Clustering

Statement for Audio and Video Learning Resources

Video and audio content at the University uses closed captions generated by automatic speech recognition (ASR). The ASR process is based on machine learning algorithms which automatically transcribe voice to text. According to our technology providers, this process is approximately 70-90% accurate depending on the quality of the audio, and consequently video and audio closed captions may include some transcription errors. It is therefore important to recognise that the original recording is the most accurate reflection of the content, and not the captions.

If you require accurate captions as part of your reasonable adjustments, please contact the Inclusion Centre to discuss your requirements.

Lab Postmortem

Something to remember/practice:

Practice “knitting” your .Rmd files occasionally (Publish or ctrl+shift+k will knit and output an html file).

Notice where the .Rmd files are saved and how you can find both that and your knitted html file.

Notice how long code takes to run.

When you create code, add notes about the results.

Content

- Introduction
- *Distance metrics*
- K-Means Clustering algorithm
    - *Centroid creation*
- Hierarchical (agglomerative) clustering
- Discussion.

Cluster learning

Unsupervised

- No target value to be predicted – no class or number to be predicted.
- Instances divided into natural groups - clusters

Different types of algorithm

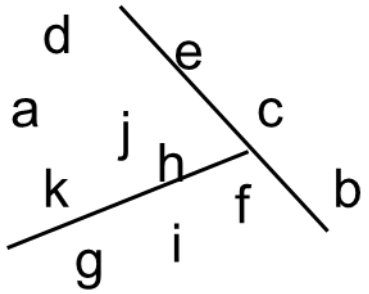
- Incremental learning
  - Instances are dealt with one at a time
- Batch learning
  - Instances are dealt with all at once

Different types of output

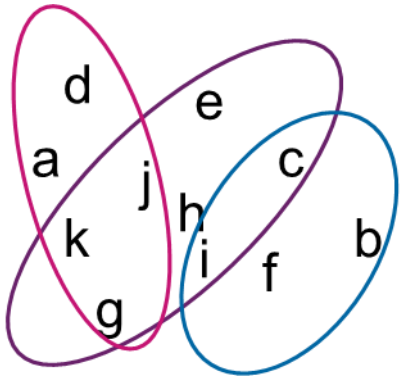
- …..examples follow

Different types of cluster

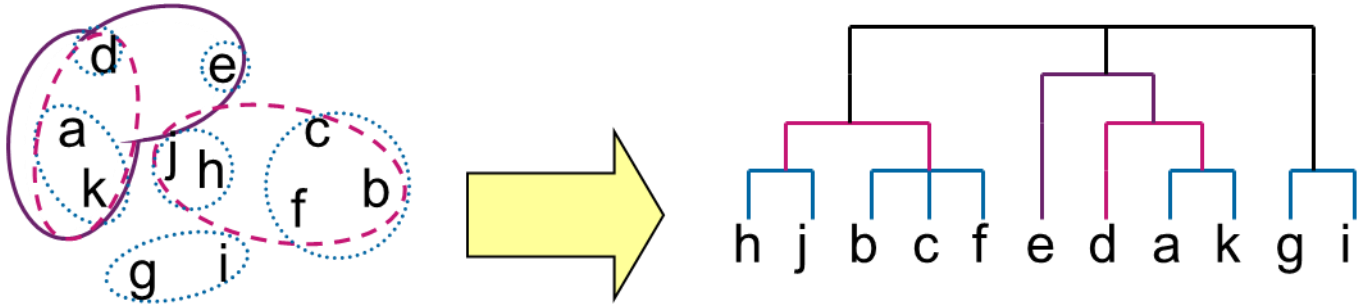
Exclusive clusters



Overlapping clusters

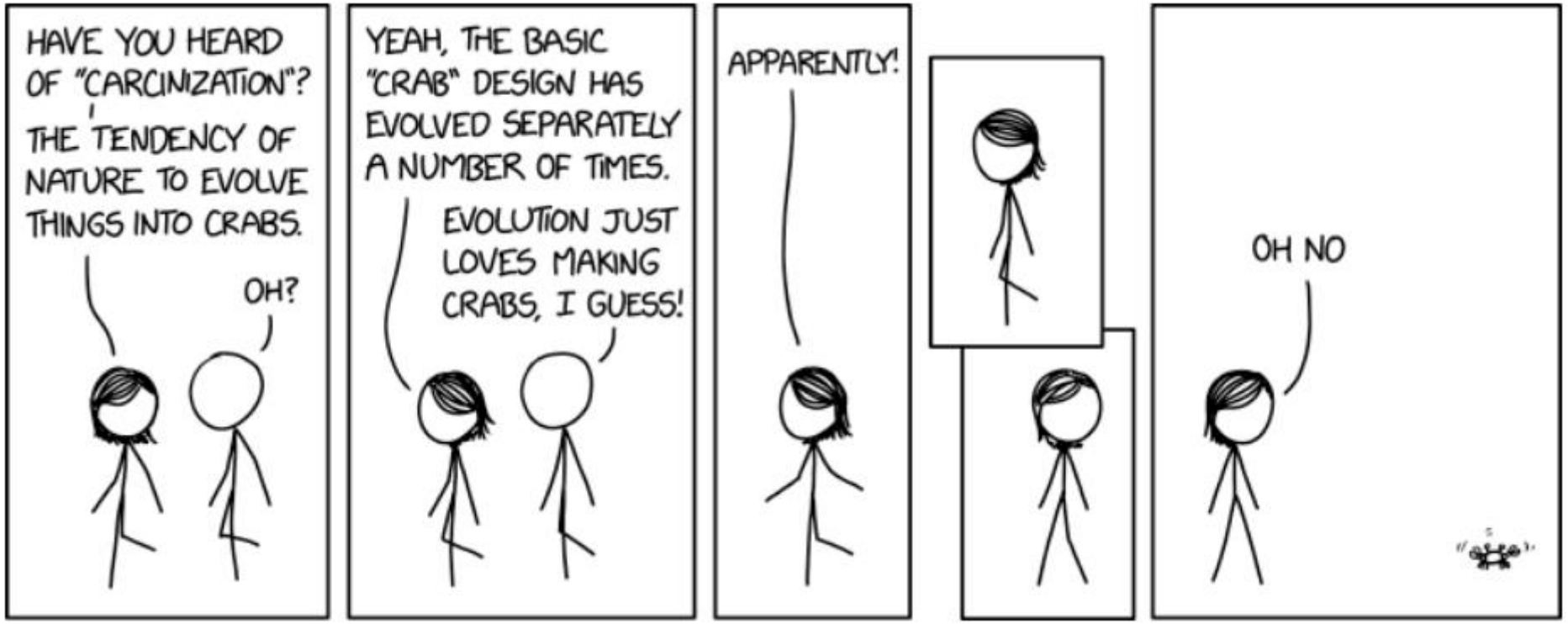


Hierarchical clusters



Dendrogram Dendron is Greek for tree

Hierarchical clusters are still a research area



How to design hierarchies such that similar “species” are on the same branch is difficult.

