H4016 Text Mining and Information Retrieval

Data Mining, Year 3
Topic 7: Clustering

Ref: Tan et al, Introduction to Data Mining, chpts 8 & 9

Overview

- What is cluster analysis?
- Examples from industry
- Clustering Algorithms:
  - K Means
  - DBScan
- Distance Measure: Manhattans and Euclidean distance
- Evaluating Clusters

What is cluster analysis

- Algorithms looked at so far in this module are classification algorithms, which predict a class variable.
- What if you don't have a class variable, or don't want to predict anything, you just want to understand patterns in the dataset, i.e. understand which groups of rows are similar, and in what way are they similar?

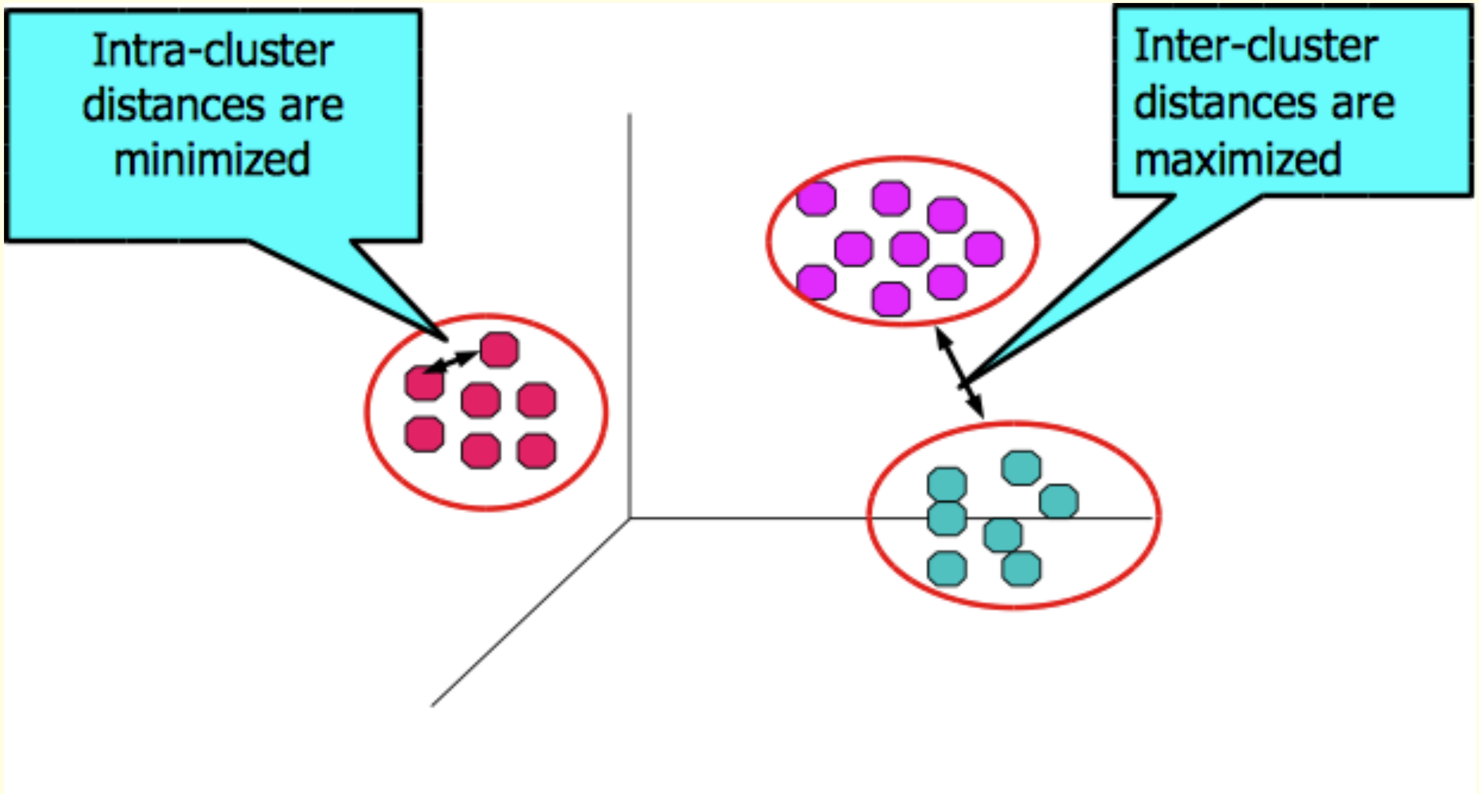
This form of analysis is called **clustering**

- the objective is NOT to predict anything
- but to understand the naturally occurring groups/clusters of rows in the dataset.

- Because the user doesn't define the groups in advance (i.e. there is no class label to group rows together), clustering is referred to as UNSUPERVISED LEARNING. Classification is referred to as supervised learning.

Introduction to clustering

The objective in clustering is to group objects such that objects within a cluster have a high similarity in comparison with one another, but are dissimilar to objects in other clusters.

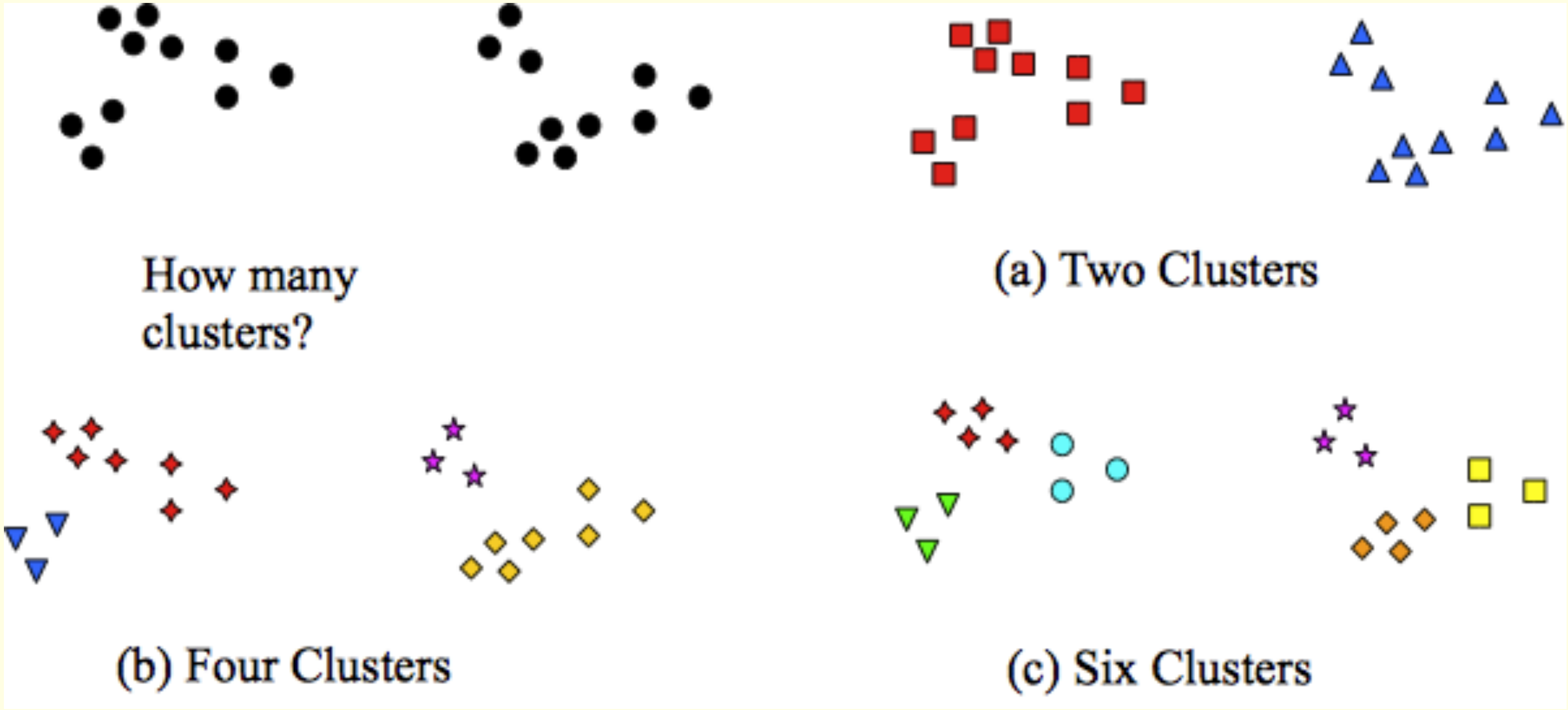

Intra-cluster distances are minimized
Inter-cluster distances are maximized

Clustering algorithms differ in how they compare objects to assess how similar they are.

Introduction to clustering

In many cases, the notion of a cluster is not well defined.

