

Cluster Analysis

Craig Alexander

craig.alexander.2@glasgow.ac.uk

Room 325, Mathematics and Statistics building

5 April 2023



University
of Glasgow | School of Mathematics
& Statistics

Cluster Analysis

- ▶ Interested in grouping objects together
- ▶ Finding new, unknown, group structure based on multivariate data
- ▶ From the day-to-day...



- To the much more challenging...



Basic Idea of Cluster Analysis

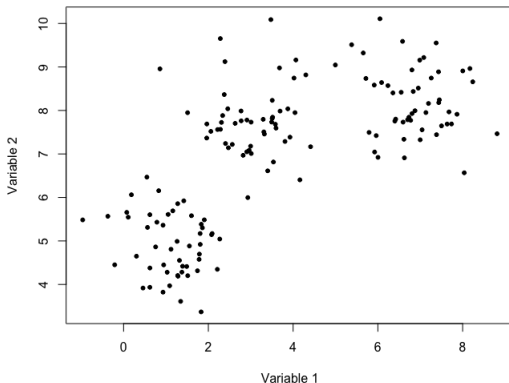
- ▶ Given multivariate data are available on a number of objects of **unknown** group or type or class, want to find groups in the data
- ▶ Our estimates of the groups are called clusters
- ▶ Another latent variable method: discovering categorical latent variable that represents manifest/observed variables

Guiding Principles

- ▶ Objects that are very dissimilar should belong to different clusters.
- ▶ Conversely, objects that are very similar should be in the same cluster.
- ▶ Overall we want the dissimilarities between clusters to be much larger than the dissimilarities within clusters.
- ▶ Once we estimate the groups with clusters, we can partition the data space into areas belonging to each cluster.

Caveats

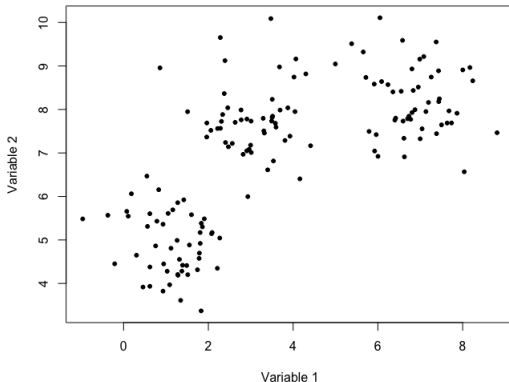
- Clustering is defined by many decisions but also by the inputs



- How many clusters do you see?

Caveats

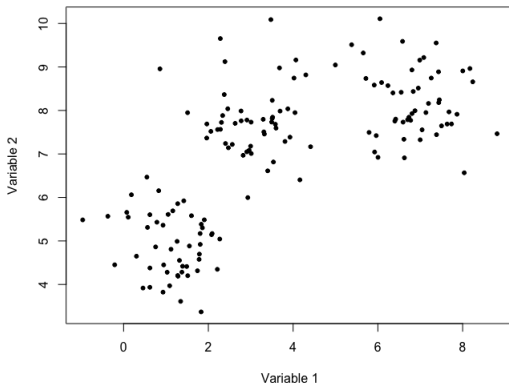
- ▶ Clustering is defined by many decisions but also by the inputs



- ▶ How many clusters do you see? 3
- ▶ If just looking at variable 2, how many clusters?

Caveats

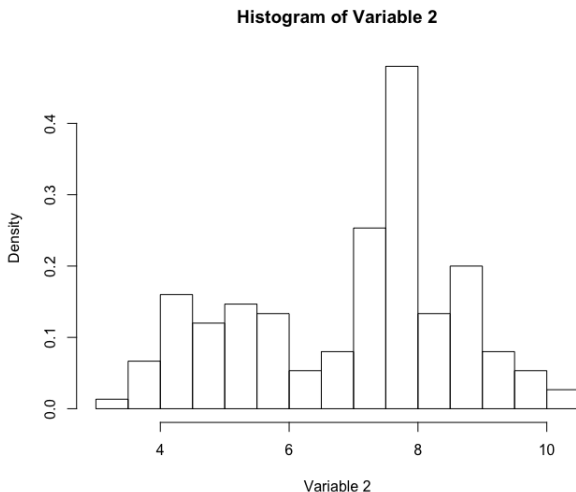
- Clustering is defined by many decisions but also by the inputs



- How many clusters do you see? 3
- If just looking at variable 2, how many clusters? 2

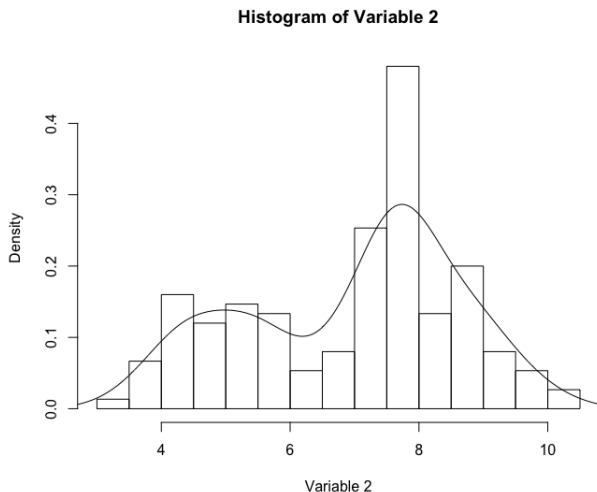
Caveats

- Clustering is defined by many decisions but also by the inputs



Caveats

- Clustering is defined by many decisions but also by the inputs



Dissimilarity?