Terminology

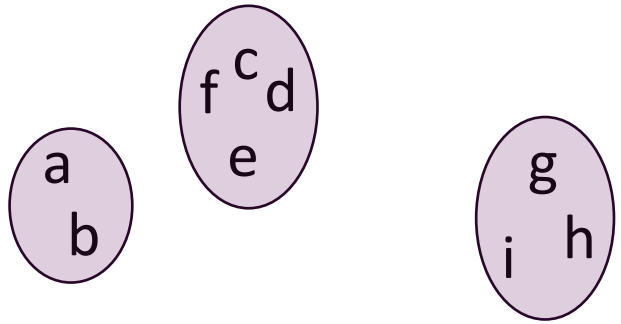
Cluster: a group of instances (objects or observations).

Cluster members: the objects (instances) in a cluster

Centroid: the centre of a cluster

Target

- For each cluster, its members are similar to each other
  - **High intra-cluster similarity**
- Instances not belonging to the same cluster are different.
  - **Low inter-cluster similarity**



Clustering in action

"Soft biometrics" of world leaders taken from videos.

Clusters form naturally and can easily be seen.

Deep fake data does not **cluster** with authentic data.

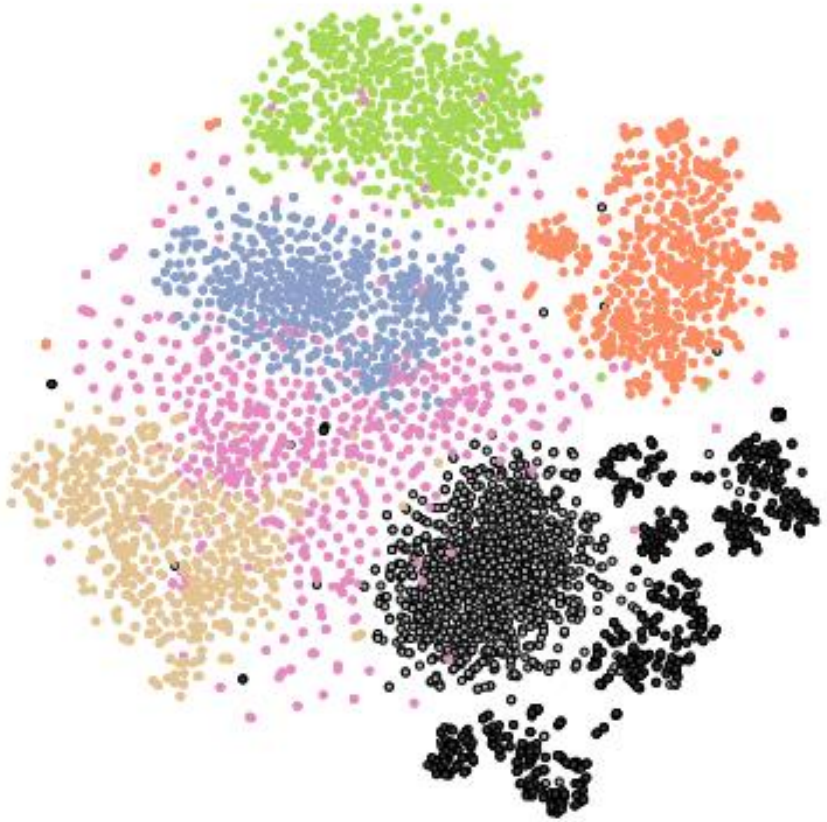
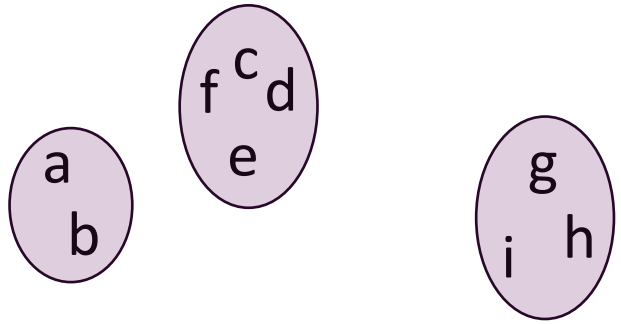


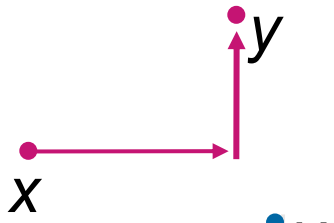
Figure 3. Shown is a 2-D visualization of the 190-D features for Hillary Clinton (brown), Barack Obama (light gray with a black border), Bernie Sanders (green), Donald Trump (orange), Elizabeth Warren (blue), random people [23] (pink), and lip-sync deep fake of Barack Obama (dark gray with a black border).

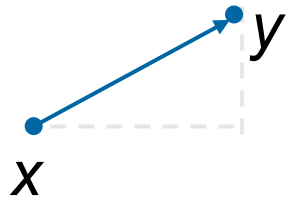
From "Protecting World Leaders Against Deep Fakes" Agarwal and Farid, CVPR, 2018

Similarity / Distance metric

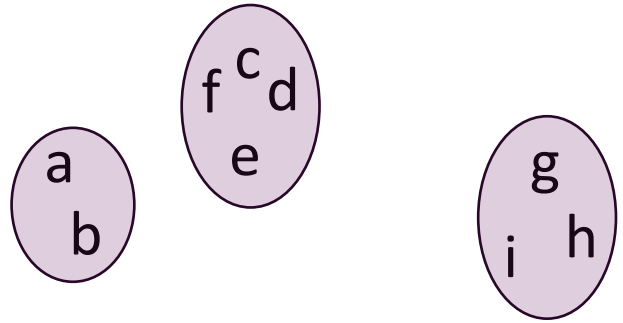
- To cluster similar instances we need a measure of similarity
- Often, we have an idea of distance
  - Low distance = high similarity
- Can construct similarity/distance measures in similar way to kNN lecture
  - Manhattan distance $\sum_i |x_i - y_i|$
  - Euclidean distance $\sqrt{\sum_i (x_i - y_i)^2}$