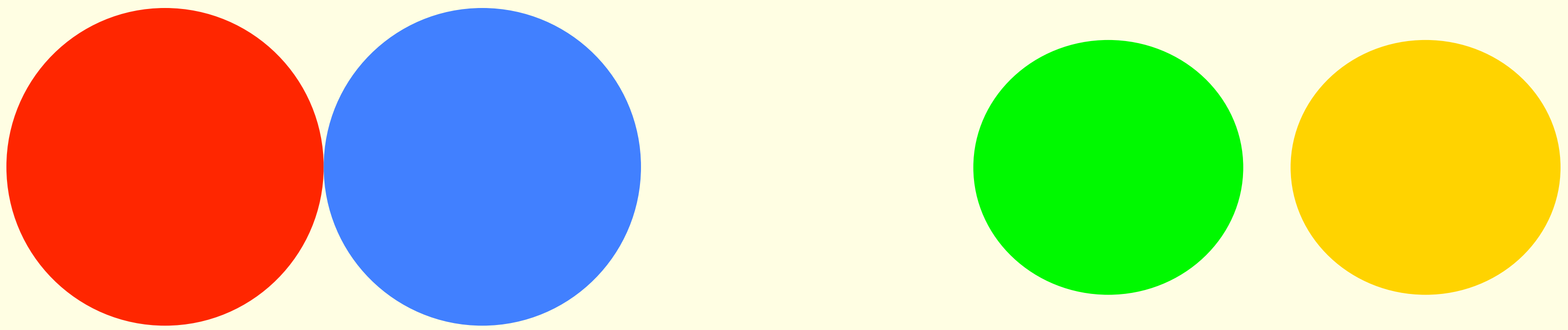
The diagram below shows 20 points (rows in a dataset), and three different ways of dividing them into clusters. The shapes of the markers indicate cluster membership.



Types of Clusters: Center-Based

Center-based cluster definitions

- A cluster is a set of objects such that an object in a cluster is closer (more similar) to the "center" of a cluster, than to the center of any other cluster

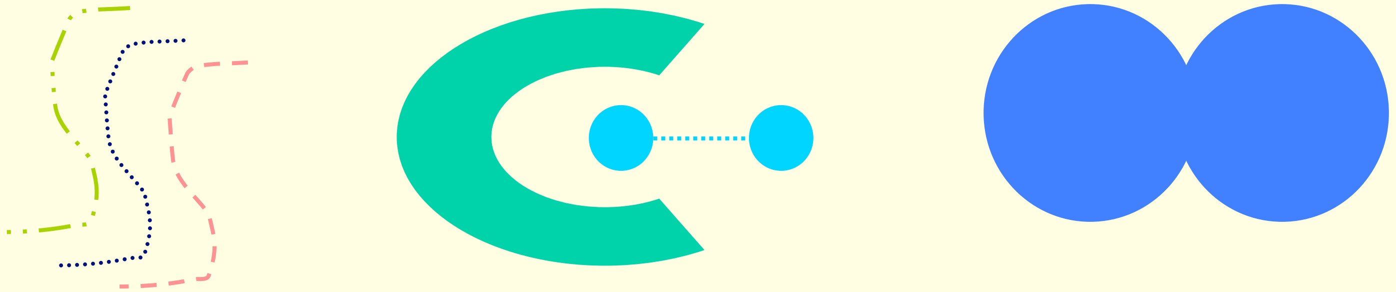


4 center-based clusters

Types of Clusters: Contiguity-Based

Contiguous Cluster (Nearest neighbor)

A cluster is a set of points such that a point in a cluster is closer (or more similar) to one or more other points in the cluster than to any point not in the cluster.

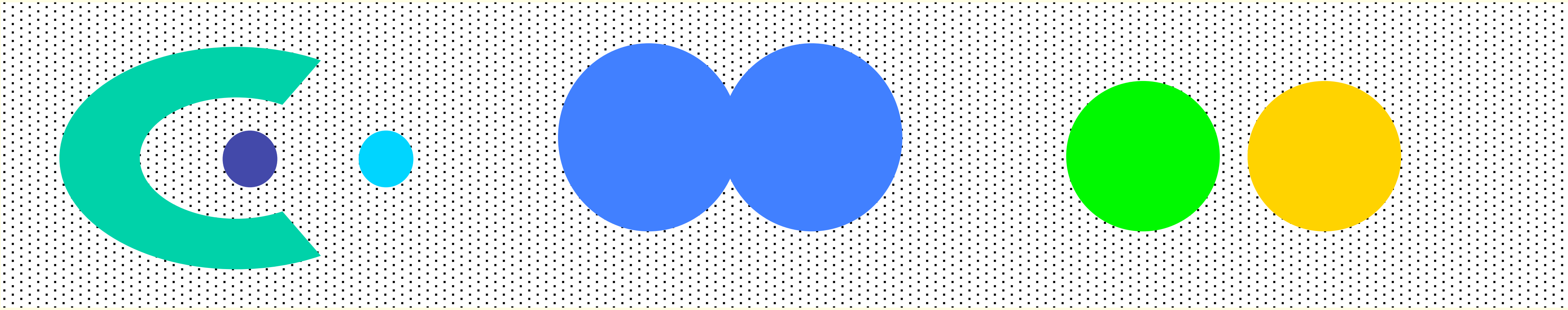


8 contiguous clusters

Types of Clusters: Density-Based

Density-based

- A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
- Used when the clusters are irregular or intertwined, and when noise and outliers are present.



6 density-based clusters

Applications of Cluster Analysis

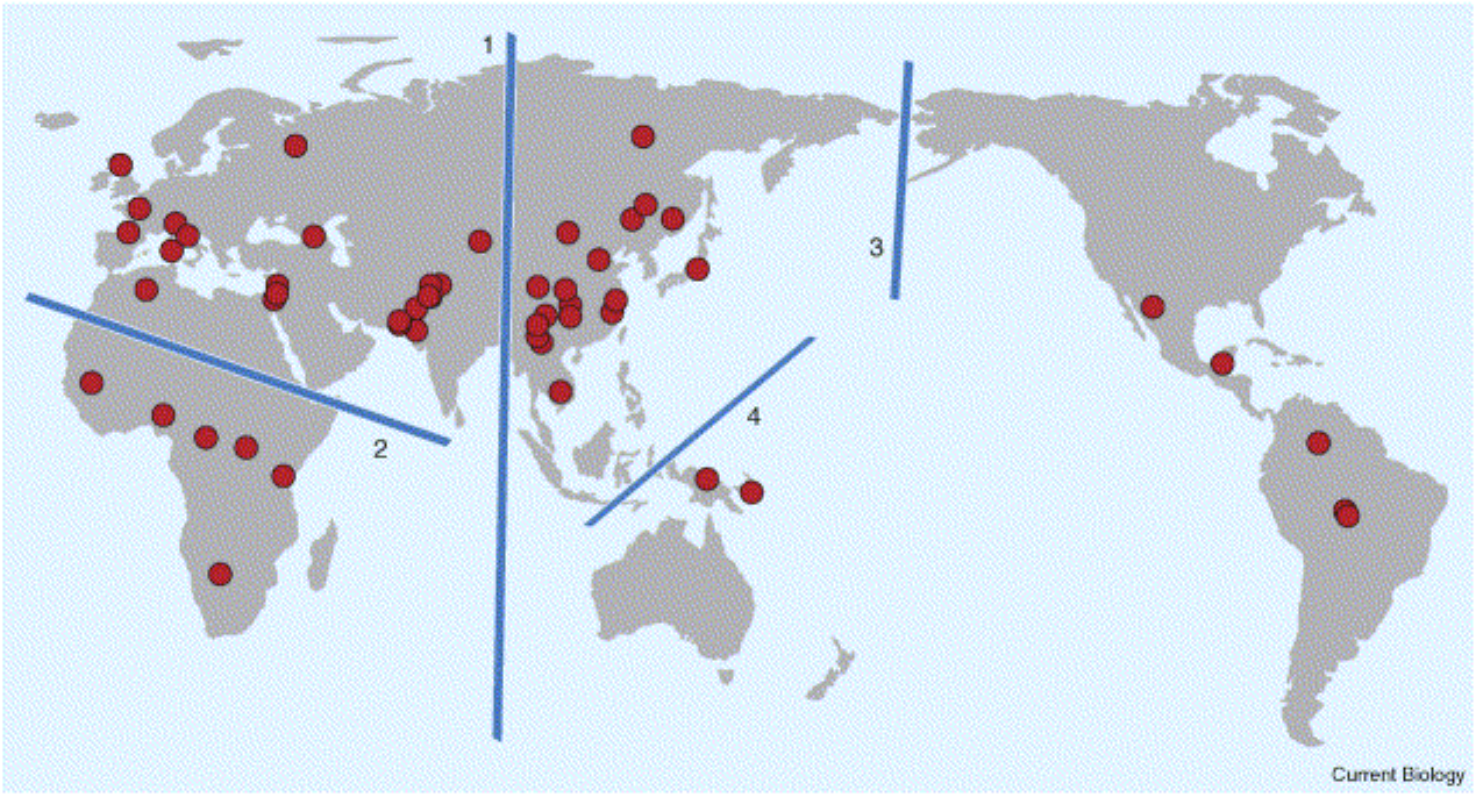
Business:

- Cluster analysis is widely used in market research to partition the general population of consumers into market segments and to better understand the relationships between different groups of consumers/potential customers, and for use in market segmentation, Product positioning, New product development and Selecting test markets.