- ▶ What exactly do we mean by similarity/dissimilarity?
- ▶ Opinions differ and resulting methods do too.
- ▶ Natural way of measuring dissimilarity: distance.
- ▶ Measure distance between all pairs of n points in the data.
- ▶ Result: $n \times n$ square, symmetric distance matrix.
- ▶ Symmetric: only need to report lower triangle.

“Visual Clustering”, what do you see?

	1	2	3	4	5	6
1	0.0					
2	3.0	0.0				
3	2.7	2.8	0.0			
4	9.0	10.0	11.0	0.0		
5	10.0	11.0	12.0	2.9	0.0	
6	11.0	10.0	12.0	2.8	2.5	0.0

	1	2	3	4	5	6
1	0.0					
2	3.0	0.0				
3	2.7	2.8	0.0			
4	9.0	10.0	11.0	0.0		
5	10.0	11.0	12.0	2.9	0.0	
6	11.0	10.0	12.0	2.8	2.5	0.0

	1	2	3	4	5	6
1	0.0					
2	3.0	0.0				
3	2.7	2.8	0.0			
4	9.0	10.0	11.0	0.0		
5	10.0	11.0	12.0	2.9	0.0	
6	11.0	10.0	12.0	2.8	2.5	0.0

► Cluster 1: Obs. 1, 2, 3

► Cluster 2: Obs. 4, 5, 6

What do you see?

	1	2	3	4	5	6
1	0.0					
2	10.0	0.0				
3	2.7	11.0	0.0			
4	12.0	2.9	11.0	0.0		
5	2.6	11.0	2.5	12.0	0.0	
6	11.0	2.8	12.0	2.8	11.0	0.0

	1	2	3	4	5	6
1	0.0					
2	10.0	0.0				
3	2.7	11.0	0.0			
4	12.0	2.9	11.0	0.0		
5	2.6	11.0	2.5	12.0	0.0	
6	11.0	2.8	12.0	2.8	11.0	0.0

	1	2	3	4	5	6
1	0.0					
2	10.0	0.0				
3	2.7	11.0	0.0			
4	12.0	2.9	11.0	0.0		
5	2.6	11.0	2.5	12.0	0.0	
6	11.0	2.8	12.0	2.8	11.0	0.0

► Cluster 1: Obs. 1, 3, 5

► Cluster 2: Obs. 2, 4, 6

Different Types of Cluster Analysis

- ▶ There are a number of different main types of cluster analysis:
 - ▶ Algorithmic
 - ▶ Parametric
 - ▶ Non-parametric.
- ▶ We'll focus algorithmic clustering which involves defining a measure of clustering and optimising it.
- ▶ These measures are typically functions of distance (between points or between points and the cluster centres)
- ▶ Two of the most popular (algorithmic) clustering methods are hierarchical clustering (a hierarchical method) and k-means (a partitioning method).

Other methods

Parametric:

- ▶ Fit a density to each sub-population
- ▶ Overall population is modelled by a weighted summation of the individual component densities which is known as a finite mixture model.

$$\underline{x} \sim \sum_{g=1}^G \pi_g f_g(\underline{x}), \quad \text{with } 0 < \pi_g \leq 1, \forall g, \quad \sum_{g=1}^G \pi_g = 1$$

- ▶ The π_g are known as the mixing weights and the full vector of them as the mixing distribution

- ▶ Common example of parametric methods: model-based clustering
 - ▶ Components are given by Gaussian densities with (different) mean vectors and (possibly different) covariance matrices
 - ▶ Also known as Gaussian mixture modelling

$$\underline{x} \sim \sum_{g=1}^G \pi_g N(\underline{x} \mid \underline{\mu}_g, \Sigma_g)$$

This is commonly estimated in a Bayesian fashion or by the EM algorithm.

- ▶ Recent advance in algorithmic clustering: spectral clustering (2001)
 - ▶ Looks at constructing a particular type of distance matrix which is then spectrally decomposed and clustering done on the eigenvectors.
- ▶ Non-parametric clustering involves hunting for local modes in the data which are said to correspond to groups.

Hierarchical Clustering

- ▶ Two flavours: agglomerative and divisive.
- ▶ Focus on agglomerative hierarchical clustering.
- ▶ In agglomerative clustering, each data point starts out as its own (singleton) cluster and at each step we merge together the two closest clusters to create a new cluster. This merging continues until all points are merged into one cluster.
- ▶ For divisive clustering, we work in reverse: every objects is contained in one cluster at the start and at each stage we split clusters in two (in some optimal way) until we end up with every point having its own cluster

Agglomerative Hierarchical Clustering Algorithm

1. All points are their own own clusters.
2. (Re-)Calculate the distance between all current clusters
3. Merge the two clusters with the smallest distance into one new cluster
4. Iterate steps 2 and 3 until there is only one cluster

If we have n points there will be $n - 1$ merge steps.

Each merge is irreversible. Once merged, a pair of clusters stay merged from that step on until the end.

Need to decide: distance measure between points and distance measure between non-singleton clusters (called a linkage method)

Distance Measures

For p continuous variables, $\underline{x}_i' = (x_{i1}, \dots, x_{ip})$ we commonly use

- ▶ Euclidean distance defined as

$$d_E(\underline{x}_1, \underline{x}_2) = \sqrt{\sum_{j=1}^p (x_{1j} - x_{2j})^2}$$

- ▶ Or, weighted Euclidean distance:

$$d_{w.E}(\underline{x}_1, \underline{x}_2) = \sqrt{\sum_{j=1}^p w_j (x_{1j} - x_{2j})^2}$$

- ▶ An alternative option is Mahalanobis distance defined as

$$d_S(\underline{x}_1, \underline{x}_2) = \sqrt{(\underline{x}_1 - \underline{x}_2)' S^{-1} (\underline{x}_1 - \underline{x}_2)}$$

where S is the covariance matrix from the common distribution of the 2 vectors.