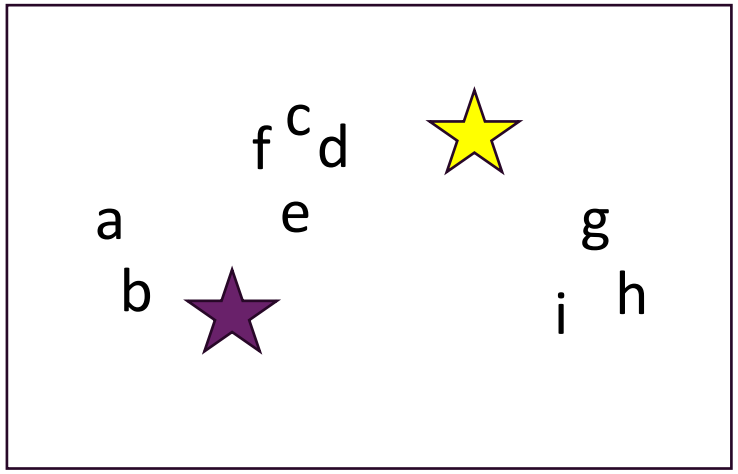


k-means Clustering



- Batch clustering
  - Iterates over all instances until convergence
- Typically forms clusters in numeric domains
  - Uses a distance metric
- Partitions instances into **disjoint clusters**

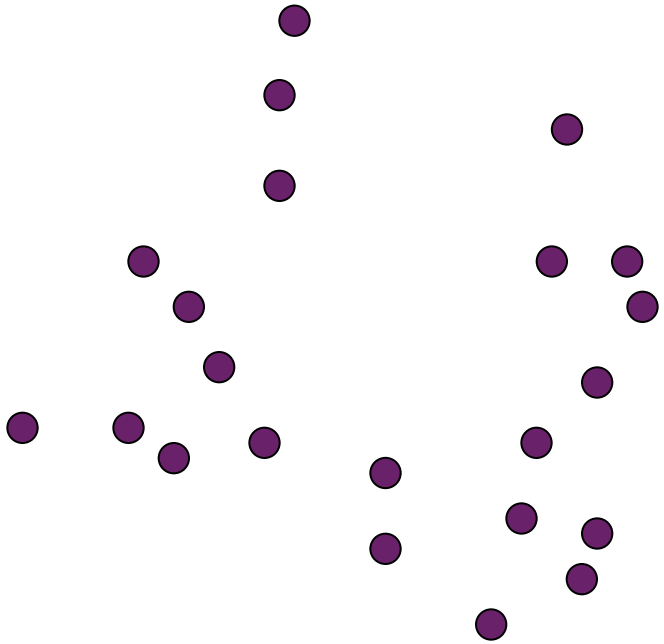
k-means Algorithm



- Groups the instances into k clusters
- Step 1: choose cluster centres (centroids). Options:
  - k instances at random
  - Randomly assign instances to clusters.
- Step 2: assign instances to clusters based on distance to cluster centroids
- Step 3: compute centroids of clusters
  - Average attribute values of cluster members
  - What if attributes are nominal? Use most frequent value?
- Repeat steps 2-3 with centroids as new centres until clusters do not change

k-means Process Example

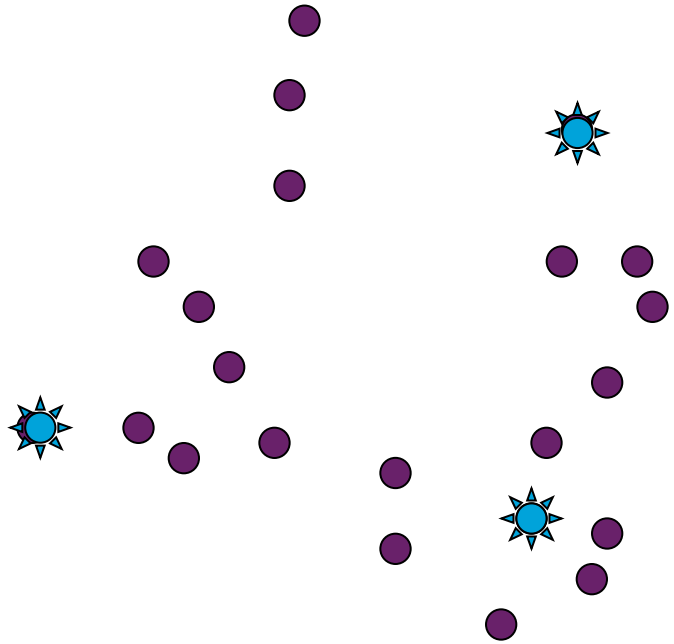
Assume k = 3



Watch the StatQuest for 1-dimensional clustering.

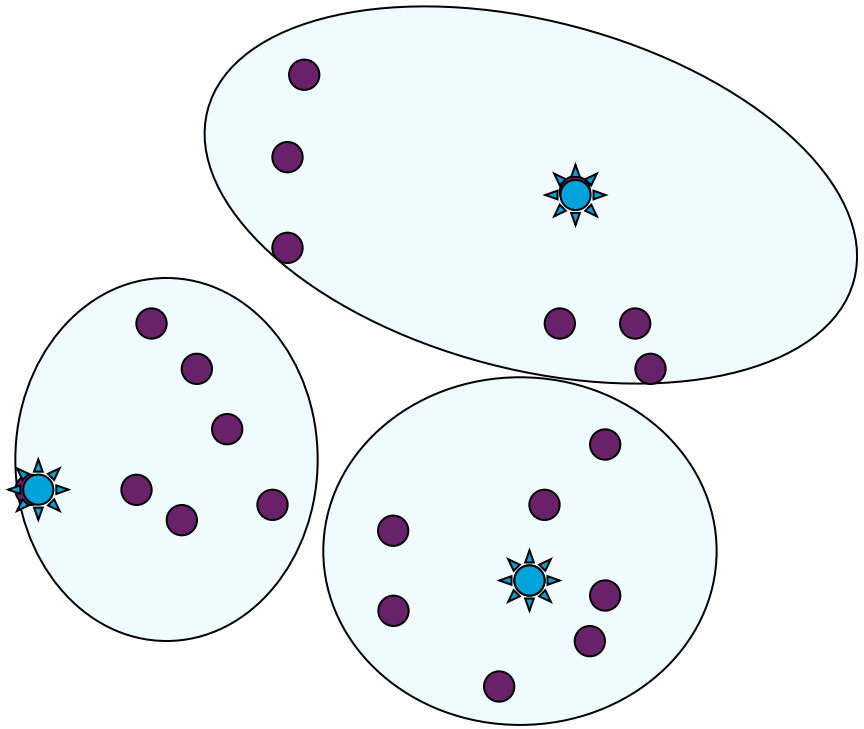
k-means Process Example (2)