Applications of clustering

Biology:

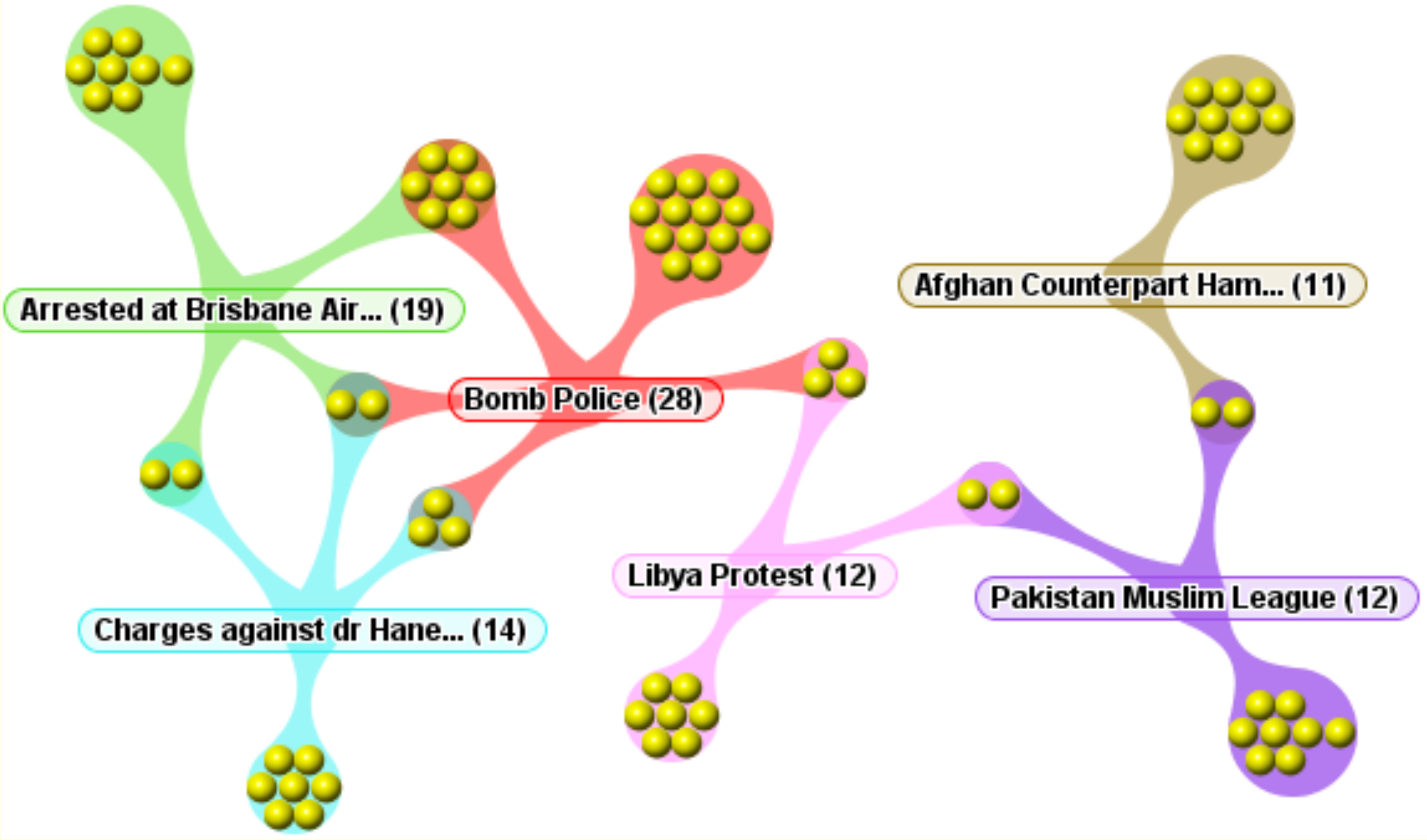
- Describe and make spatial and temporal comparisons of communities of organisms.
- Build groups of genes with related expression patterns
- The similarity of genetic data is used in clustering to infer population structures



Current Biology

Figure 1. Geographical location of the 52 population samples studied by Rosenberg *et al.* [7]. The barriers numbered 1 to 4 correspond to the sequential partition of the sampled populations into genetic clusters.

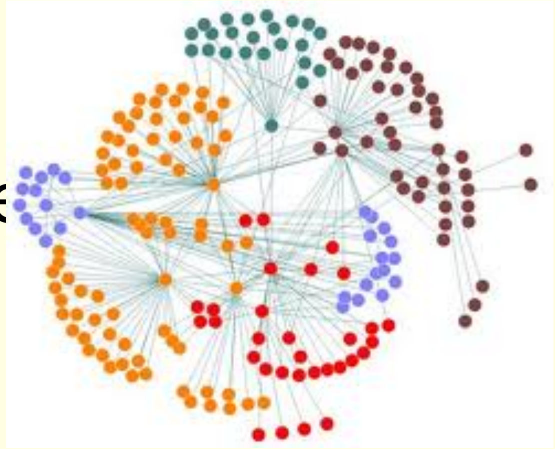
Clustering text documents


Arrested at Brisbane Air... (19)
Bomb Police (28)
Afghan Counterpart Ham... (11)
Libya Protest (12)
Pakistan Muslim League (12)
Charges against dr Hane... (14)

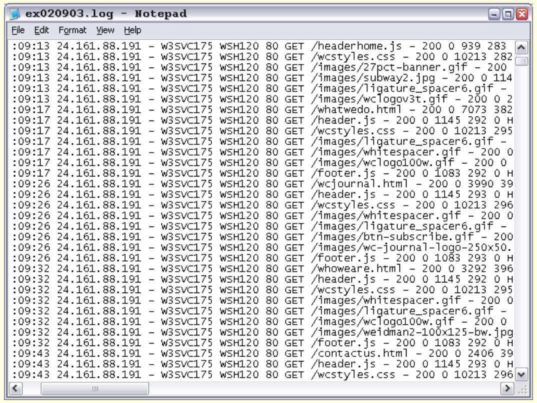
Applications of clustering

World Wide Web

- Social network analysis
  - In the study of social networks, clustering may be used to recognize communities within large groups of people.



- Cluster web log data
  - Discover web site usage patterns, and recurring access patterns

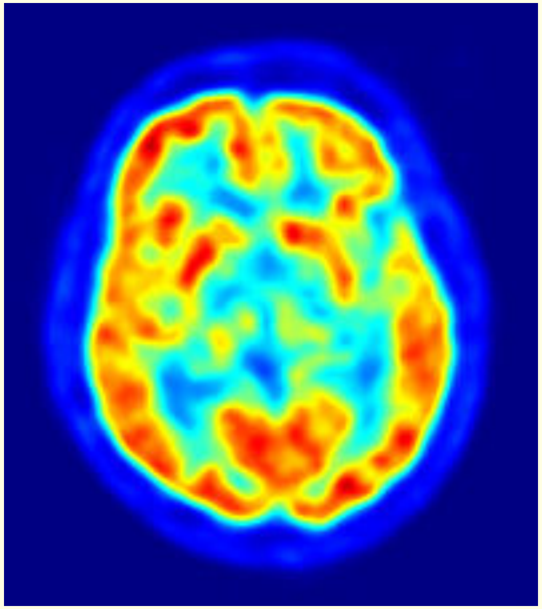


```
:09:13 24.161.88.191 - W3SVC175 WSH120 80 GET /headerhome.js - 200 0 939 283
:09:13 24.161.88.191 - W3SVC175 WSH120 80 GET /wcstyles.css - 200 0 10213 282
:09:13 24.161.88.191 - W3SVC175 WSH120 80 GET /images/27pct-banner.gif - 200
:09:13 24.161.88.191 - W3SVC175 WSH120 80 GET /images/subway2.jpg - 200 0 114
:09:13 24.161.88.191 - W3SVC175 WSH120 80 GET /images/ligature_spacer6.gif -
:09:13 24.161.88.191 - W3SVC175 WSH120 80 GET /images/wclogov3t.gif - 200 0 2
:09:17 24.161.88.191 - W3SVC175 WSH120 80 GET /whatwedo.html - 200 0 7073 382
:09:17 24.161.88.191 - W3SVC175 WSH120 80 GET /header.js - 200 0 1145 292 0 H
:09:17 24.161.88.191 - W3SVC175 WSH120 80 GET /wcstyles.css - 200 0 10213 295
:09:17 24.161.88.191 - W3SVC175 WSH120 80 GET /images/ligature_spacer6.gif -
:09:17 24.161.88.191 - W3SVC175 WSH120 80 GET /images/whitespacer.gif - 200 0
:09:17 24.161.88.191 - W3SVC175 WSH120 80 GET /images/wclogo100w.gif - 200 0
:09:17 24.161.88.191 - W3SVC175 WSH120 80 GET /footer.js - 200 0 1083 292 0 H
:09:26 24.161.88.191 - W3SVC175 WSH120 80 GET /wcjournal.html - 200 0 3990 39
:09:26 24.161.88.191 - W3SVC175 WSH120 80 GET /header.js - 200 0 1145 293 0 H
:09:26 24.161.88.191 - W3SVC175 WSH120 80 GET /wcstyles.css - 200 0 10213 296
:09:26 24.161.88.191 - W3SVC175 WSH120 80 GET /images/whitespacer.gif - 200 0
:09:26 24.161.88.191 - W3SVC175 WSH120 80 GET /images/ligature_spacer6.gif -
:09:26 24.161.88.191 - W3SVC175 WSH120 80 GET /images/btn-subscribe.gif - 200
:09:26 24.161.88.191 - W3SVC175 WSH120 80 GET /images/wc-journal-logo-250x50.
:09:26 24.161.88.191 - W3SVC175 WSH120 80 GET /footer.js - 200 0 1083 293 0 H
:09:32 24.161.88.191 - W3SVC175 WSH120 80 GET /whoweare.html - 200 0 3292 396
:09:32 24.161.88.191 - W3SVC175 WSH120 80 GET /header.js - 200 0 1145 292 0 H
:09:32 24.161.88.191 - W3SVC175 WSH120 80 GET /wcstyles.css - 200 0 10213 295
:09:32 24.161.88.191 - W3SVC175 WSH120 80 GET /images/whitespacer.gif - 200 0
:09:32 24.161.88.191 - W3SVC175 WSH120 80 GET /images/ligature_spacer6.gif -
:09:32 24.161.88.191 - W3SVC175 WSH120 80 GET /images/wclogo100w.gif - 200 0
:09:32 24.161.88.191 - W3SVC175 WSH120 80 GET /images/weidman2-100x125-bw.jpg
:09:32 24.161.88.191 - W3SVC175 WSH120 80 GET /footer.js - 200 0 1083 292 0 H
:09:43 24.161.88.191 - W3SVC175 WSH120 80 GET /contactus.html - 200 0 2406 39
:09:43 24.161.88.191 - W3SVC175 WSH120 80 GET /header.js - 200 0 1145 293 0 H
:09:43 24.161.88.191 - W3SVC175 WSH120 80 GET /wcstyles.css - 200 0 10213 296
```

Applications of Cluster Analysis

Medicine

- Image analysis, e.g. distinguishing between areas of blood and tissue in a PET Scan



PET scan of human brain

Clustering Algorithms

There are a large variety of clustering algorithms.
It is advisable to use several algorithms to see what the data might disclose.
The four most common categories of clustering algorithms are as follows:

Clustering Algorithms

1. Partitioning methods – you specify the number of clusters in advance, and an algorithm organises objects into the specified number of clusters. Each object is allocated to exactly one cluster (complete clustering), and so clusters are non-overlapping.

