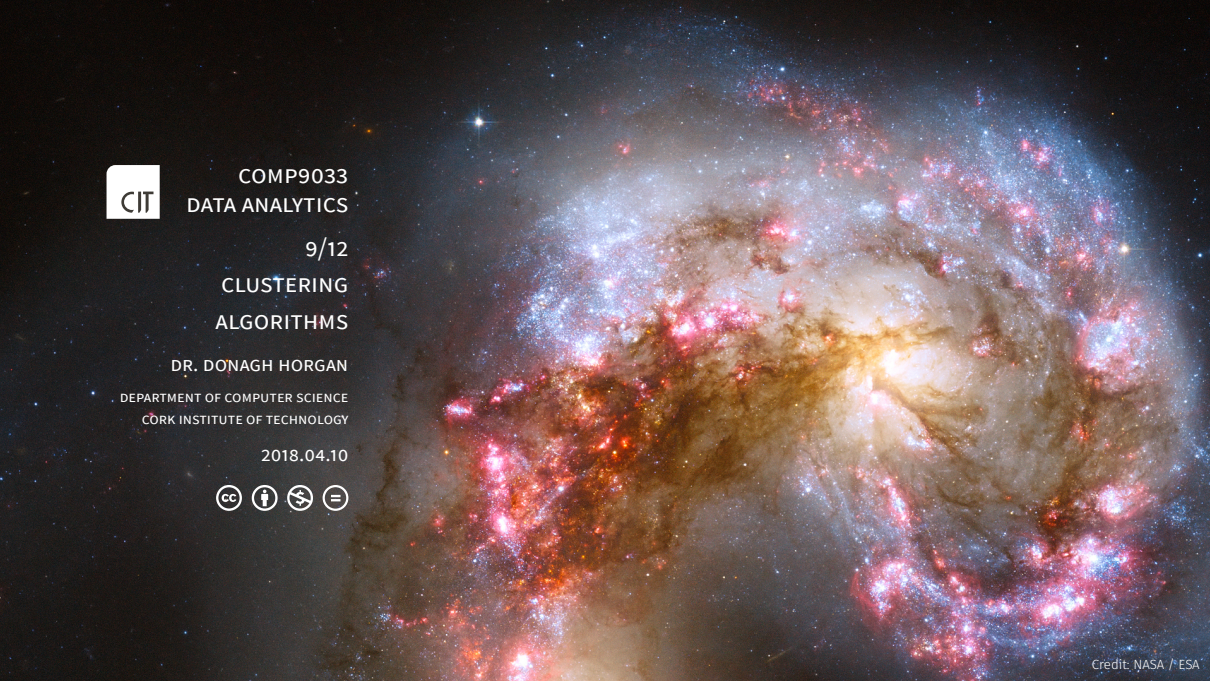


COMP9033 DATA ANALYTICS

9/12 CLUSTERING ALGORITHMS

DR. DONAGH HORGAN

DEPARTMENT OF COMPUTER SCIENCE
CORK INSTITUTE OF TECHNOLOGY

2018.04.10

Overview

1. Recommender systems:
   - The long tail phenomenon.
   - Content-based recommenders.
   - Collaborative filters.
   - Advantages and disadvantages.
   - The Netflix Prize.

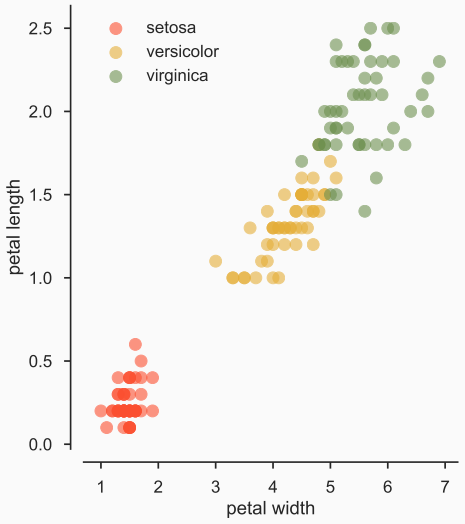
2. Nearest neighbours:
   - k nearest neighbours.
   - Regression.
   - Classification
   - Hyperparameters.
   - Advantages and disadvantages.