Randomly select instances to be centroids (or randomly select centroids)



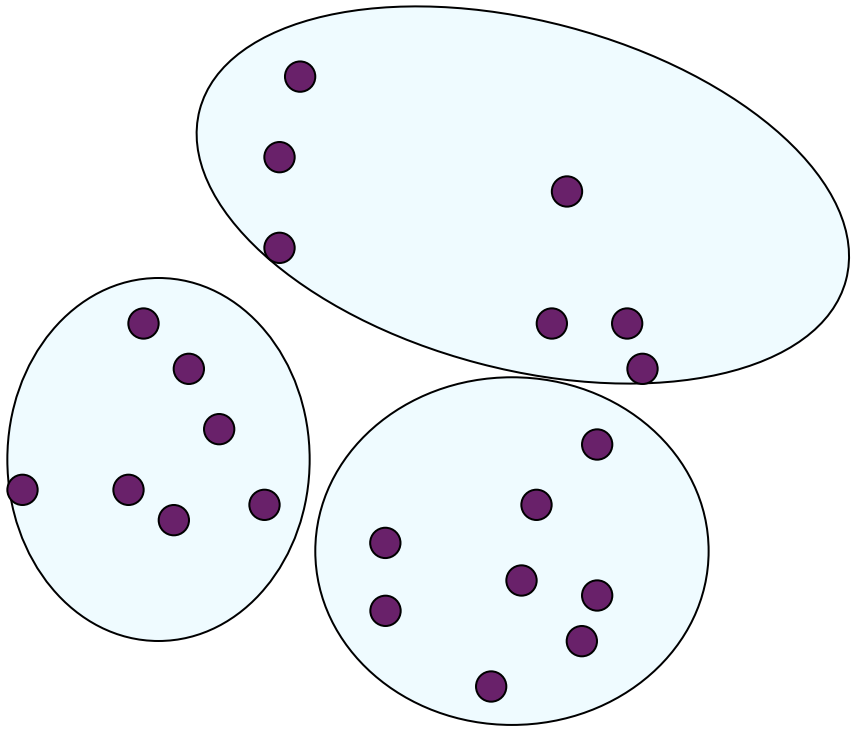
k-means Process Example (3)

Assign instances to the nearest cluster (centroid).



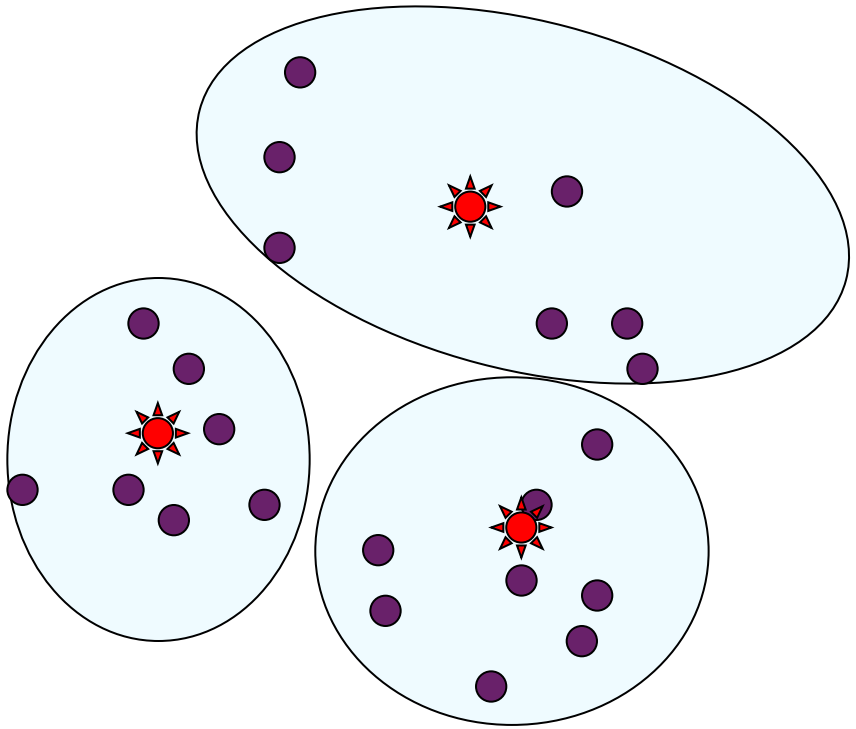
k-means Process Example (4)

Re-calculate new centroids (mean of all instances in the cluster).



k-means Process Example (5)

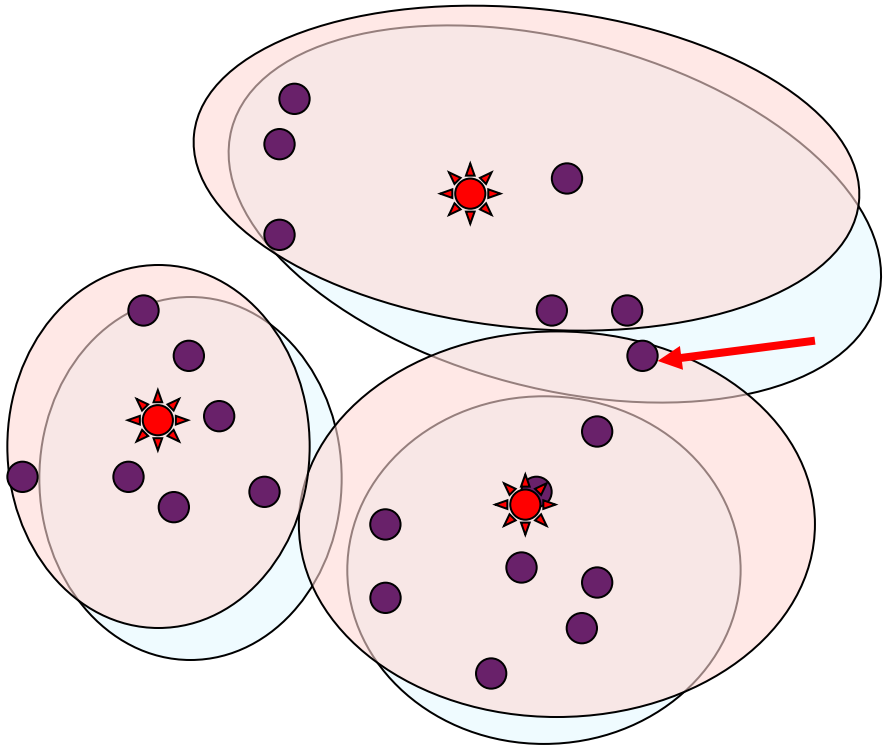
New centroids (they're *probably* not actual instances now).