2. Hierarchical methods – clusters are organized into a tree structure, where children represent the optimal slit of the parent group. Each parent is a union of its child sub-clusters. The Root node is all objects, and frequently the leaf nodes contain one element. At any one level in the tree, each object is allocated to exactly one cluster, and so clusters are non-overlapping. As every object is allocated to exactly one cluster, it is also complete clustering.

Clustering Algorithms

3. Grid-based methods – organises objects into a grid structure such that clusters at opposite ends of the grid have the largest distance measurement. Those left in the center do not naturally belong to any cluster (partial clustering). This is useful for identifying outliers, i.e. objects that do not easily fall into a cluster. Its non-overlapping.
4. Density based methods – Used where there are dense areas of objects separated by areas of low density. This method identifies more randomly shaped cluster than the previous methods above. It also excludes noise points (partial clustering), but objects can not belong to more than one cluster (non-overlapping).

- We will now look at two algorithms in more detail. . .
  1. Kmeans (a partitioning method)
  2. DBScan (a density based method)

1. K-Means (Partitioning method)

- Given a database of *n* tuples (rows), a partitioning method constructs *k* partitions of the database, such that *k* ≤ *n*.
- Each group must contain at least one row of data, and each row of data must belong to exactly one group.
- **The number of groups, *k*, is specified in advance.**
- A good partitioning will result in objects from the same cluster being ‘close’ or related to each other, while objects from different clusters are “far apart” or very different.

- The most popular partitioning algorithm is the **k-means** algorithm which works as follows:

1. K-Means (Partitioning method)

1. *k* rows of the data set are selected at random, each of which initially represents a cluster mean or centre
2. For each remaining object, the object is assigned to the cluster to which it is most similar, based on the similarity between it and the cluster's mean.
3. Once all objects are allocated to a cluster, a new mean is calculated for each cluster.
4. All tuples in the database are now compared to this new mean, and again each object is allocated to the cluster to which it is most similar.
5. Once all objects a re-allocated, a new mean is again calculated for the cluster and the process is done again.
6. This process iterates until no document has moved to a new bin, or the distances between documents and their cluster centre converges.

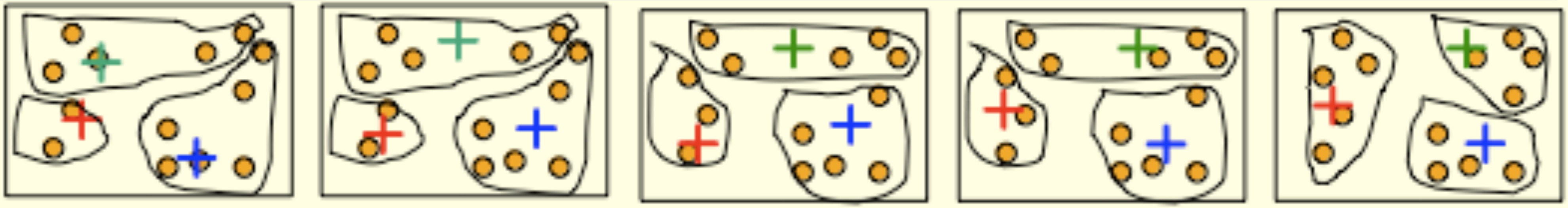


Diagram 1: The first three rows representing the initial cluster centres (marked with green red and blue crosses respectively) are chosen at random. Three clusters are formed based on the proximity of other points to these three.

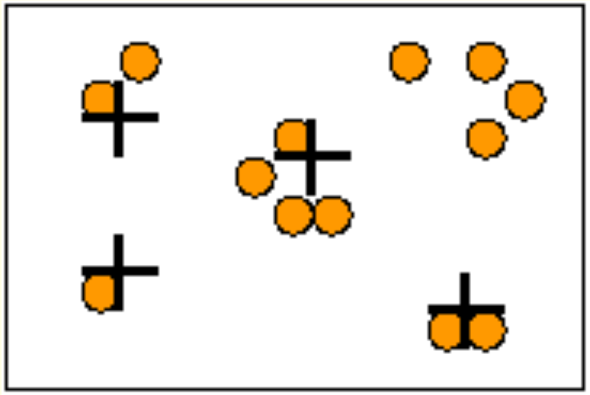
Diagram 2: A new mean is calculated for each cluster, indicated by the new positions of the crosses in diagram 2 above.

Diagram 3: Each point is now reallocated to a cluster based on its proximity to the new cluster centres. This changes the composition of each of the three clusters as shown.

Diagram 4: The centre point of these new clusters is calculated, indicated by the new positions of the crosses in diagram 4 above.

Diagram 5: Each point is again reallocated to a cluster based on its proximity to each new cluster centre. This again changes the composition of each of the three clusters.

Exercise

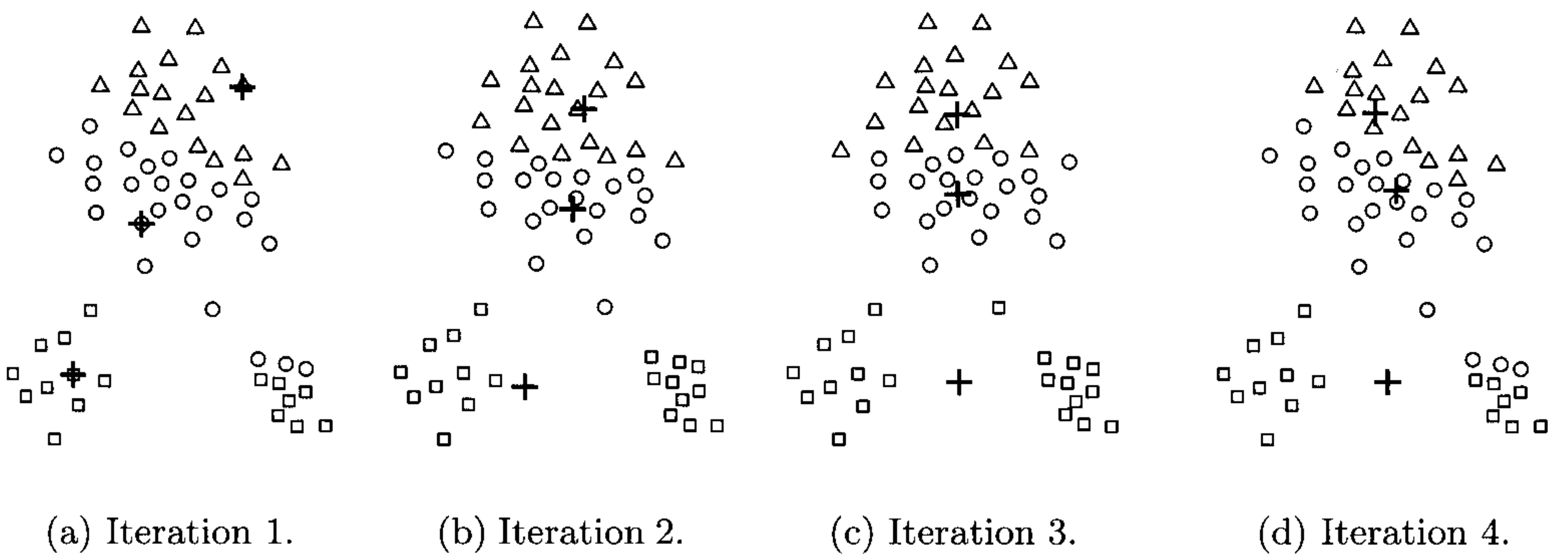


Track the iterations of a k-means clustering algorithm in the diagram above where *k* is 4.

1. K-Means (Partitioning method)

- Strengths and Weaknesses
  - ✓ K-means is the most widely used method for statistical clustering.
  - ✓ It is computationally inexpensive
  - ✗ However you need to specify K in advance. The algorithm does not perform well if K does not match the actual number of clusters in the data set
  - ✗ It is also adversely effected by outliers.
  - ✗ It performs poorly if clusters are not distinct
  - ✗ The outcome of the algorithm depends on the initial cluster centres chosen, which is random

1. K-Means (Partitioning method)

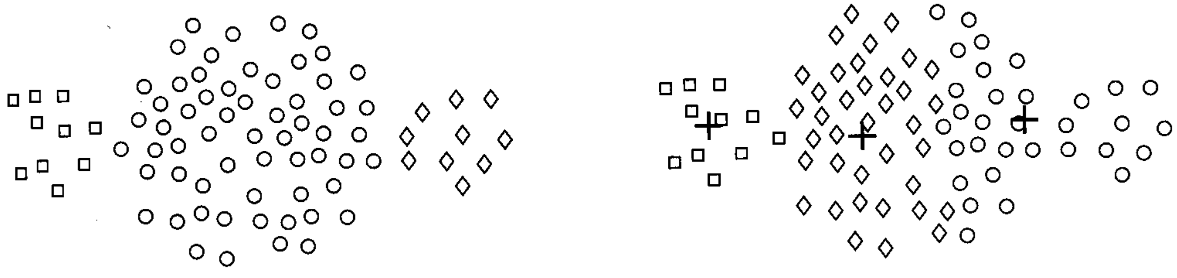


(a) Iteration 1. (b) Iteration 2. (c) Iteration 3. (d) Iteration 4.