- ▶ Another alternative is (weighted) Manhattan distance

$$d_M(\underline{x}_1, \underline{x}_2) = \sum_{j=1}^p w_j |x_{1j} - x_{2j}|$$

- ▶ Another distance is maximum distance

$$d_{\max}(\underline{x}_1, \underline{x}_2) = \max_j |x_{1j} - x_{2j}|$$

- ▶ The cosine distance looks at the angle between the two vectors

$$d_{\cos}(\underline{x}_1, \underline{x}_2) = \frac{\underline{x}_1 \cdot \underline{x}_2}{\|\underline{x}_1\| \|\underline{x}_2\|}$$

Distance Measures

For p binary variables, we look at how many variables have the same value for the two objects resulting in a table of the form

		Object x_1	
		0	1
Object	0	a	b
x_2	1	c	d

There are two common measures of distance or dissimilarity based on this table.

- The first is the Jacard index

$$d_J(\underline{x}_1, \underline{x}_2) = \frac{b + c}{b + c + d}$$

which only considers variables where at least one object has a value 1.

- An alternative to the Jacard index is the distance comparing dissimilarities to all comparisons made:

$$d(\underline{x}_1, \underline{x}_2) = \frac{b + c}{a + b + c + d}$$

Linkage

- ▶ The linkage criteria determines the distance between clusters containing more than one point,
 - ▶ often as a function of the pairwise distances between observations in different clusters.
- ▶ Denote $L(A, B)$ as the linkage value between two clusters A and B and $d(x_a, x_b)$ as the distance between two points x_a and x_b .

Common Linkage Methods

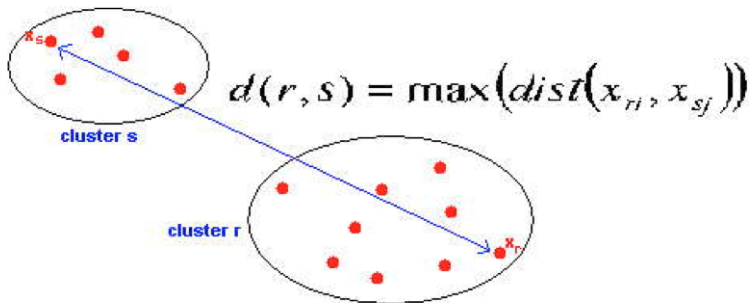
- ▶ Also known as *farthest neighbour* method
- ▶ Complete Linkage: the distance between clusters A and B is defined to be the maximum of the distances between all pairs of points with one point from cluster A and one point from cluster B.

$$L(A, B) = \max\{d(x_a, x_b) : x_a \in A, x_b \in B\}$$

Common Linkage Methods

- ▶ Also known as *farthest neighbour* method
- ▶ Complete Linkage: the distance between clusters A and B is defined to be the maximum of the distances between all pairs of points with one point from cluster A and one point from cluster B.

$$L(A, B) = \max\{d(x_a, x_b) : x_a \in A, x_b \in B\}$$



Example

Five multivariate observations were found to have the following dissimilarity matrix:

$$D = \begin{pmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 5 & 0 & & \\ 10 & 9 & 4 & 0 & \\ 9 & 8 & 5 & 3 & 0 \end{pmatrix}$$

Perform hierarchical clustering with complete linkage on these points

Example

- ▶ Which are the two most similar objects (smallest distance)?

Example

- ▶ Which are the two most similar objects (smallest distance)?
observations 1 and 2 with distance 2
- ▶ Merge these two and recalculate the distance between all
remaining single observations and the new cluster containing

Example

- ▶ Which are the two most similar objects (smallest distance)? observations 1 and 2 with distance 2
- ▶ Merge these two and recalculate the distance between all remaining single observations and the new cluster containing {1, 2}.
- ▶ The distance between observations 1 and 3 is 6 and 2 and 3 is 5. Since we are using complete linkage we take the maximum, so the distance between the cluster {1,2} and 3 is therefore 6. The distance between observations 1 and 4 is 10 and 2 and 4 is 9. Therefore, the distance between the cluster {1,2} and 4 is 10. The distance between 1 and 5 is 9 and 2 and 5 is 8. Therefore the distance between cluster {1,2} and 5 is 9. The new distance matrix at this step is given below:

	{1,2}	{3}	{4}	{5}
{1,2}	0			
{3}	6	0		
{4}	10	4	0	
{5}	9	5	3	0

Example

- ▶ Which are the two most similar clusters (smallest distance)?

Example

- ▶ Which are the two most similar clusters (smallest distance)?
observations 4 and 5 with distance/height 3
- ▶ Merge these two and recalculate the distance between all
remaining clusters and the new cluster containing

Example

- ▶ Which are the two most similar clusters (smallest distance)? observations 4 and 5 with distance/height 3
- ▶ Merge these two and recalculate the distance between all remaining clusters and the new cluster containing {4, 5}.
- ▶ The distance between {1,2} and 4 is 10 and {1, 2} and 5 is 9, therefore the distance between {1,2} and {4,5} is 10. The distance between 3 and 4 is 4 and 3 and 5 is 5, therefore the distance between 3 and {4,5} is 5. The new distance matrix at this step is given below:

$$D = \begin{array}{cc} & \begin{array}{ccc} \{1, 2\} & \{3\} & \{4, 5\} \end{array} \\ \begin{array}{c} \{1, 2\} \\ \{3\} \\ \{4, 5\} \end{array} & \begin{array}{ccc} 0.0 & & \\ 6 & 0.0 & \\ 10 & 5 & 0 \end{array} \end{array}$$

Example

- ▶ Which are the two most similar clusters (smallest distance)?