k-means Process Example (6)

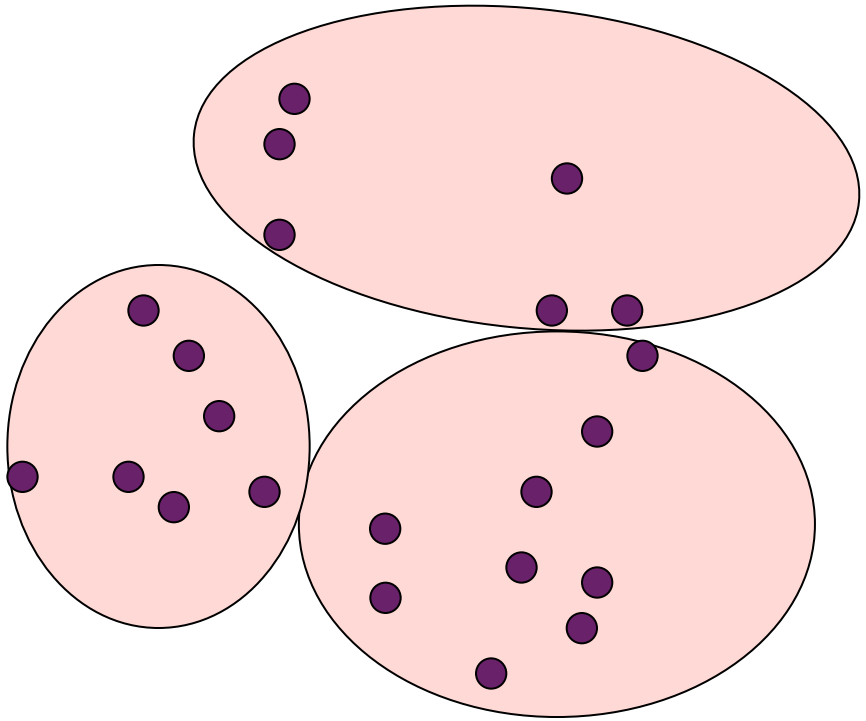
Assign instances to clusters according to distance to centroids.

One instance changes cluster



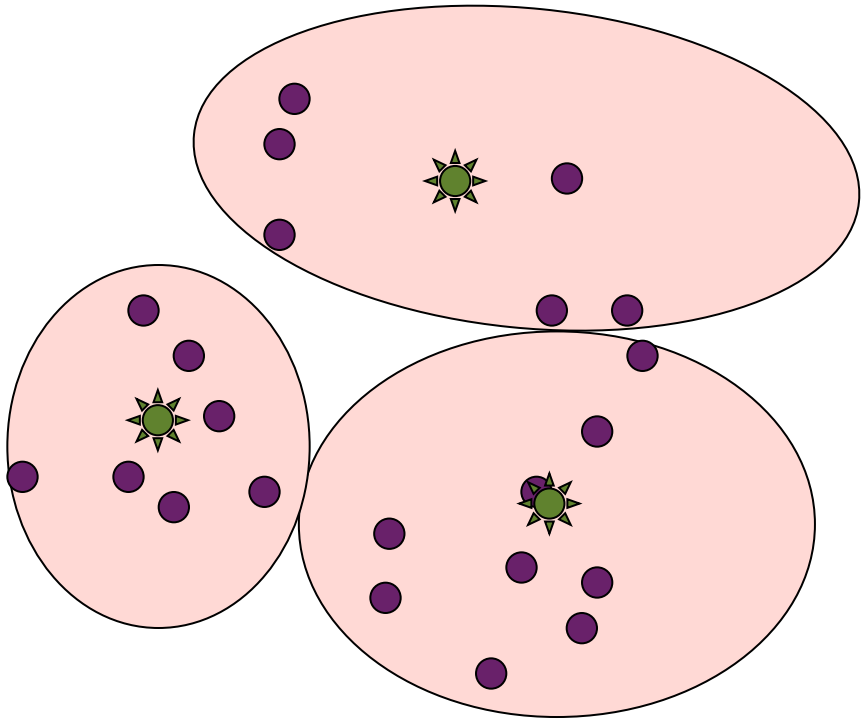
k-means Process Example (7)

New clusters formed



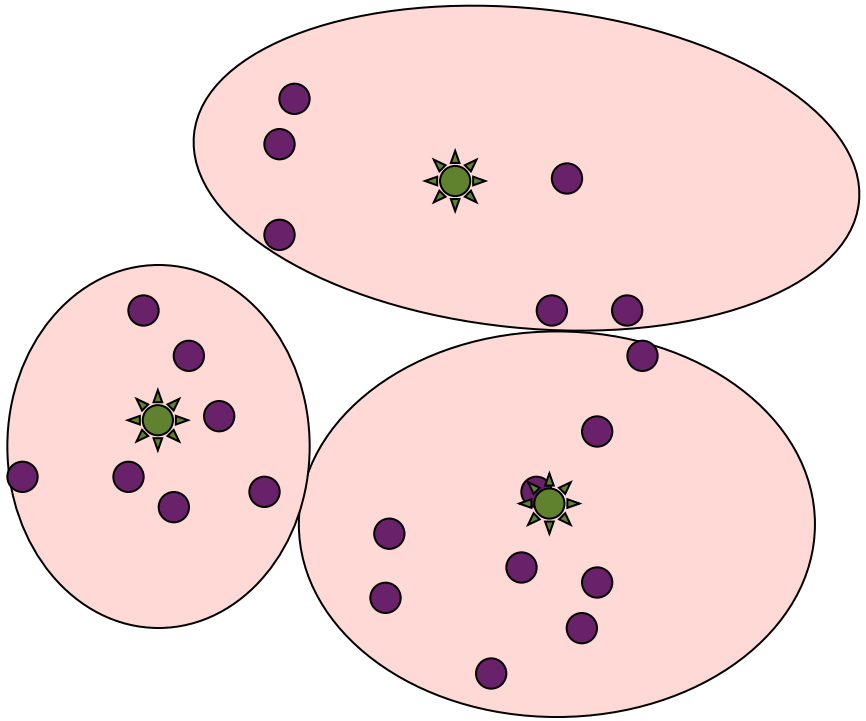
k-means Process Example (8)

Calculate new centroids



k-means Process Example (9)

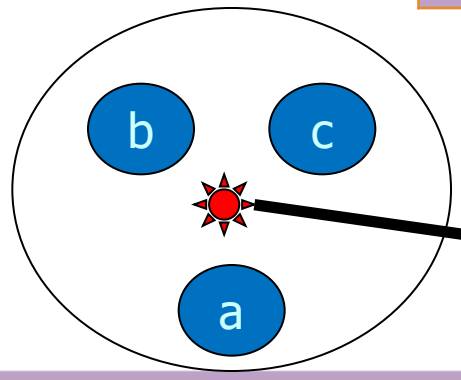
Assign instances to clusters (nearest centroid). No change in cluster members so stop.