Illustration of poor starting centroids (from Tan/Steinbach/Kumar 'Introduction to Data Mining', pg 503)

1. K-Means (Partitioning method)

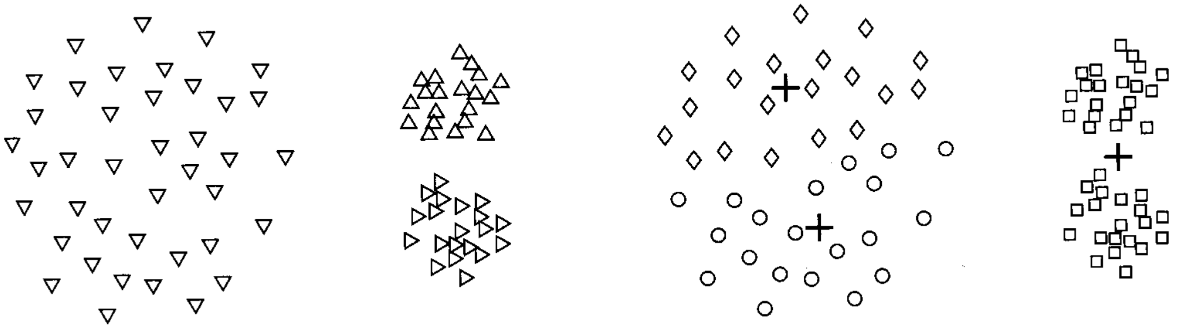
Limitations of K-Means (from Tan/Steinbach/Kumar 'Introduction to Data Mining', pg 511)



(a) Original points.

(b) Three K-means clusters.

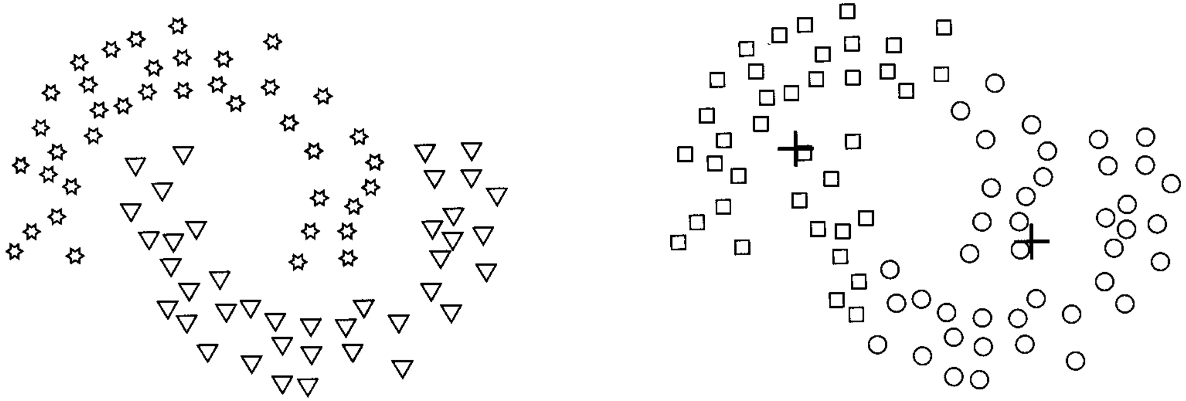
Figure 8.9. K-means with clusters of different size.



(a) Original points.

(b) Three K-means clusters.

Figure 8.10. K-means with clusters of different density.



(a) Original points.

(b) Two K-means clusters.

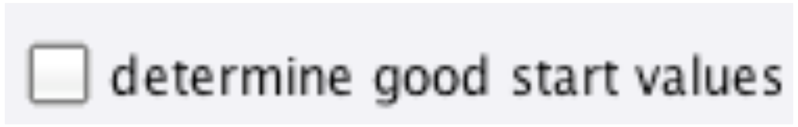
Figure 8.11. K-means with non-globular clusters.

Variation on k-means

K-Means++: is a variation on k-means that attempts to improve on how the initial cluster centres are selected.

- The first cluster centre is picked at random.
- All rows are then weight based on their proximity to the cluster centre so that the further away they are, the better chance they have of being pick as the next cluster centre. This continues iteratively, i.e.
  - Before selecting each remaining initial cluster centre: iteratively all rows are weighted based on their proximity to their current closest cluster centre, influencing their likelihood of being selected as the next initial cluster centre.

- K-means++ aims to improve the placement of initial cluster centres.



□ determine good start values

K-Means in RapidMiner

Open sample/processes/07_clustering/01_kmeans.xml which uses the k-means algorithm to cluster the Iris data set.

K-means parameters:

Add_cluster_attribute - adds a cluster ID to the dataset.

Add_as_label – give a clusterID a role of label to allow a classification aglrothm attempt to predict the cluster ID. Change this to true (check box).

K – number of clusters

Max_runs – how many times to run k-means. Each run starts with an initial k, randomly chosen cluster centroids.

Max_optimisation steps – within each run, how many optimisation steps (regrouping rows into clusters) are preformed.

Disable the SVD Reduction operator.
Run the process.

K-Means in Rapid Miner

The exampleset (retrieve) tab shows the original table with a cluster attribute added.

The cluster model output has five views:

- Text view shows the size of each cluster
- Folder view and graph view list the object_ids in each cluster
- Centroid table shows, for each cluster, the value of the cluster centroids for each attribute
- Centroid plot view graphically illustrates how attribute values vary across each cluster, using a parallel plot.

DBScan

Density based clustering locates regions of high density that are separated by regions of low density in a state space of objects.

One example of a density based clustering algorithm is DBSCAN, which is a simple but effective algorithm

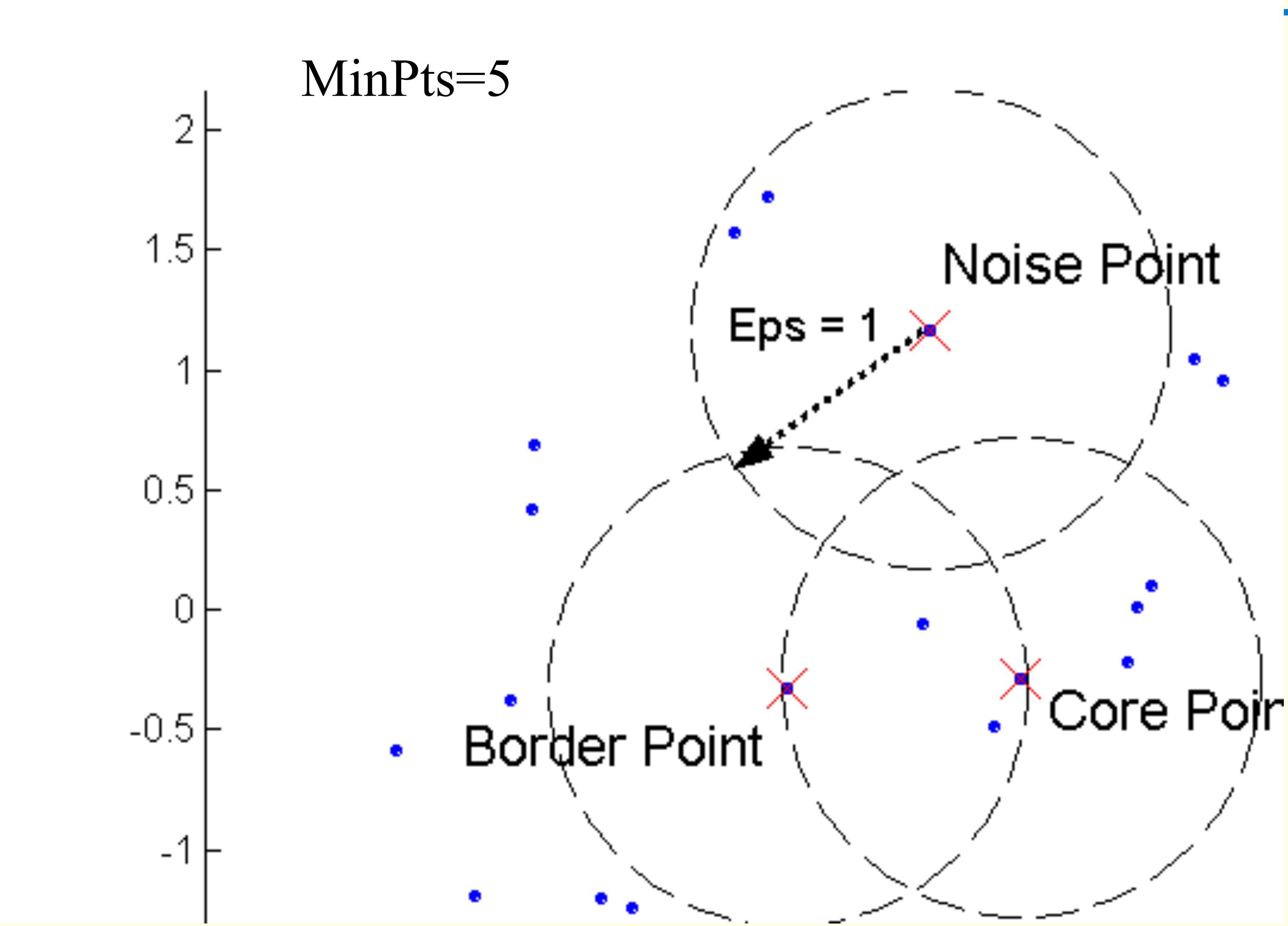
Density is estimated for a particular point in the data set by counting the number of points within a specified radius, *Eps*, of that point. This includes the point itself. Each point can be classified as one of the following:

A **core** point is a point in the interior of a dense region, and is defined as having more than a specified number of points (*MinPts*) within a radius of *Eps*.

