**supervised VS unsupervised learning:**

supervised:

* algorithms are given labelled examples (target class) for the various types of data that need to be learned
* eg, classification algorithms (decision trees, ANN, Bayesian)

unsupervised:

* data is unlabelled (no predefined classes) and learning algorithms attempt to find patterns within the data to put into groups or sets
* eg. clustering algorithms

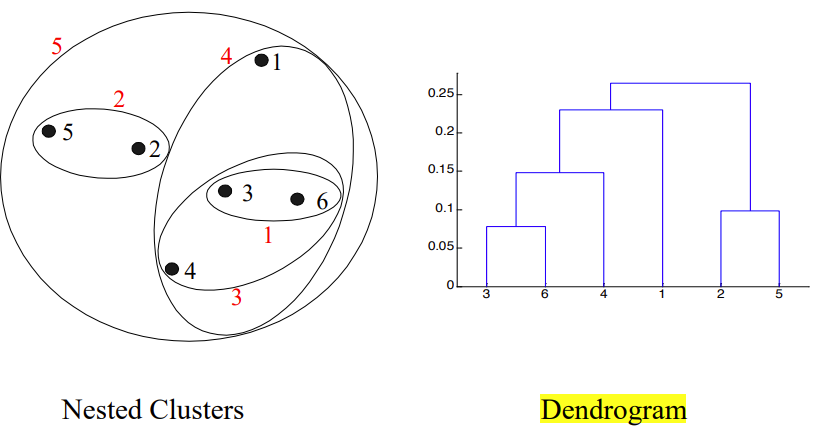
cluster analysis:

finding groups of points such that the points in a group will be similar or related to one another and different from or unrelated to the points in other groups

* minimised intra-cluster distances
* maximised inter-cluster distances

types of clustering:

* partitional: the division of data points into non-overlapping subsets (clusters) such that each data point is in exactly one subset
* hierarchical: a set of nested clusters organised as a hierarchical tree



k-means:

* partitional clustering approach
* each cluster is associated with a centroid (centre point)
* each point is assigned to the cluster with the closest centroid
* K (number of clusters): must be specified
* algorithm:
  + select k points at random as the initial centroids
  + repeat:
    - form k clusters by assigning all points to the closest centroid
    - re-compute the centroid of each cluster
  + until centroids don’t change

evaluate k-means by SSE (sum of squared errors):

* for each point the error is the distance to the nearest cluster
* one easy way to reduce SSE: increase k
* smaller SSE: good clustering

processing:

* pre-processing:
  + normalise data
  + eliminate outliers
* post-processing:
  + eliminate small clusters that may represent outliers
  + split ‘loose’ clusters (high SSE)
  + merge clusters that are ‘close’ and that have relatively low SSE

totss: total sums of squares from a single centroid

withinss: sums of squares within each cluster (variance)

tot.withinss: total within clusters

betweenss: total sums of squares with clusters

between\_ss: variance between clusters

total\_ss: all variance in data

**silhouette:**

how well each data point sits within its cluster

Text

Description automatically generated with low confidence

ai: average distance between that point and all other points in the same cluster

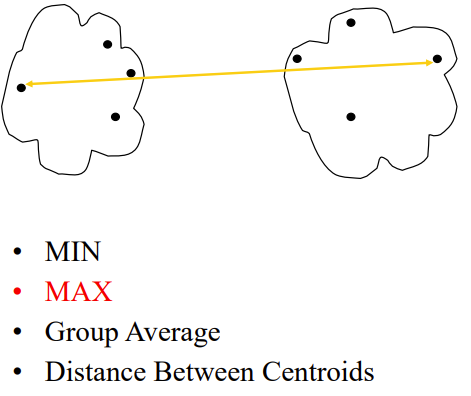
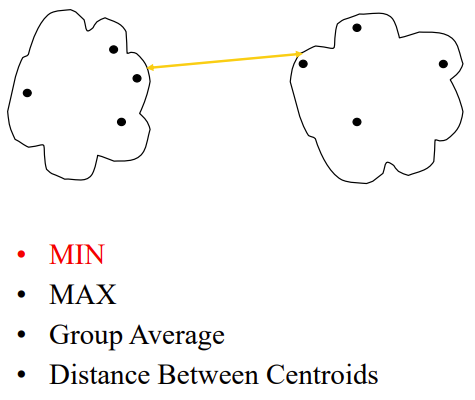
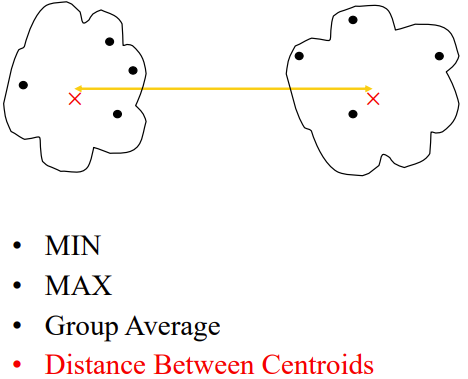
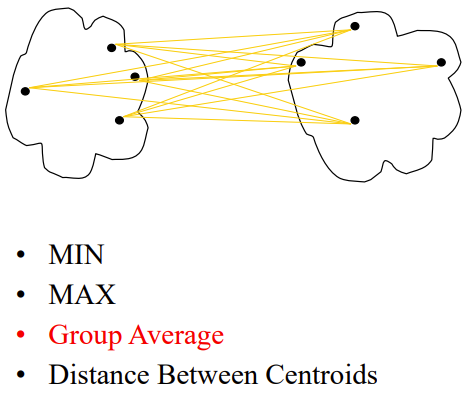
bi: smallest average distance to any cluster it doesn’t belong to

ideally: small ai & large bi

hierarchical clustering:

* create a set of nested clusters organised as a hierarchical tree that
* record the sequences of merges or splits
* can be visualised as a dendrogram
* advantages:
* don’t have to assume any particular number of clusters (any number of clusters can be obtained by ‘cutting’ the dendrogram at the appropriate level)
* may correspond to meaningful taxonomies (eg. plant & animal kingdom)

Inter-Cluster Similarity

x

dendrogram:

* decompose data points into several levels of nested portioning (tree of clusters)
* a clustering of data points is obtained by cutting the dendrogram at the desired level then each connected component forms a cluster
* branch height: the distance between clusters

**2 main types of hierarchical clustering:**

* agglomerative:
  + start with the points as individual clusters
  + at each step, merge the closest pair of clusters until only one 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 similarity or distance matrix and merge or split one cluster at a time

**agglomerative clustering algorithm:**

* more popular hierarchical clustering technique
* distance matrix stores the distances between each cluster
* algorithm:
  + compute the distance matrix
  + let each data point be a cluster
  + repeat:
    - merge 2 closest clusters
    - update the distance matrix
  + until only a single cluster remains
* define inter-cluster similarity:
* min:
  + can handle non-elliptical shapes
  + sensitive to noise & outliers
* max:
  + less susceptible to noise & outliers
  + tend to break large clusters biased towards elliptical shapes
* group average:
  + compromise between single & complete link
  + less susceptible to noise & outliers
  + biased towards globular clusters
* distance between centroids