1. Clustering:
    - What it is.
    - How it is used.
    - Real world examples.
    - Flavours / variants.
2. Combinatorial clustering:
    - Brute force.
    - *K*-means.
    - *K*-medoids.
    - Performance considerations.
    - How to choose *K*.
3. Hierarchical clustering:
    - Divisive vs. agglomerative.
    - Dendrograms.
    - Determining similarity.

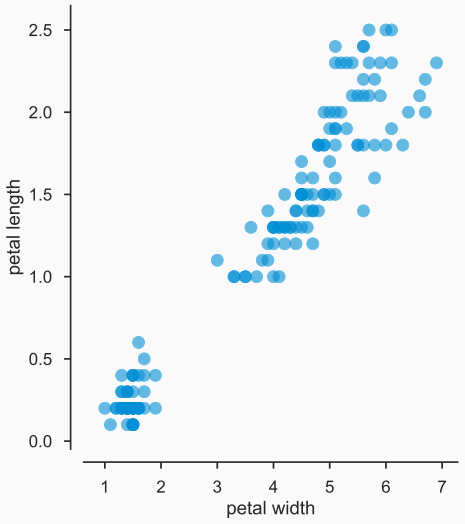
Clustering

- *Clustering* algorithms are a class of machine learning algorithms that can be used to analyse the structure of data, *e.g.*
  - To determine whether data consists of a number of distinct subgroups.
  - To determine whether the data supports a hierarchical structure.
- Clustering is an *unsupervised* machine learning technique:
  - Supervised learners extract information from data that provides some some form of feedback to indicate whether what is learned is “right” or “wrong”.
  - Unsupervised learners extract information from data without any form of feedback or ground truth.
- It is similar to classification, in the sense that it can be used to categorise data — but we usually don’t know how many categories there are or the true category label of any sample.

- Recall the Iris flower data set, which contains 150 samples relating to three distinct species: *Iris setosa*, *Iris versicolor* and *Iris virginica*.
- The Iris data set is *labelled*, *i.e.* it's possible to check which species a given sample belongs to in advance of any analysis.
- Let's pretend that these labels aren't available, and try to identify the categories instead using clustering.



- Stripping away the labels, we are left with 150 unclassified data samples:
  - We don't know how many categories are contained in the data.
  - We don't know which samples belong to which categories.



- The image on the right shows what happens when we try to segment the data into *three* clusters.
- It's not perfect, but it's pretty close to the original!

- However, we often don't know how many clusters are contained in our data.
- The image on the right shows what happens when we try to partition the data into *five* clusters.

1. Search engines: grouping related search results.
2. Market research: partitioning of survey responses into distinct categories.
3. Social network analysis: determining distinct groups within a wide network of contacts.
4. Localisation: determine the position of an asset by clustering its radio activity.
5. Insurance fraud: clustering people who have had accidents and then finding deviations.
6. Text analytics: finding similar documents, determining topics.
7. Human genome research: DNA clusters correspond to geographic origins.

- There are many algorithms available for clustering data, *e.g.*
  - Combinatorial clustering: clusters are assigned based on a combinatorial optimisation, *e.g.* *K*-means and *K*-medoids.
  - Hierarchical clustering: clusters are assigned and arranged in a hierarchical fashion, *e.g.* agglomerative and divisive clustering.
  - Mode seeking: clusters are assigned based on how close they are to the modes of an estimated distribution of the data, *e.g.* the PRIM algorithm.
  - Mixture modelling: The data is modelled as a mixture of one or more known distributions (*e.g.* via Gaussian mixture modelling), which are then used to form clusters.
- While combinatorial, hierarchical and mode seeking clustering do not make assumptions about the distribution of the data, mixture modelling does.