A **border** point is a point on the edge of a dense region. It has less than *MinPts* within *Eps* but is in the neighbourhood of a core point.

A **noise** point is any point that is not core or border.

DBScan



Neighbouring core points and their border points are identified as a cluster. Noise points are not included in a cluster.

DBScan

- The values of *MinPts* and *Eps* must be decided before points can be classified.
  - These two values are parameter settings in RapidMiner

- The appropriate values for these will depend on:
  - The random distribution of points:
    - how far apart, on average, are points not belonging to a cluster &
    - how far apart, on average, are points within a cluster.
  - The density of the clusters – how many points would typically be ‘close by’ for points within a cluster

- You need to experiment with these parameters to get the best value.

DBScan – Strengths and Weaknesses

✓ It is relatively resistant to noise, as it disregards noise points, and can handle clusters of arbitrary shapes and sizes.

✓ You DON'T need to know the number of clusters in advance

X However it assumes all clusters have the same density:

- i.e: the distances between points in each cluster is similar

X Its computationally expensive

DBScan in RapidMiner

An example of DBSCAN can be found at ../sample/prcesses/07_clustering/12_DBScan.xml

Run the process

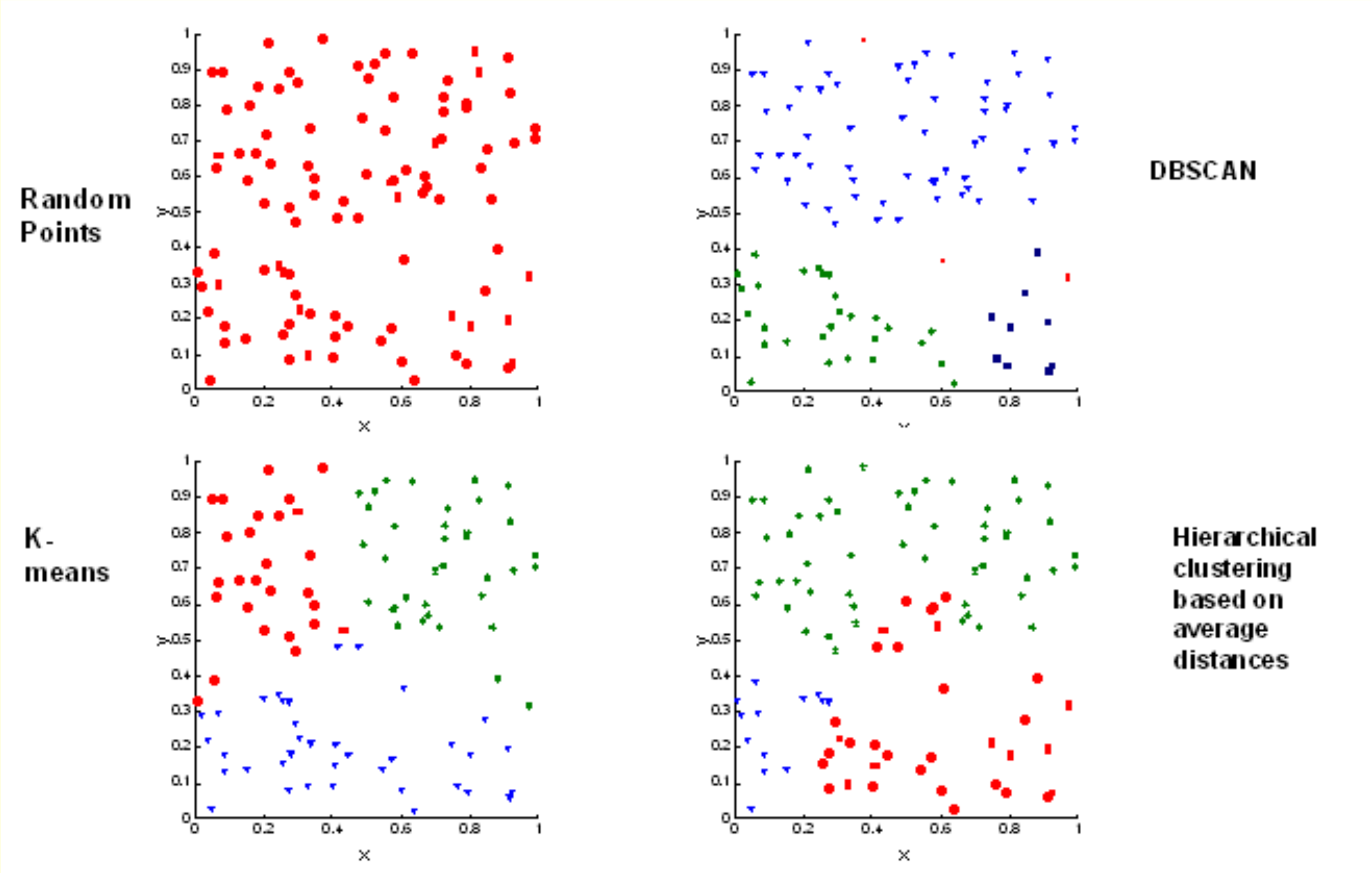
Plot the data set using a scatter colour plot, plotting Atr1 against Atr2, with colour set to cluster.

Exercise:
Model this data using K-Means, and set K=3. Can it find the three clusters?

Comparison of Algorithms

Algorithm	Scalable	Handles outliers /noise	Specify number of clusters in advance	Find arbitrary shapes.
K-Means	Yes (fastest algorithm for large dataset)	No	Yes	No
DBScan	No	Yes	No	Yes

Comparison of Algorithms



Distance measures

Manhattan distance

The simplest distance measure for numeric attributes is the **Manhattan** distance measure as detailed below. The equation is as follows:

$$d(x, y) = \sum_{k=1}^{n} |x_k - y_k|$$

For example: the Manhattan distance between Person X and Person Y in the example below would be:

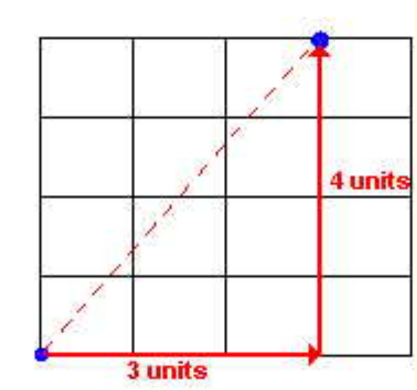
Distance(X, Y) = 20+30+1 = 51

Object	Age	Bank Balance	Credit Rating
Person X	25	€500.00	2
Person Y	45	€530.00	3

Manhattan distance

The Manhattan distance gets it's name because computationally it is equivalent to taking a 'city walk' between two points, as can be seen from the solid red arrows in the diagram (walk south 4 blocks, and east 3 blocks).

It is useful for discrete data sets, as the answer will always be a discrete number (whole unit).



For data points:
(0,0) and (3,4),
distance is
(3-0) + (4-0) = 7

Euclidean Distance

There are a range of other methods for calculating the distance between two rows of data, but the most common method is Euclidean Distance. It is calculated as follows:

$$d(x, y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$

where *x* and *y* are the two objects; *n* is the number of attributes, and x_k and y_k is the *k^th^* attribute of *x* and *y* respectively.

lets look at an example . . . . .

Euclidean Distance

For example, take the same two objects with three attributes:

Object	Age	Bank Balance	Credit Rating
Person X	25	€500.00	2
Person Y	45	€530.00	3

The Euclidean distance is calculated as follows:

$$\text{distance(person X, person Y)} = \sqrt{(25\text{-}45)^2 + (500\text{-}530)^2 + (1)^2}$$

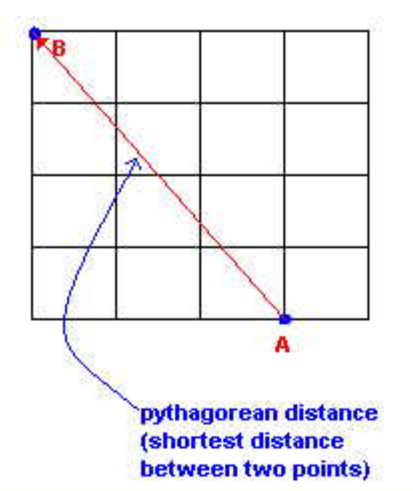
$$= \sqrt{(-20)^2 + (-30)^2 + (1)^2}$$

$$= \sqrt{400 + 900 + 1}$$

$$= \sqrt{1301} \qquad = \mathbf{36.07}$$

Euclidean Distance

Note: The formula above for Euclidean distance is based on Pythagorean distance (Pythagoras theorem) and finds the shortest distance between two points.



Note: Age and bank balance determined the outcome. The credit rating attribute was not significant in this calculation.