Example

- ▶ Which are the two most similar clusters (smallest distance)? 3 and {4,5} with distance/height 5.
- ▶ Merge these two and recalculate the distance between all remaining clusters and the new cluster containing

Example

- ▶ Which are the two most similar clusters (smallest distance)? 3 and {4,5} with distance/height 5.
- ▶ Merge these two and recalculate the distance between all remaining clusters and the new cluster containing {3, 4, 5}.
- ▶ The distance between {1,2} and 3 is 6 and {1,2} and {4,5} is 10, therefore the distance between {1,2} and {3,4,5} is 10. The new distance matrix at this step is given below:

$$D = \begin{array}{cc} & \begin{matrix} \{1, 2\} & \{3, 4, 5\} \end{matrix} \\ \begin{matrix} \{1, 2\} \\ \{3, 4, 5\} \end{matrix} & \begin{bmatrix} 0.0 & 10 \\ 10 & 0 \end{bmatrix} \end{array}$$

- ▶ The final join is {1,2} with {3,4,5} at height 10.

Note: Only the order of the distances mattered, not the actual values. So complete linkage clustering can be used on ordinal as well as metric distance measures. Same is true of the next method.

Common Linkage Methods

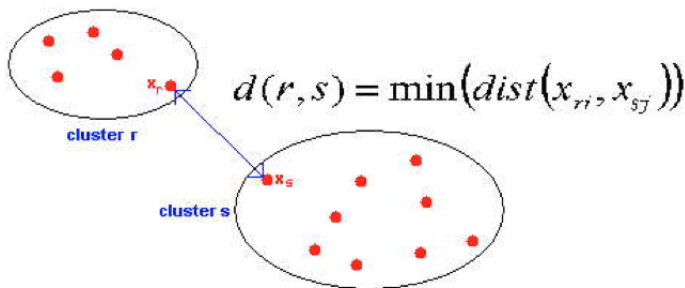
- ▶ Also known as *nearest neighbour* method
- ▶ Single Linkage: the distance between clusters A and B is defined to be the minimum of the distances between all pairs of points with one point from cluster A and one point from cluster B.

$$L(A, B) = \min\{d(x_a, x_b) : x_a \in A, x_b \in B\}$$

Common Linkage Methods

- ▶ Also known as *nearest neighbour* method
- ▶ Single Linkage: the distance between clusters A and B is defined to be the minimum of the distances between all pairs of points with one point from cluster A and one point from cluster B.

$$L(A, B) = \min\{d(x_a, x_b) : x_a \in A, x_b \in B\}$$



- ▶ Try clustering the previous example using the single linkage criterion

Common Linkage Methods

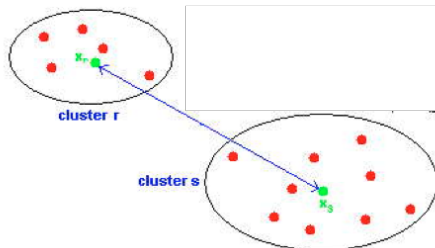
- ▶ Centroid Linkage: the distance between clusters A and B is defined to be the distance between the centroid of cluster A and the centroid of cluster B (where centroid is the point found by averaging all points in the cluster).

$$L(A, B) = d(\bar{x}_A, \bar{x}_B) \quad \text{where } \bar{x}_A = \frac{1}{|A|} \sum_{a: x_a \in A} x_a$$

Common Linkage Methods

- Centroid Linkage: the distance between clusters A and B is defined to be the distance between the centroid of cluster A and the centroid of cluster B (where centroid is the point found by averaging all points in the cluster).

$$L(A, B) = d(\bar{x}_A, \bar{x}_B) \quad \text{where } \bar{x}_A = \frac{1}{|A|} \sum_{a: x_a \in A} x_a$$



Note: This method requires us to treat the data as metric rather than ordinal.

Common Linkage Methods

- Average Linkage: the distance between clusters A and B is defined to be the average of all the distances between all pairs of points with one point from cluster A and one point from cluster B.

$$\begin{aligned} L(A, B) &= \frac{1}{\text{no. of pairs}} \sum_{\{a,b\}: x_a \in A, x_b \in B} d(x_a, x_b) \\ &= \frac{1}{|A||B|} \sum_{\{a,b\}: x_a \in A, x_b \in B} d(x_a, x_b) \end{aligned}$$

Common Linkage Methods

- ▶ Ward's Linkage: the distance between clusters A and B is defined to be the difference in the sum of squares for the combined cluster resulting from merging A and B, and the sum of the sum of squares for the two clusters separately
- ▶ The larger the increase in sum of squares from merging, the greater the distance.

$$L(A, B) = SS(A, B) - (SS(A) + SS(B))$$

$$\text{where } SS(A) = \sum_{a: \underline{x}_a \in A} (\underline{x}_a - \bar{\underline{x}}_A)^2$$

Properties of Different Linkages

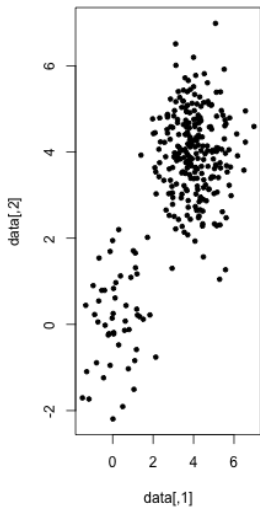
IMPORTANT: Regardless of linkage used we are always joining the clusters with the smallest distance/linkage at each step of the clustering algorithm.