Forming a cluster centroid

- Centre of the cluster
  - An instance with average values

Instances	age	salary	insurance
A	25	40K	Yes
B	22	35K	No
C	26	38K	Yes
centroid	AVG(25,22,26)	AVG(40, 35, 38)	MajorityVote(Yes, Yes, No)



Cluster

Age=24.3
Salary= 37.7
Insurance = Yes

Note that in reality numeric attributes should have been normalised or standardised

Distance - Numeric Attributes

- E.g. *C1 & C2* are centroids and *x* is an instance

	age	salary
C1	55	55k
C2	50	60k
X	35	35k

Note that in reality numeric attributes Age and Salary should have been normalised or standardised

$$dist(C1, X) = \sqrt{20^2 + 20^2} = 28.28$$
$$dist(C2, X) = \sqrt{15^2 + 25^2} = 29.15$$

X is closest to C1 so assign it to cluster 1

Normalisation or standardisation

- Attributes are measured on different scales.
  - Attributes measured on larger scales have higher impact.
  - Need to
    - Normalise: transform to scale [0..1]

      or
    - Standardise (centre and scale): transform to mean of zero and standard dev. of 1.

(covered in CMM535?)

Aside: k-means as a mini-max algorithm

Minimise intra-cluster distance

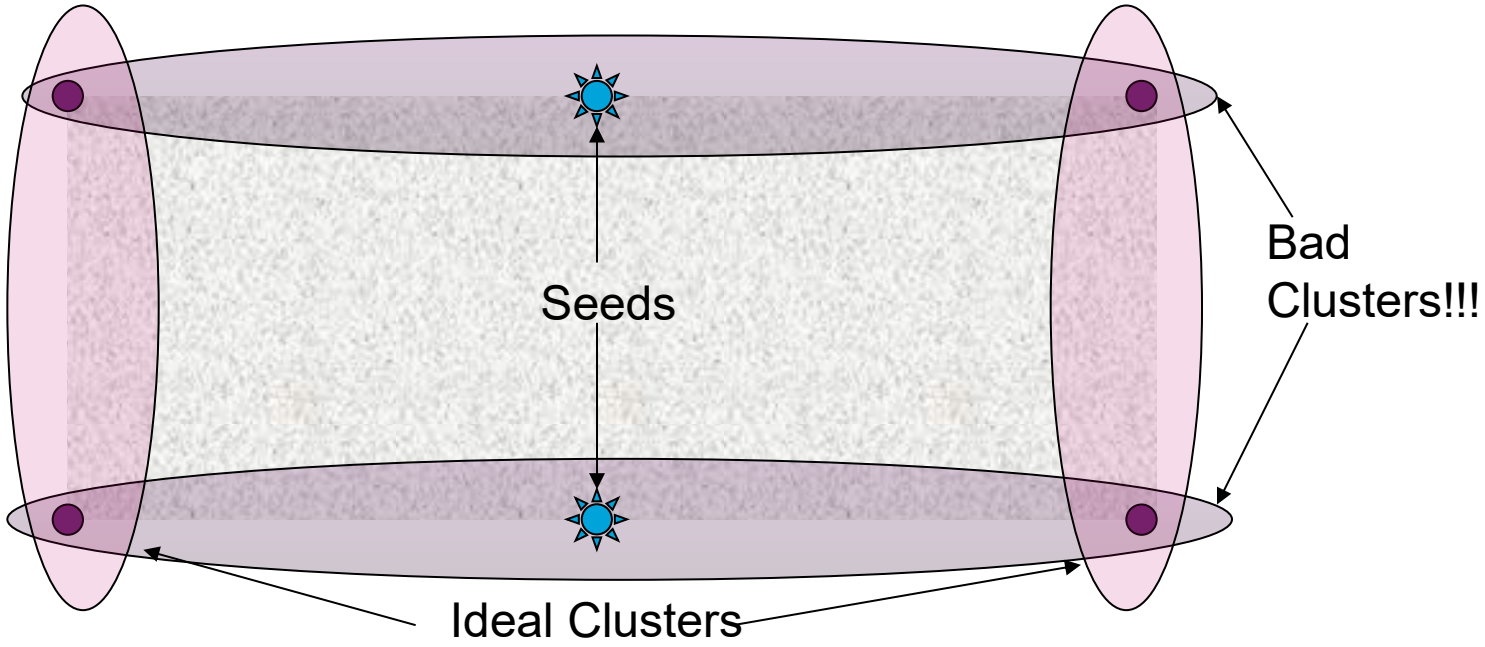
Maximise inter-cluster distance

Some algorithms (e.g. matching networks) actually create a latent space or mapping of attributes that specifically does this for (supervised) classification.

k-means Discussion

- Results vary based on initial choice of centroids
- But we can run it several times with different initial centroids, evaluate the results and choose the best clusters.
- Algorithm can get trapped in a local minimum
  - Example: 4 instances at rectangle vertices
    - Local minimum: cluster centres at long edge midpoints
  - Simple way to increase chance of global optimum
    - restart with different random centroids