To increase the importance of credit rating, all values should be converted to the same scale.

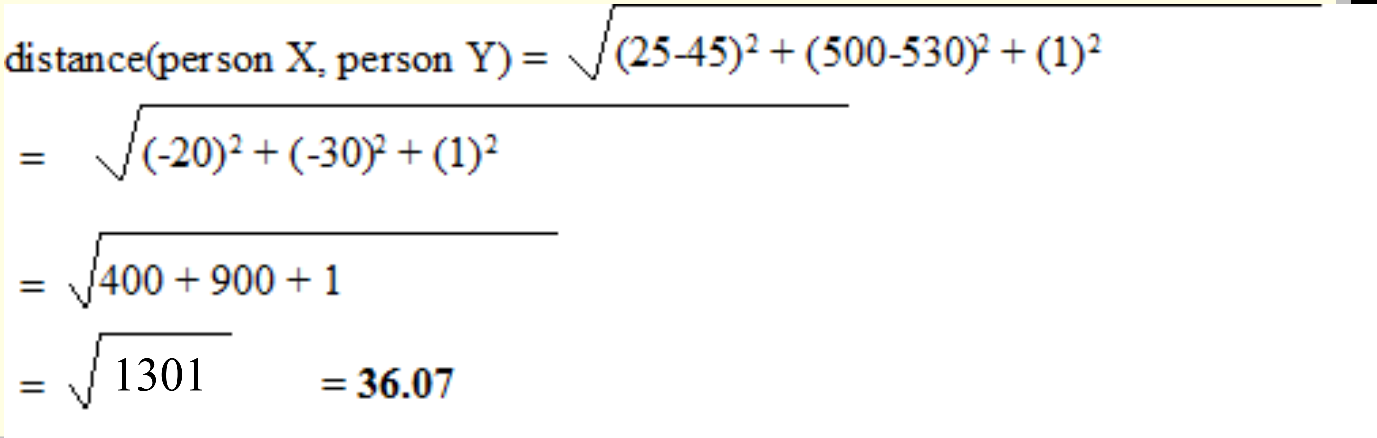
Mixed Euclidean

In the previous example, all attributes were numeric.
Mixed Euclidean adapts this for nominal attributes as follows:

- distance for nominal attributes is 1 if they different, and 0 if they are the same.

For example, take the two objects from our earlier example, with credit rating left as nominal:
The Mixed Euclidean Distance would be the same as on slide 45, since distance (poor, fair) = 1

Object	Age	Bank Balance	Credit Rating
Person X	25	€500.00	poor
Person Y	45	€530.00	fair



$$\text{distance(person X, person Y)} = \sqrt{(25\text{-}45)^2 + (500\text{-}530)^2 + (1)^2}$$

$$= \sqrt{(-20)^2 + (-30)^2 + (1)^2}$$

$$= \sqrt{400 + 900 + 1}$$

$$= \sqrt{1301} \quad = \mathbf{36.07}$$

Exercise

Calculate the Euclidean distance and the Manhattan distance between the two rows of data below. All attributes have been normalised to the scale [0,10]

Object	Age	Bank Balance	Credit Rating
Person X	10	8	2
Person Y	8	5	4

Some points to consider . . .

Numeric Variables

The **measurement** unit used will impact on distance calculations.

- For example measurements taken in centimetres will generate greater distances than measurements taken in metres.
- There will also be a larger range of values in a centimetre scale than in a meter scale (e.g. object 2 and object 3 below both round to the same value in metres).

Numeric Variables

	Diameter in centimetres	Diameter in metres
Object 1	100	0.1
Object 2	540	0.5
Object 3	520	0.5
Object 4	1400	1.4

Range of values in centimetres: [100, 1400]

Range of values in meters: [0.1, 1.4]

d(object1, object2) = 440cm or 0.4m
d(object1, object3) = 420cm or 0.4m
d(object2, object3) = 20cm or 0m
d(object1, object4) = 1300cm or 1.3m
d(object2, object4) = 860cm or 0.9m

Numeric Variables

- The centimetres figures will have a much bigger impact on measuring distances between objects than the metre figures, and so will have a greater influence on how the data is clustered.
- If all variables have equal importance, then all data ranges should be standardised using a technique such as normalisation (scaling).
- Where a greater emphasis needs to be placed on certain variables, their measurements should be standardised to a larger range of values, e.g. [0,10] rather than [0,1].

Exponential Increases

Where an attribute represents values that increase exponentially, such as the growth of a bacteria population, distances will be skewed by the rate at which the values are increasing, giving them a greater weighting than other variables.

This can be overcome by replacing each value by its $\log_{10}$.

x	2	4	16	256	65536	4294967296	1.84467E+19	3.40282E+38	1.15792E+77
logx	0.30	0.60	1.20	2.41	4.82	9.63	19.27	38.53	77.06

Ordinal Values

Ordinal variables have an inherent ranking, e.g. hot, mild, cold, freezing.

Each of these values should be replaced by a number representing it's rank as shown below.

Variable	Freezing	Cold	Mild	hot
Rank	1	2	3	4

Variables are then treated as numeric.

Assumes equal intervals . . .

This approach assumes **equal intervals** between each value.

In other words it assume the distance between *Freezing* and *Cold* is the same as the distance between *Cold* and *Mild*.

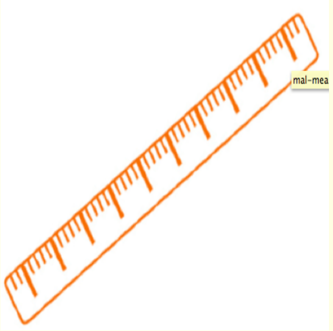
If these intervals were even, then an interval or ratio value would probably have been used in the first place. However in the absence of more information, this is the standard approach.

Cluster Evaluation

Cluster Evaluation

Cluster Evaluation can be

1. Subjective: Investigate the make up of each cluster, and make a judgement call on how well the data was segmented into clusters. This is time consuming, and subject to human error



2. Objective: use metrics to measure the performance of the clustering algorithm. Metrics would be based on inter- and intra- cluster distances

Subjective Evaluation

- Subjective Evaluation is comprised of a domain expert ‘looking’ at the make up of each cluster to decide if the grouping of rows ‘makes sense’ in the context of the business objective.

- Tools that can aid the process include:

  - Cluster Visualisation tools:

    - There are a range of such tools

  - A classification algorithm such as a Decision Tree predicting cluster membership

Using a decision tree:

A clustering algorithm will add a clusterID attribute to the dataset, allocating each row to a cluster.

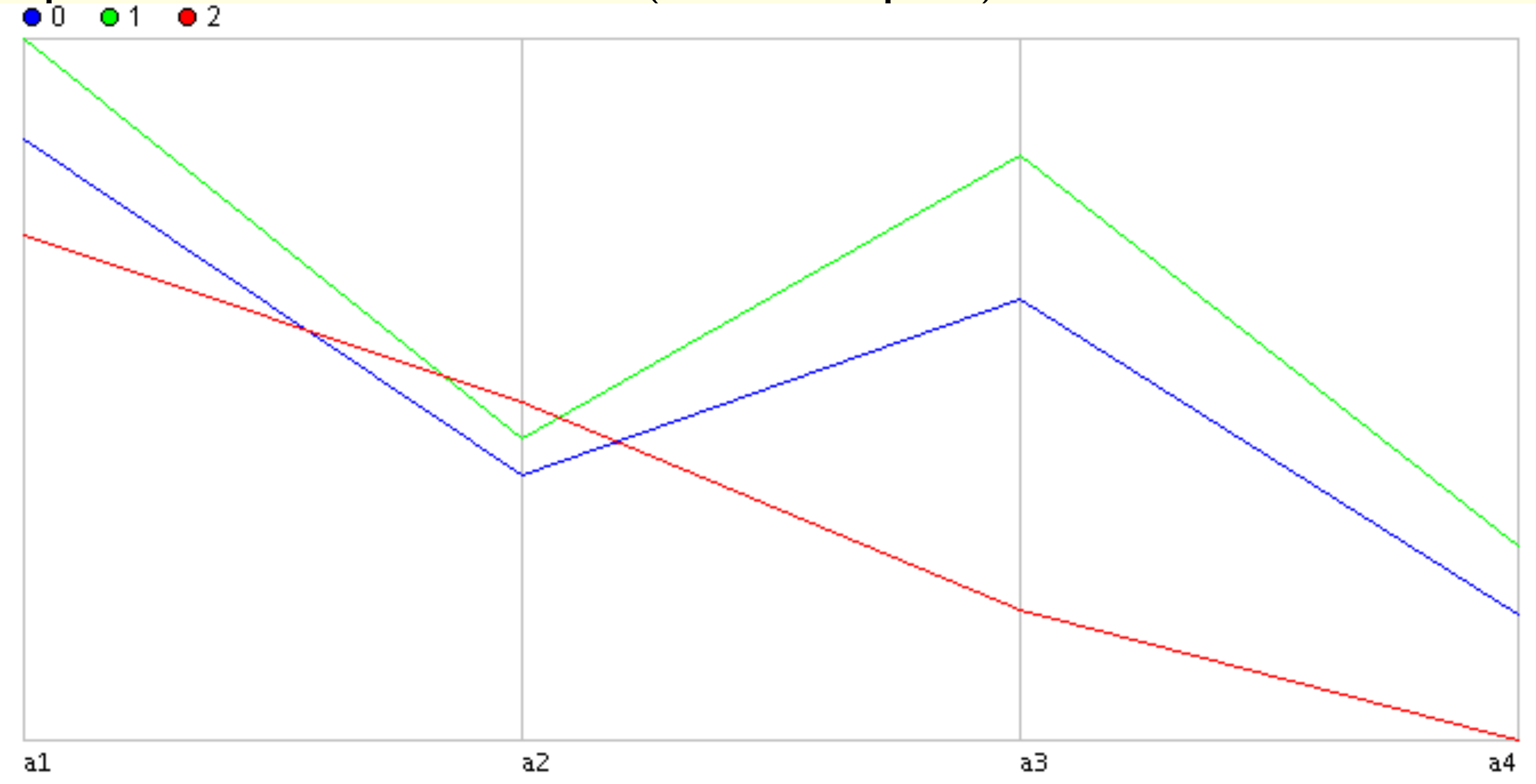
This clusterID can be set as a class label, to allow a classification algorithm, such as a decision tree, to evaluate the pattern governing cluster membership.