Each linkage has different properties in the types of clusters it tends to find:

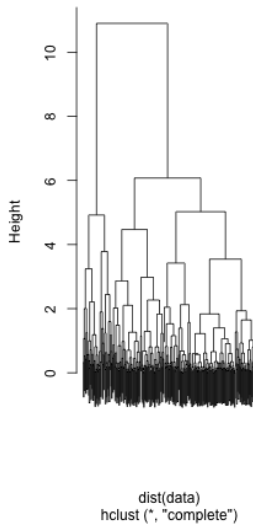
- ▶ Complete Linkage: good at separating overlapping, roughly spherical groups, bad at finding groups with non-standard shape
- ▶ Single Linkage: good at finding unusual shaped groups (as it tends to “chain” along, adding points to existing clusters, rather than starting merges in new clusters), tends to suggest joining overlapping groups
- ▶ Ward's Linkage: good at finding equal-sized spherical groups

Dendrogram

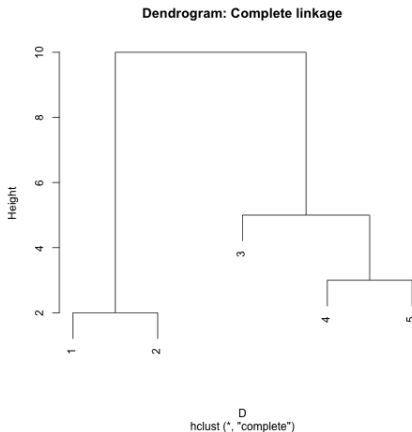
- ▶ A dendrogram (also sometimes called a dendrogram) is a tree structure which visualises the merges in the agglomerative hierarchical clustering procedure.
- ▶ Each observation is a leaf which merge in turn at different heights.
- ▶ The heights on the y-axis are the distances between the clusters being merged.



Cluster Dendrogram



Example Dendrogram



Try drawing a dendrogram for the results of single linkage clustering on the example distance matrix. Are they similar?

- ▶ Dendrograms are used to judge how many clusters are present in the data.
- ▶ Look for a large gap between joins far up in the tree
- ▶ Taken as evidence that far apart clusters have begun to be (incorrectly) merged and that the true clusters are the partition of data points defined by the merges prior to that point.
- ▶ A dendrogram is a useful tool for visualising the structure of data (how close different points are) in arbitrary dimensions.

R code for Hierarchical Clustering

```
> d<-dist(data)
> # dist by default given Euclidean distance,
for another type one must specify the method
argument
> # d<-dist(data, method="manhattan")
> res.clust.compl <- hclust(d,
method="complete")
> res.clust.singl <- hclust(d,
method="single")
> # hclust objects are a list with element
merge identifying which clusters were joined
at each step, element height identifies the
value of linkage between the two clusters
merged at each step and other elements.
> res.clust.compl$merge
```

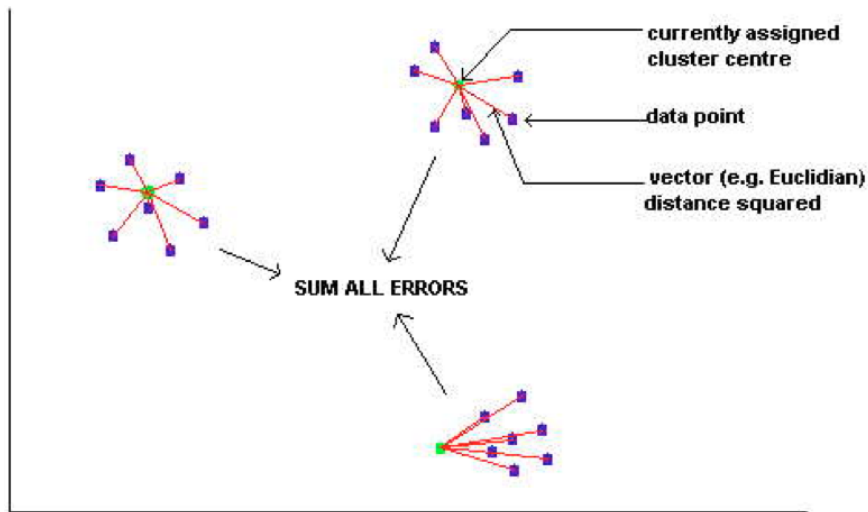
```
> #To get the dendrogram use plot on the  
fitted hclust object  
> plot(res.clust.compl)  
> # If we decide on a number of clusters from  
the dendrogram, 2 say, we use the command  
cutree to find which observations are assigned  
to each cluster  
> # We can either specify a height to cut at  
via argument h or a number of clusters via  
argument k  
> clusters<-cutree(res.clust.compl, k=2)
```

K-Means

- Try to find the assignment of observations to a fixed number of clusters K , that minimises the sum over all clusters of the sum of squares within clusters.

$$\sum_{k=1}^K \sum_{i: \underline{x}_i \in C_k} (\underline{x}_i - \bar{\underline{x}}_{C_k})^2$$

where $\bar{\underline{x}}_{C_k}$ is the average of all the points belonging to cluster k .