k-means - local minimum

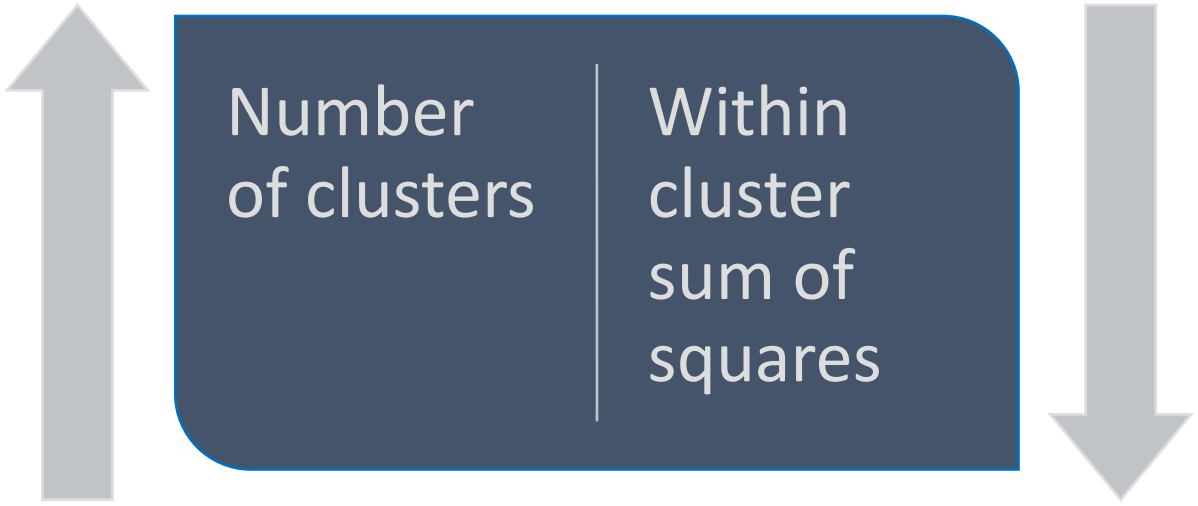


How many clusters?

- If **desired** number is known, use that number
- Otherwise **try** several values for k and **select** the k which gives **best** results.
- How is k for “best results” selected?
  - **Elbow** method – within clusters sum of squares
    - Uses sum of squared error for each cluster
  - **Silhouette** method – high cohesion and separation
    - **Cohesion** – the degree to which an instance is similar to others in its cluster
    - **Separation** – the degree to which an instance is not similar to instances in other clusters.
  - **Jump** method – choose the value of k which causes the biggest jump in distortion.
    - Distortion is a measure of within cluster dispersion

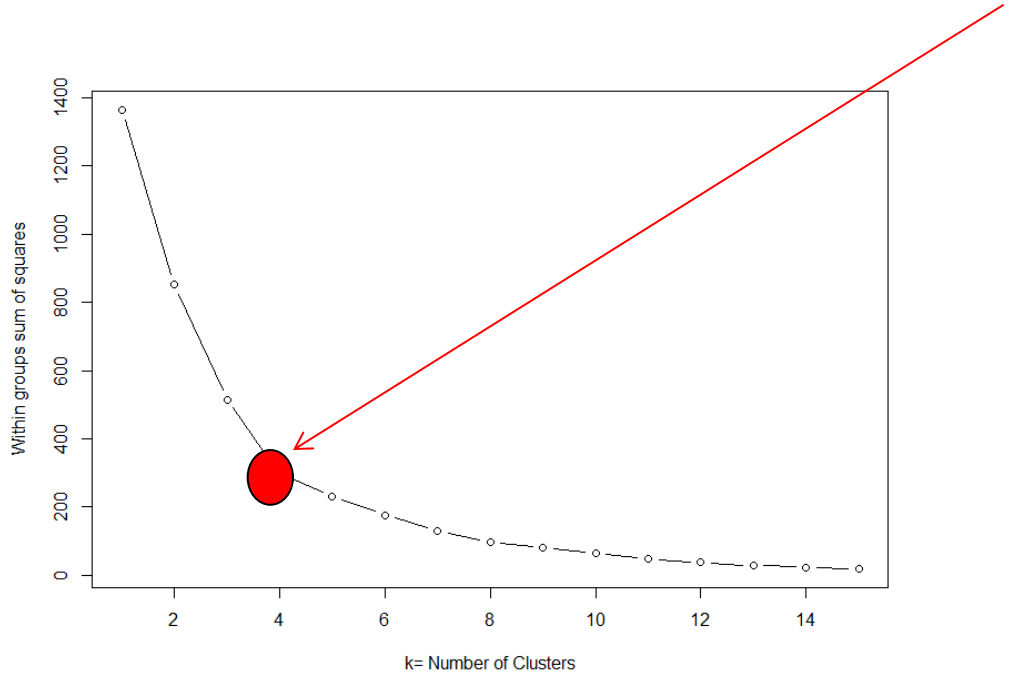
How many clusters? Elbow method

- Measure of variability of instances within each cluster
- Based on the square of the distance of each instance to its cluster centroid.



How many clusters? Elbow method (cont)

- For each k value
  - Cluster the data
  - Calculate the ***within cluster sum of squares***
- Plot WCSS for each k value
- Pick the k value at the 'elbow' in the plot where
  - X axis – number of clusters
  - Y axis – within cluster sum of squares



Within cluster sum of squares

$$\mathrm{WC} = \sum_{k=1}^{K} \sum_{i \in Sk} \sum_{j=1}^{P} (x_{ij} - \bar{x}_{kj})^2$$

For each attribute in instance

Calculate square error ("distance") to centroid

For each cluster

For each instance within cluster

Where

- K is the number of clusters
- S_k is the set of instances in the kth cluster
- $\bar{x}_{kj}$ is the centroid for j th attribute in cluster k
- x_{ij} is the value for the j th attribute of instance i in cluster k

- i.e. the sum of the squared errors (distances to centroids).

How many clusters? Silhouette

- For each k value
  - Cluster the data.
  - Calculate the silhouette value for each instance.
  - Silhouette value for instance: number between -1 and 1
    - Low or negative value: instance is not similar to other instances in the cluster.
    - High value: instance is similar to other instances in its cluster and dis-similar from instances in neighbouring clusters.
  - If most instances in most clusters have a high silhouette value, the clustering is appropriate.
    - Otherwise – a different k value may be appropriate
  - Calculate the average silhouette value.
- Choose the k value which gives the highest average silhouette.

How many clusters? Jump method

- Create distortion curve by running k-means with all values of k and calculating distortion.
  - Run k-means for each k value
  - For each k, calculate *distortion* – a measure associated with the dispersion of instances in a cluster.
  - Select transformation power *y* so negative power of *y* is applied to each distortion.
    - Typically y=p/2 where p is the number of attributes
  - Calculate the jumps. Look at the change in transformed distortion (the "jump") between consecutive distortion values for k.
- Choose k which maximises the jump.
- We will not see the jump method in the labs.

Advantages

- K-Means is easy to implement
- Reasonably efficient
- Easy to explain
- Non-deterministic (also a disadvantage)
  - Different start centroids may produce different clusters
  - Can be repeated with different seeds and “best” results can be used.