The branches of the tree will define ‘cluster membership’, facilitating subjective analysis of each cluster.

Take a look at: ../sample/processes/07_clustering/07_clusterclassification.xml in RapidMiner

Cluster visualisation

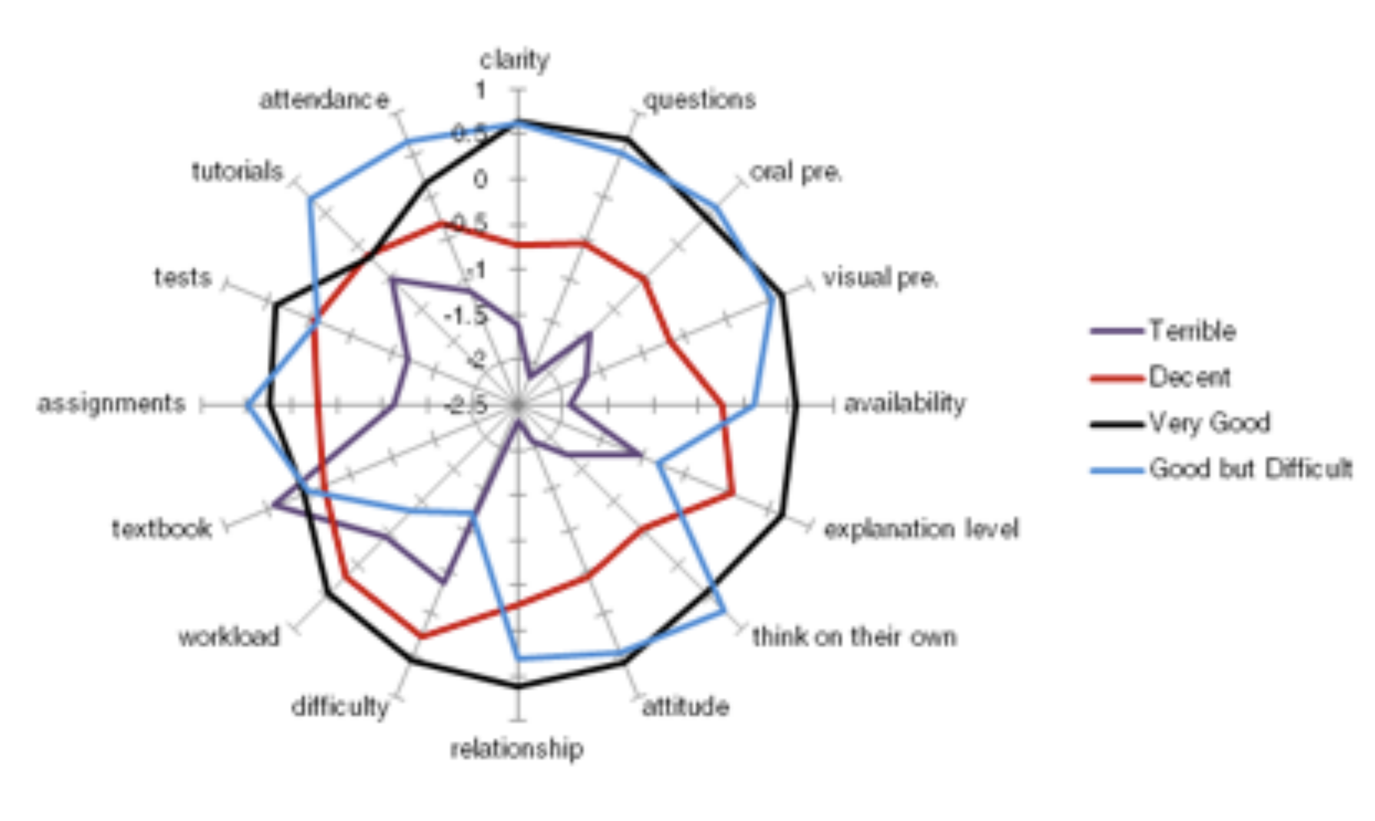
Parallel plot of cluster centroids (centroid plot).



Each line is a different cluster. a1 to a4 are the attributes.

Looking at a1: the centre of cluster 1 has highest values of a1; the centre of cluster 0 has mid-range values in a1; the centre of cluster 2 has lower values for a1. Cluster 2 has the highest values for a2, but significantly lower values for a3 and a4.

Cluster visualisation



Cluster Evaluation using a Radar Plot.

The four clusters are shown in colour.

The cogs represent each attribute in the dataset.

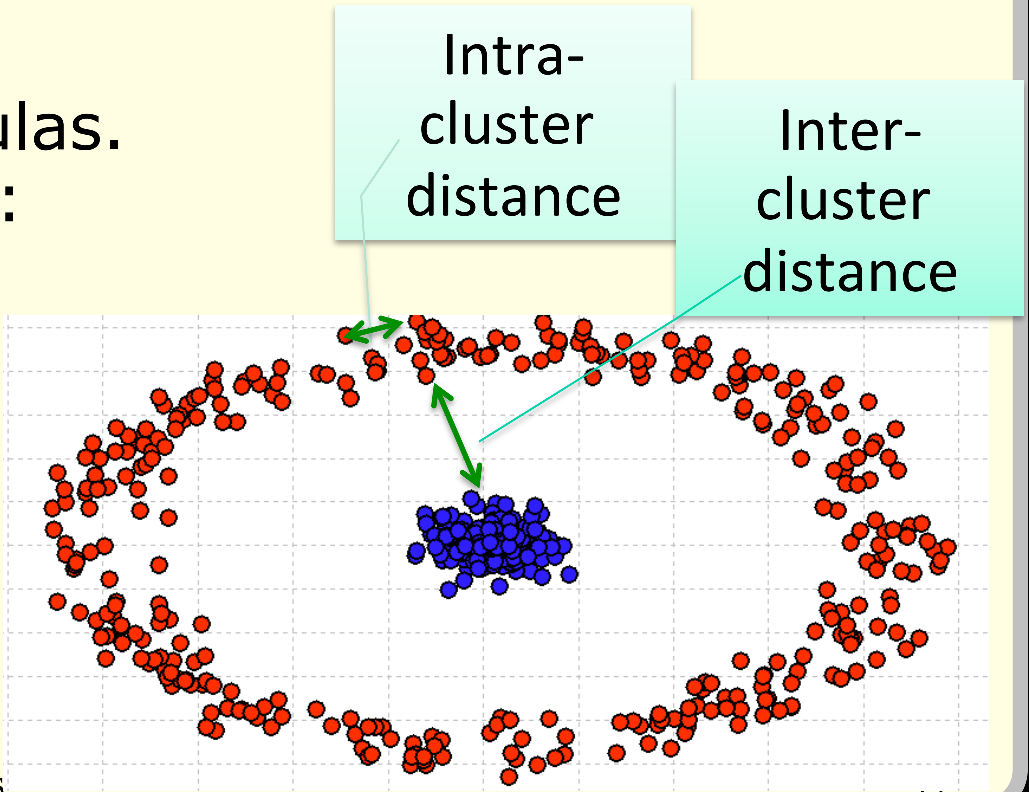
Can see, at a glance, the average value for each attribute in each cluster.

Cluster Evaluation - Objective Measures

Recap: Clustering algorithms attempt to minimise intra-cluster distance, and maximise inter-cluster distances.

Measures to evaluate clusters generally involve calculating a ratio between these two distances.

There are a range of such formulas.
We will look at the simplest one:
– the Dunn index



Intra-cluster distance

Inter-cluster distance

Dunn index

The Dunn index defines the ratio between the smallest inter-cluster distance – $d_{min,}$ and the largest intra-cluster distance - d_{max}, and is defined as follows:

$$D = \frac{d_{\min}}{d_{\max}}$$

i.e. d_{min} is the smallest distance between two objects from DIFFERENT clusters, and d_{max} is the largest distance between two objects from the SAME cluster.

The result will be a number in the interval [0,∞]. The larger the number, the better the cluster definition.

While relatively simple to calculate, the formula is sensitive to noise.

Summary

- Clustering is unsupervised learning, i.e. a class label is NOT used.
- Clustering algorithms attempt to organise rows of data into groups, such that distances between objects in the same group are small, and distances between objects in different groups are large.

- There are a number of categories of clustering algorithms:
  - Partitioning methods for which you need to specify the number of clusters in advance, e.g K-Means.
  - Density based methods, look for dense regions of points, e.g. DBScan.

  Most Algorithms calculate the distance between two rows as a Euclidean Distance.

- Evaluating clusters is difficult, as there is no 'actual cluster ID' to compare the suggested cluster ID with. Evaluation can be:
  - Subjective: Using visualisation tools, or a decision tree to help a domain expert evaluate cluster membership.
  - Objective: using a measure such as the Dunn index