- ▶ In theory we would like to simply look at all possible assignments of points to clusters and choose the assignment that minimises the above criterion.
- ▶ **However**, the numbers involved in a complete enumeration of every possible assignment is so *vast* that for most problems it is impossible to do even on the fastest computer.

To illustrate the scale of the problem:

number of points	number of clusters	number of possible partitions
15	3	2,375,101
20	4	45,232,115,901
25	8	690,223,721,118,368,580
100	5	10^{68}

K-Means Algorithm

As a feasible alternative we consider the following algorithm (also known as Lloyd's algorithm):

1. Begin with K starting centres
2. Assign each observation to the cluster with the closest centre
3. Re-calculate the cluster centres by finding the centroids of each cluster's assigned observations
4. Iterate steps 2 and 3 until convergence.

Starting Values

- ▶ Usually we randomly assign K points to be the starting centres of the clusters.
- ▶ K-means, unfortunately, has a sensitivity to starting values and it is important to always run the algorithm from multiple random starts and then select the “best” run as the final result.

Selecting K - Elbow Plot

Need to specify K in advance

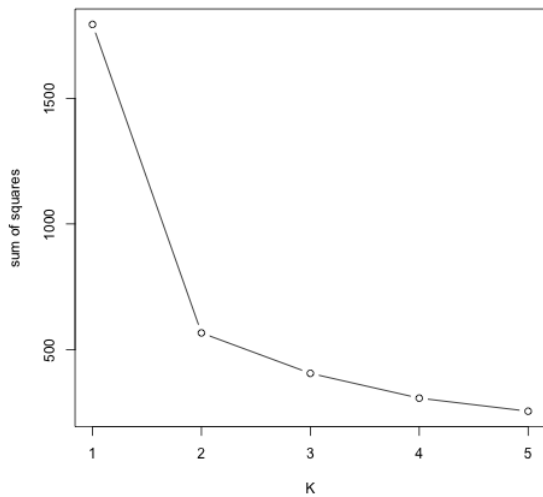
- ▶ Usually choose a range of sensible numbers of clusters (starting at 2 say up to a K_{max})
- ▶ K-means (with multiple random starts) is run on for each number of K
- ▶ A plot of the total sum of squares, W_K , versus K is drawn

$$D_k = \sum_{i:x_i \in C_k} \sum_{j:x_j \in C_k} d(x_i, x_j)^2$$

$$W_K = \sum_{k=1}^K \frac{1}{2} D_k$$

- ▶ The number of K is chosen usually to be where there is a bend in the lines joining the values, hence it is called an “elbow plot”

Elbow Plot



Selecting K - Gap Statistic

- ▶ Issues with Elbow plot:
 - ▶ Subjective
 - ▶ No reference distribution for the null hypothesis
 - ▶ Difference in W_k 's not normalized.
- ▶ Instead use the Gap statistic:

$$\text{Gap}(k) = E(\log(W_K)) - \log(W_K)$$

where the expectation is taken with respect to the (null hypothesis) reference distribution

- ▶ Choose K that optimises the gap statistic

Selecting K - Calculating the Gap Statistic

for b in 1 to B

Simulate a non-clustering dataset x_{1b}, \dots, x_{nB}

for K in 1 to K_{\max}

Cluster the original observations x_1, \dots, x_n into K clusters and compute $\log(W_K)$

for b in 1 to B

Cluster the b^{th} bootstrap sample into K groups and compute $\log(W_{Kb})$

Compute $\text{Gap}(K) = \frac{1}{B} \sum_{b=1}^B \log(W_{Kb}) - \log(W_K)$

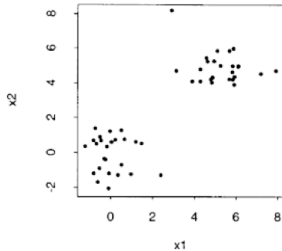
Compute $s.d.(K)$, the standard deviation of $\{\log(W_{Kb})\}_{b=1, \dots, B}$

Compute the total standard error, s_K as

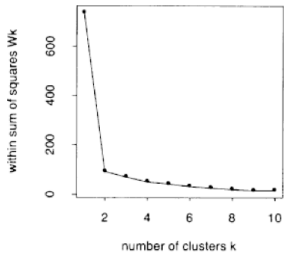
$$s_K = \sqrt{1 + \frac{1}{B}} \times s.d.(K)$$

Choose the smallest K such that $\text{Gap}(K) \geq$

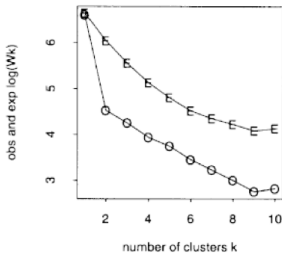
$\text{Gap}(K+1) - s_{K+1}$



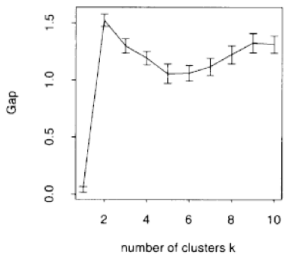
(a)



(b)



(c)



(d)

Fig. 1. Results for the two-cluster example: (a) data; (b) within sum of squares function W_k ; (c) functions $\log(W_k)$ (O) and $\hat{E}_n^*(\log(W_k))$ (E); (d) gap curve