Combinatorial clustering

- The term *combinatorial clustering* describes a class of clustering algorithms that group items in a similar fashion, *e.g.*
  - K-means clustering.
  - K-medoids clustering.
  - Brute force combinatorial clustering.
- Generally, combinatorial clustering works as follows:
  1. Randomly assign each observation to one of K clusters.
  2. Compute how well each cluster fits using *a measure of fit*.
  3. Change the cluster assignment *in some way* and reassess the fit.
  4. Repeat Step 3 until a *satisfactory* cluster assignment is found.
- Different algorithms are distinguished by the use of different measures of fit (in Step 2), different cluster assignment strategies (in Step 3) and different satisfaction criteria (in Step 4).

- Generally, the goodness of fit of a particular cluster assignment is measured by its *inertia*.
- For quantitative variables, inertia is usually measured as

$$I_j = \sum_{\mathbf{x}_i \in C_j} d(\mathbf{x}_i, \mu_j), \tag{9.1}$$

  where I_j is the inertia of the j^{th} cluster, C_j is the set of points in the j^{th} cluster, $\mathbf{x}_i$ is the i^{th} point in C_j, μ_j is the *centre value* of C_j and $d(\mathbf{a}, \mathbf{b})$ measures the distance between the vectors $\mathbf{a}$ and $\mathbf{b}$.
- The centre value, μ_j, may be calculated in a number of different ways, *e.g.* mean, median, mode.
- The distance function can also be specified in a number of different ways, *e.g.* Euclidean, Manhattan.

- Brute force combinatorial clustering involves checking every possible cluster assignment and selecting the best one.
- The inertia can be computed using any suitable centre value measure and distance measure.
- The approach is exhaustive, but the final cluster assignment is *optimal* in the sense that it minimises the inertia across all clusters.
- Generally, this approach is only practical when dealing with very small numbers of samples.

- For N samples and K clusters, the total number of assignment combinations is given by

$$C(N, K) = \frac{1}{K!} \sum_{k=1}^{K} \binom{K}{k} (-1)^{K-k} k^N. \tag{9.2}$$

- Equation 9.2 implies that the brute force approach won't work for large N:
  - For instance, $C(10, 4) = 34105$ (reasonably easy to compute), while $C(19, 4) \approx 10^{10}$ (much more difficult).
  - Generally, we have far more than 19 samples!

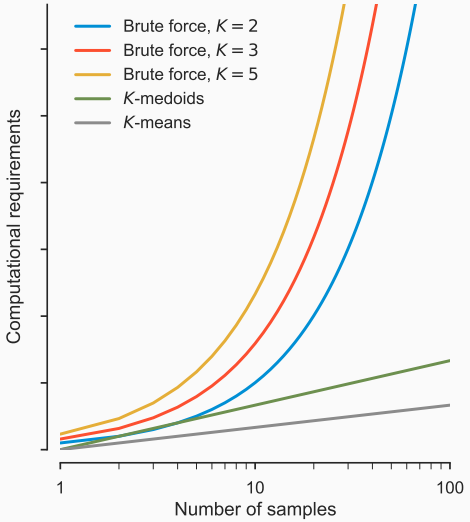
- *K*-means clustering is a very popular combinatorial clustering algorithm and works as follows:
  1. Randomly assign each observation to a cluster.
  2. Calculate the mean of the samples in each cluster.
  3. Calculate the Euclidean distance of each sample to each cluster mean.
  4. Assign each sample to the cluster whose mean it is closest to.
  5. Repeat Steps 2-4 until the cluster assignments cease to change.
- Typically, the algorithm is run several times with different random initialisations and the best outcome (*i.e.* minimum overall inertia) is selected.

- *K*-means is generally faster than a brute force approach when clustering large numbers of samples.
- However, the cluster assignment is heuristic, so the final cluster assignment may not be globally optimal.
- It can only be used with *quantitative* data as it depends the arithmetic mean and Euclidean distance to compute the cluster centre value and individual point distances, respectively.
- It is also not robust to outliers, as cluster inertia depends on the mean and Euclidean distance, which are themselves not robust measures.

