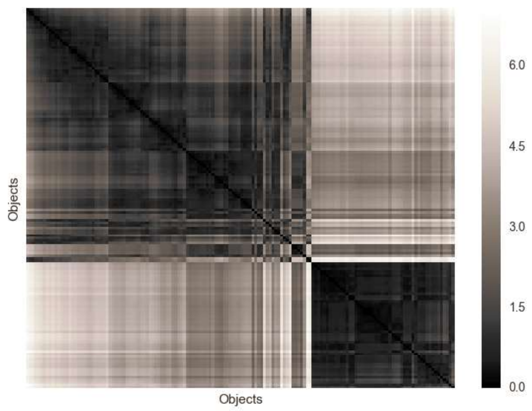
Another example: dissimilarity matrix for Iris dataset



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

THE UNIVERSITY OF MELBOURNE

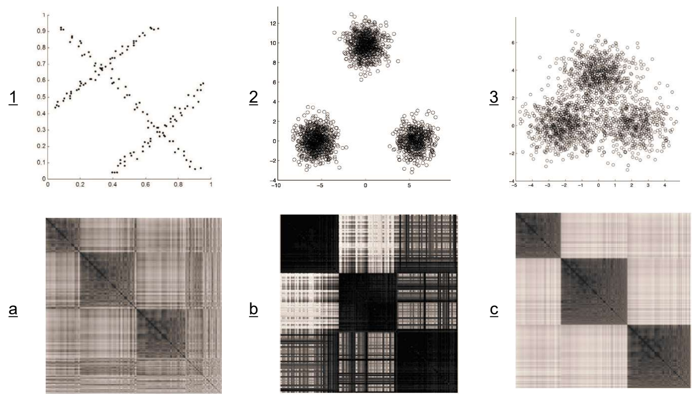
Dissimilarity matrix for Iris data (reordered)



THE UNIVERSITY OF MELBOURNE

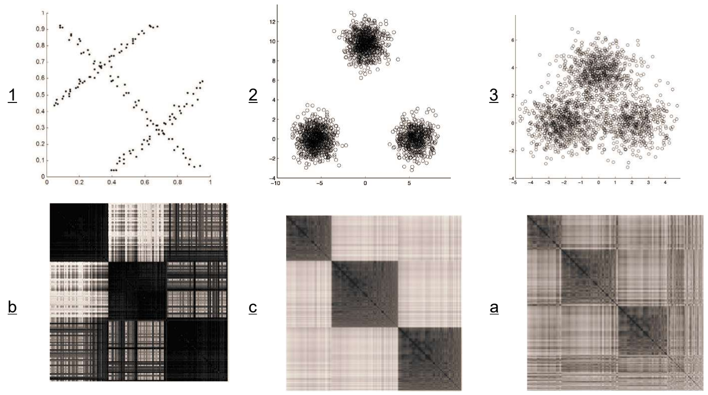
Exercise - *An algorithm for clustering Tendency Assessment*, Hu, Y., Hathaway, R. WSEAS Transactions on Mathematics, 2008.

- Match the datasets with the VAT images



Exercise - *An algorithm for clustering Tendency Assessment*, Hu, Y., Hathaway, R. WSEAS Transactions on Mathematics, 2008.