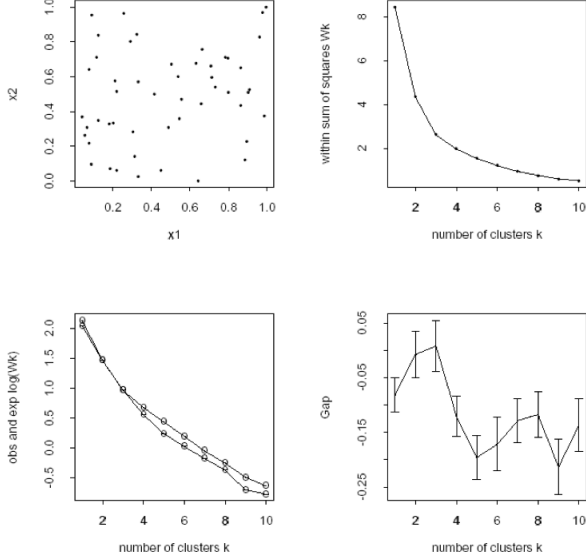


Figure 2: Results for uniform data example. The data is in top left, and the within sum of squares function W_k is displayed in the top right. The functions $\log(W_k)$ and $\hat{E}_n^*(\log(W_k))$ are shown in the bottom left panel (plotting symbol "o" and "e" respectively), with the gap curve displayed in the bottom right.

Selecting K - Average Silhouette Width

Another method for choosing K is selecting the value in the range that gives the largest silhouette width.

For each observation x_i , the silhouette width $s(i)$ is defined as follows:

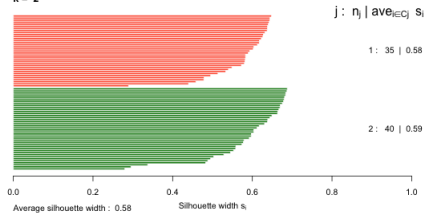
- ▶ Put $a(i)$ = average dissimilarity between x_i and all other points of the cluster to which x_i belongs (if x_i is the only observation in its cluster, $s(i) = 0$ without further calculations).
- ▶ For each of the other clusters C , put $d(i, C)$ = average dissimilarity of x_i to all observations of C .
- ▶ The smallest of these $d(i, C)$ is $b(i) = \min_C d(i, C)$, and can be seen as the dissimilarity between x_i and its “neighbour” cluster, i.e., the nearest one to which it does not belong. Finally,

$$s(i) = \frac{(b(i) - a(i))}{\max(a(i), b(i))}$$

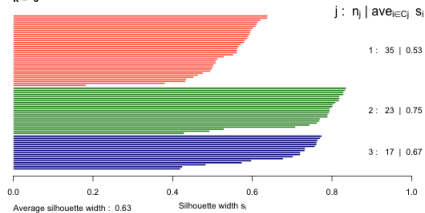
Silhouette Width Properties

- ▶ Observations with a large $s(i)$ (almost 1) are very well clustered
- ▶ A small $s(i)$ (around 0) means that the observation lies between two clusters
- ▶ Observations with a negative $s(i)$ are probably placed in the wrong cluster.

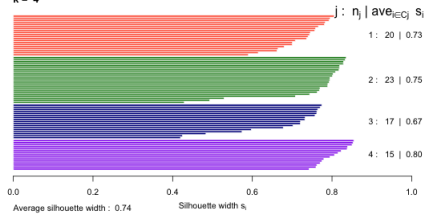
k = 2



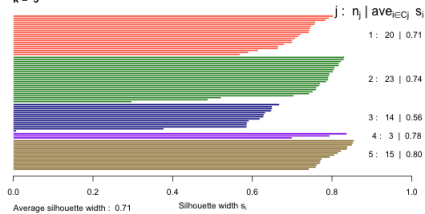
k = 3



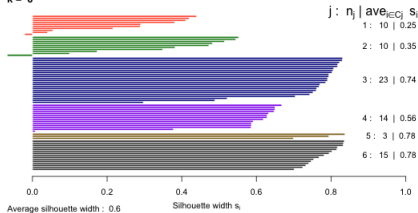
k = 4



k = 5



k = 6



R code for K-means

```
> # To run k-means on a dataset data for K = 4  
and with 3 random starts  
> res.kmeans <- kmeans(data,4,nstart=3)  
> # This produces a list with a vector cluster  
identifying which cluster each point in data  
belongs to,  
> # centres, a Kxd matrix with each row being  
the final estimate of a cluster centre
```

```
> # To produce the elbow plot we separately
calculate the sum of squares for 1 cluster
n <- nrow(data)
ss<-rep(0,K.max)
ss[1] <- (n - 1) * sum(apply(data, 2, var))
for(i in 2:K.max)
{
ss[i] <- sum(kmeans(data,centers = i,
nstart=3)$withinss)
}
plot(c(1:K.max),ss,type="b",main="Elbow
Plot",xlab="K",ylab="sum of squares")
```

R code for Gap Statistic

```
> # To calculate the Gap statistic on k-means  
for a maximum of 5 clusters with  
> #100 bootstrap samples  
> library(cluster)  
> gap.st <- clusGap(data, FUN = kmeans, nstart  
= 20, K.max = 5, B = 100)  
> # To plot this against the number of  
clusters we would do the following:  
> plot(gap.st)
```

R code for Silhouette Width

```
> library(cluster)
> si <- silhouette(cluster, dist(data))
> summary(si)
> plot (si)
```

K-Medoids

Instead of using centroids, we could use actual points as the “prototypes” or centres of our clusters. This idea results in the k-medoids method.