- *K*-medoids clustering is a popular alternative to *K*-means clustering and works as follows:
  1. Randomly assign each observation to a cluster.
  2. Calculate the *medoid* of the samples in each cluster, *i.e.* the sample that has minimum distance to all of the other samples in the cluster.
  3. Calculate the distance of each sample to the cluster medoids.
  4. Assign each sample to the cluster whose medoid it is closest to.
  5. Repeat Steps 2-4 until the cluster assignments cease to change.
- Again, typically, the algorithm is run several times and the best outcome is chosen.

- *K*-medoids is faster than a brute force approach, but slower than *K*-means.
- Like *K*-means, the cluster assignment is heuristic, and so the final result may not be globally optimal.
- However, as the medoid is used to measure the cluster centre and the distance metric is not restricted to Euclidean distance, *K*-medoids can be more robust in the presence of outliers.
- Also, depending on the distance measure used, it can be used with categorical data.

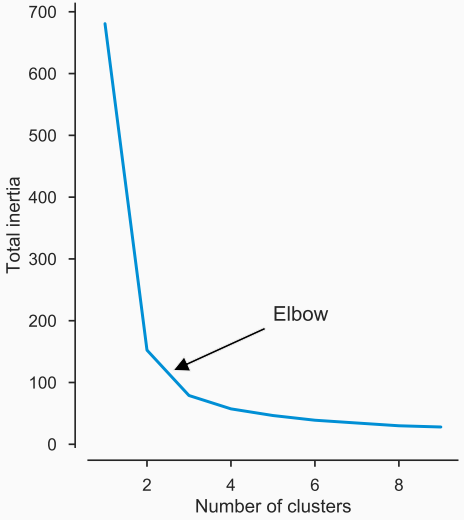
- We can express the amount of computation required to partition N samples into K clusters using big O notation:
  - Brute force: $O(K^N)$.
  - K-means: $O(N)$.
  - K-medoids: $O(N^2)$.
- The graph opposite shows how these vary as the number of samples grows.
- While K-means is the least expensive algorithm, this comes at the cost of increased susceptibility to outliers and a restriction to quantitative data only.



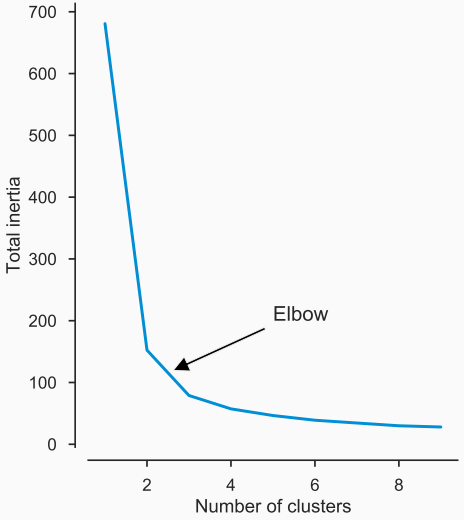
- Sometimes, the number of clusters to partition the data into is known in advance (*e.g.* the Iris data set has three categories).
- However, in general, this is not the case and we are interested in finding the choice of K that partitions the data optimally.

- As we have no labelled examples, we cannot use cross validation to select parameters or test our generalisation error.
- Instead, we must rely on a *heuristic* approach.