- Match the datasets with the VAT images



VAT ordering example

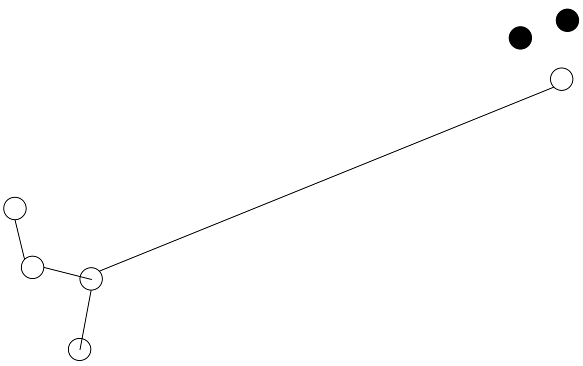
VAT ordering example

VAT ordering example

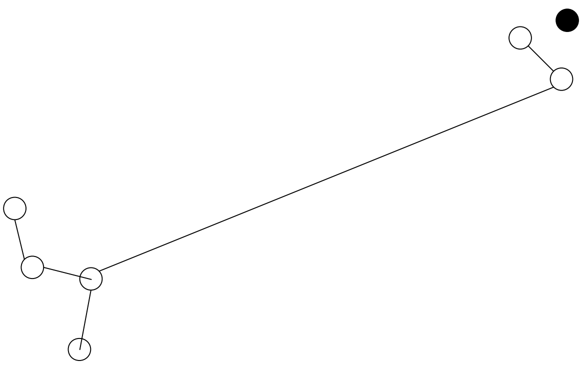
VAT ordering example

VAT ordering example

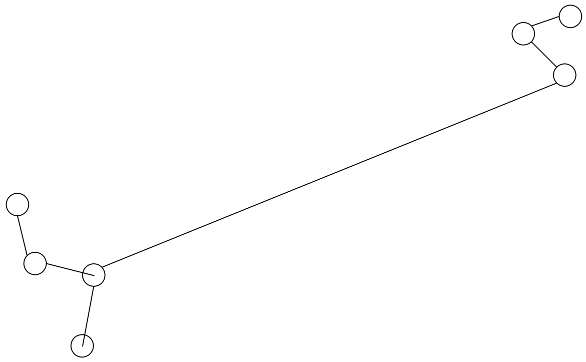
VAT ordering example



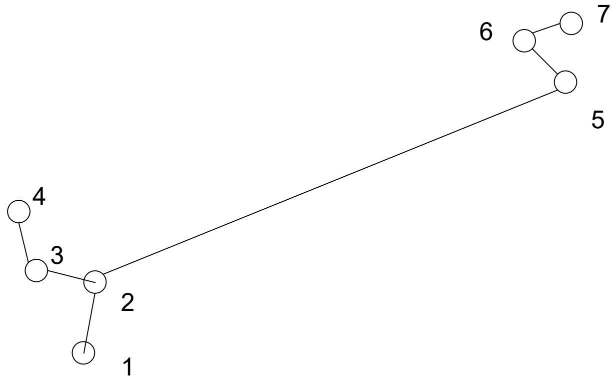
VAT ordering example



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'

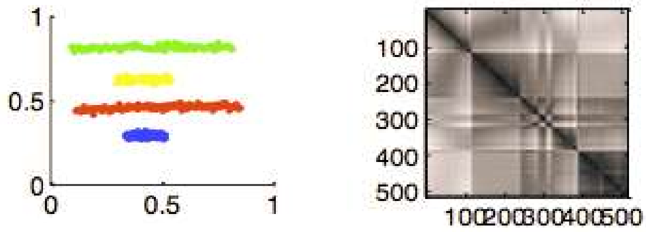
P(r) = j; I = I ∪ {j}; J=J - {j}; ### '∪' means 'union'

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 ….

Dissimilarity matrix cont.

- VAT algorithm won't be 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.



VAT algorithm python code

- You will practice in workshop

Computer Associates Video

- An application of VAT, role discovery in company data
  http://www.youtube.com/watch?v=l3tkUpGTTmQ&authuser=0

- In this video, they represent each employee by a vector describing their level of access to various organisational entities

Employee	Entity 1	Entity 2	…	Entity N
James	Yes	No	….	Yes
Bob	No	No	….	Yes

Can also represent each entity by a vector (row) describing which employees have access to it

Entity	James	Bob	…	Kate
1	Yes	No		No

Acknowledgements

- Material partly adapted from
  - "Data Mining Concepts and Techniques", Han et al, 2^{nd} edition 2006.