The algorithm for fitting k-medoids is very similar to that of k-means:

1. Start with K randomly selected points for cluster medoids
2. Assign each observation to the cluster with the closest medoid
3. For each of the K clusters:
 - 3.1 For each non-medoid point in the cluster k , make this the new cluster medoid
 - 3.2 Compute the cost of the configuration
 - 3.3 Choose the point with the lowest cost as the new cluster medoid
4. Iterate steps 2 and 3 until convergence

R code for K-medoids

```
> library(cluster)
> # To fit k-medoids for 2 clusters:
> res.pam <- pam(data, 2)
> plot(res.pam)
> # To extract the cluster assignments
> res.pam$clust
```

Comparing Clusterings

- ▶ Want to measure how similar two different clustering assignment of (the same set of n) points to clusters are
- ▶ Number of clusters is arbitrary we cannot simply look to see if points are assigned to the same number cluster in both clusterings.
- ▶ For a single clustering and a pair of points there are two possibilities:
 - ▶ Both points are assigned to the same cluster
 - ▶ The points are assigned to different clusters

Comparing Clusterings

- For two clusterings CA_1 and CA_2 we have four possibilities:

		Under CA_1	
		Pairs assigned to same cluster	Pairs assigned to different clusters
Under CA_2	Pairs assigned to same cluster	a	b
	Pairs assigned to different clusters	c	d

- ▶ Count the number of pairs in the data falling in each of the four options and construct a measure of similarity between the 2 clusterings based on these four numbers
- ▶ Similar clusterings will give large values of a and d and small values of b and c .
- ▶ $a + b + c + d = \binom{n}{2}$, the total number of possible pairs of n points.
- ▶ We have already seen another measure for comparison in the Jacard index above, which for comparison is changed to $\frac{a}{a+b+c}$.

Note that this table does not requires the two clusterings to have the same number of clusters (e.g. CA_1 could have 2 clusters and CA_2 could have 3.)

Rand Index

- ▶ The Rand Index is

$$\frac{a + d}{a + b + c + d} = \frac{a + d}{\binom{n}{2}}.$$

- ▶ The Rand Index gives only positive numbers.
- ▶ The maximum number of the Rand Index is 1, indicating perfect agreement between the two clusterings.

Adjusted Rand Index

- ▶ The Adjusted Rand Index (ARI) due to Hubert and Arabie is the Rand Index adjusted for chance (so that the expected ARI of two random clusterings is 0).
- ▶ The Adjusted Rand Index is given by

$$ARI = \frac{\text{Rand Index} - \text{Expected Value}}{\text{Max Index} - \text{Expected Value}}$$

- ▶ The maximum value of the adjusted Rand Index is still 1 but it can now occasionally give negative numbers.

Fowlkes and Mallows Index

- ▶ The Fowlkes and Mallows Index is defined to be

$$\frac{a}{\sqrt{(a+b)(a+c)}}$$

R code for Comparing Clusterings

```
> library(clues)
> adjustedRand(clust1, clust2, method=c("Rand",
"HA", "FM", "Jacard") )
```


Comparing clusterings example

- ▶ For a dataset of 5 items, the following 2 clustering assignment results were produced:

Clustering A: 1, 2, 2, 1, 1

Clustering B: 2, 1, 2, 1, 1


- ▶ These clusterings differ only in the cluster assignments of the first 2 observations.
- ▶ Let's construct a table comparing the clusterings (looking at all pairs of observations) and using the table calculate the Rand, Fowlkes-Mallows and Jacard indices.

Comparing clusterings example

- There are $\binom{5}{2} = 10$ possible pairs in the data. We list these in a table and check to see whether they are in the same cluster (1) or not (0) in each assignment.

Pair	A	B
1,2	0	0
1,3	0	1
1,4	1	0
1,5	1	0
2,3	1	0
2,4	0	1
2,5	0	1
3,4	0	0
3,5	0	0
4,5	1	1


Comparing clusterings example



		Under <i>Cl.A</i>	
		Pairs assigned to same cluster	Pairs assigned to different clusters
Under	Pairs assigned to same cluster	1	3
<i>Cl.B</i>	Pairs assigned to different clusters	3	3

► The Rand index is therefore =

Comparing clusterings example



		Under <i>Cl.A</i>	
		Pairs assigned to same cluster	Pairs assigned to different clusters
Under	Pairs assigned to same cluster	1	3
<i>Cl.B</i>	Pairs assigned to different clusters	3	3


- ▶ The Rand index is therefore $= \frac{1+3}{10} = 0.4$,
- ▶ The Fowlkes-Mallow index =

Comparing clusterings example

		Under <i>Cl.A</i>	
		Pairs assigned to same cluster	Pairs assigned to different clusters
Under	Pairs assigned to same cluster	1	3
<i>Cl.B</i>	Pairs assigned to different clusters	3	3

- ▶ The Rand index is therefore $= \frac{1+3}{10} = 0.4$,
- ▶ The Fowlkes-Mallow index $= \frac{1}{\sqrt{(1+3)(1+3)}} = 0.25$
- ▶ The Jacard index =

Comparing clusterings example



		Under <i>Cl.A</i>	
		Pairs assigned to same cluster	Pairs assigned to different clusters
Under	Pairs assigned to same cluster	1	3
<i>Cl.B</i>	Pairs assigned to different clusters	3	3

- ▶ The Rand index is therefore $= \frac{1+3}{10} = 0.4$,
- ▶ The Fowlkes-Mallow index $= \frac{1}{\sqrt{(1+3)(1+3)}} = 0.25$
- ▶ The Jacard index $= \frac{1}{1+3+3} = 0.143$.