Disadvantages

- The value of k must be specified – tricky. Use elbow, silhouette or jump method
  - But this equates to running-means with different k values
- The choice of initial k centroids is key: results vary according to choice – non deterministic
- May not work well with noisy data
- Applicable only when **averaging is meaningful** to the given data set – mean is calculated to calculate centroids

Solutions to weaknesses

- Use **elbow**, **silhouette**, or jump method for k determination
  - Run several times with different k values
- Repeat k-means with different initial centroids; select result with highest quality
  - i.e. run several times with same k but different seeds.
- Use hierarchical clustering to locate the centres (see below)
- Find centres that are not close to each other
- K-medoid: use the nearest instance to cluster centre as centroid if mean cannot be defined (i.e. when attributes are nominal or discrete) .

Summary – k-means

- Group similar cases on their attributes
- **K-means** is a **batch clustering** approach
- Quality of clusters strongly related to
  - Selecting the value for k
  - Initial centroids
- **Hierarchical k-means**
  - Apply k-means to the dataset
    - Apply hierarchical k-means to each resulting cluster
- K-means can be computationally costly due to pair-wise distance calculations
- Binary k means – repeatedly apply 2-means
  - Results in a binary tree of clusters

Notes

- Centroids
  - Start centroids are often called initial seeds
    - Instances (or averages of instances) selected to be at the centre of the initial clusters
  - NOT the “seed” that you set in R for reproducibility
- K-means vs kNN
  - k-means is an unsupervised clustering
  - kNN is a supervised classification algorithm.
  - They both use distance between instances (Euclidean, Manhattan).
  - For k-means, k is the number of clusters
  - For kNN, k is the number of neighbours

Content

- Introduction
- *Distance metrics*
- K-Means Clustering algorithm
    - *Centroid creation*
- Hierarchical (agglomerative) clustering
- Discussion

Agglomerative clustering

- Each instance is an individual cluster
- Build a n x n distance matrix.
  - distance between any pair of instances (clusters).
- While the number of clusters > 1 do:
  - Find a pair of clusters with the minimum distance
  - Join the two clusters - merge
  - Replace the entries in the matrix for the 2 original clusters by the new cluster
    - Re-calculate distances to other clusters and update the matrix

Example:

- Assume the following matrix with distances

	A	B	C	D
A		-	-	-
B	2.3		-	-
C	3.6	4		-
D	2.7	**0.8**	1.2	

Cluster instances with shortest distance, B and D

- Merge B and D as they have the shortest distances. Create cluster {B,D}
- Compute the distance between {B,D} and A and C

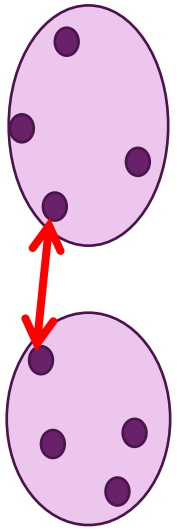
	A	{B,D}	C
A		-	-
{B, D}	2.55		-
C	3.6	2.6	

- Repeat with smaller matrix

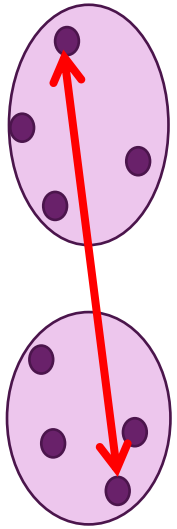
Agglomerative clustering schemes

- What distance is used as inter-cluster distance?
- There are different schemes
  - Single link: the distance between two closest points
  - Complete link: the distance between two farthest points
  - Group average: the average of all pair-wise distances
  - Centroids: the distance between the centroids

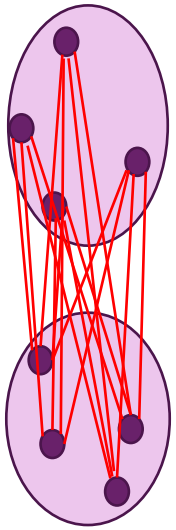
Clustering types/schemes



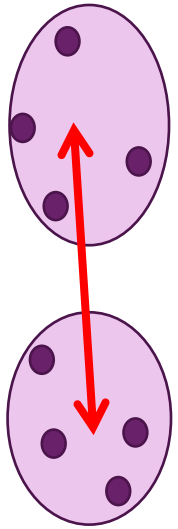
Single link



Complete link



Group average



Centroids

Advantages of agglomerative clustering

- Advantages
  - Deterministic results – not dependent on initial centroids
  - K not specified
  - Clusters of arbitrary shapes can be created (single-link)
- Disadvantages
  - Very slow for large data sets
- Cannot undo membership like the K-means

Evaluating cluster quality

- Good clustering if
  - High-level similarity within each cluster – high **cohesion**
    - Low **w**ithin **c**luster sum of squares (WC)
  - Large distance between clusters – high **separation**
    - Sum of distances **b**etween **c**lusters (BC) is high
  - Combination of cohesion and separation
    - **BC/WC** - good indicator of overall quality.

Agglomerative clustering – how many clusters?

- Start at the root –
- Repeat
  - Move one level down at a time
  - At a level, evaluate the overall quality of clusters (e.g. within clusters sum of squares
  - If the quality is not acceptable , move to the next level down - repeat

  Until quality is acceptable
- Take the clusters at the last level as the final result.

Discussion

- Evaluation of clustering
  - Often by inspection - do the clusters “make sense”?
  - Clusters can be visualised if low dimensionality
  - “Classes to clusters” – known class values vs clusters.
    - Need a class attribute (used only for evaluation) to check if cluster members share the class value
- Interpretation of clusters
  - Supervised learning in a post-processing step
    - Training data is clustered data labelled by cluster id
    - Decision tree or rule-set inferred to predict cluster

… discussion

- Clusters can be used to fill in missing data values
  - E.g. average value within cluster
- Clustering can improve classification
  - Cluster and add new attribute with cluster label to dataset
- Assumed that attributes are independent
  - pre-processing step to make more independent
    - i.e. using *principal component* analysis

Applications

- Recommendation systems
  - Similar products in same cluster
  - Recommend products in cluster of current product
- Social network analysis
- Search result grouping
- Medical imaging
- Image segmentation
- Anomaly detection

Summary

- Clustering: dividing instances into groups
- Methods:
  - K-means – needs k set in advance
- Agglomerative clustering
  - 4 ways of assigning instances to clusters
  - Produces a dendogram which can be `cut' at a level which gives the required quality
    - E.g BC/WC [between cluster sum of squares over within cluster sum of squares]