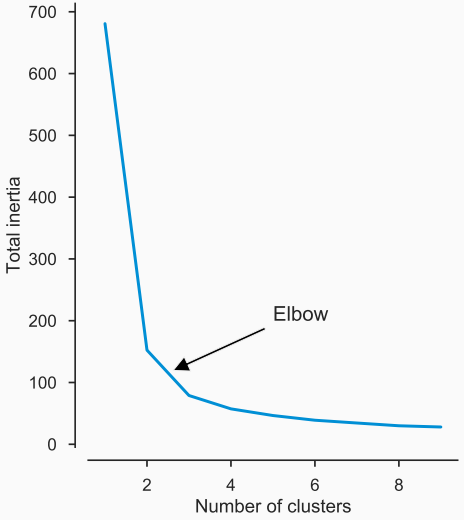
- Let's assume that there's a value $K = K'$ that represents the true number of clusters in the data.
- If we plot the total inertia ($\sum_j I_j$), we might expect to find value of K that corresponds to some minimum.
- However, the total inertia always decreases with increasing K: we can always get a better fit by adding another cluster!

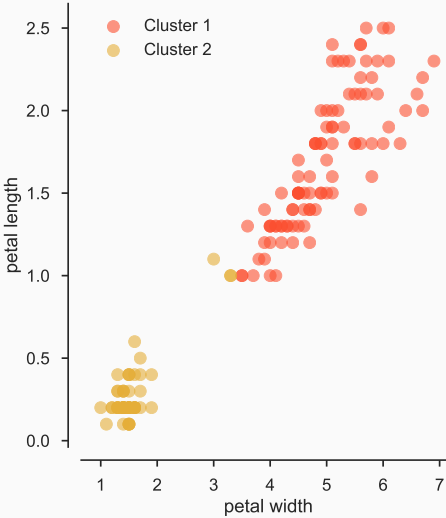


- However, as K increases towards K', then the total inertia should decrease rapidly as the clusters better fit the data.
- After we reach K', further increases in K have a diminishing return: we start to create unnecessary clusters, that don't add as much value as the previous ones.
- The optimum choice of K is therefore at the "elbow" of the curve, where a further increase in K does not produce a significant reduction in the sum.



- Unfortunately, this is often difficult to define precisely.
- For instance, in the curve to the right, we could say that $K' = 2$ or $K' = 3$.
- As it happens, this curve represents a clustering of the Iris dataset, and so we know that $K' = 3$ in reality.

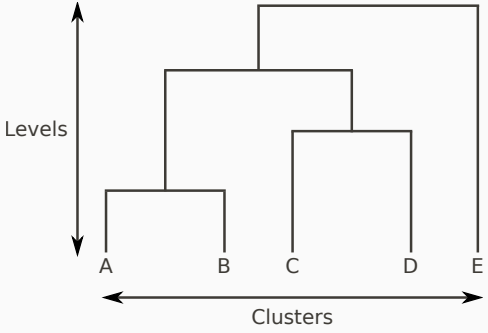



Cluster 1
Cluster 2
petal length
petal width
0.0
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7

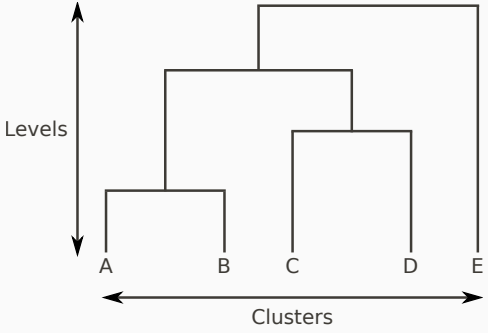

Cluster 1
Cluster 2
Cluster 3
petal length
petal width
0.0
0.5
1.0
1.5
2.0
2.5
1
2
3
4
5
6
7

Hierarchical clustering

- *Hierarchical clustering* is an alternative approach to combinatorial clustering.
- Unlike combinatorial clustering, it does not require us to choose a value for *K*.
- There are two basic variations:
  1. Divisive hierarchical clustering, *i.e.* a top down approach.
  2. Agglomerative hierarchical clustering, *i.e.* a bottom up approach.



- For a sample with N observations, hierarchical clustering results in the formation of a tree structure with $N - 1$ levels.
- Typically, the cluster hierarchy is represented using a dendrogram (shown opposite).
- It is a matter of choice as to which level represents a “natural” clustering of the input data.

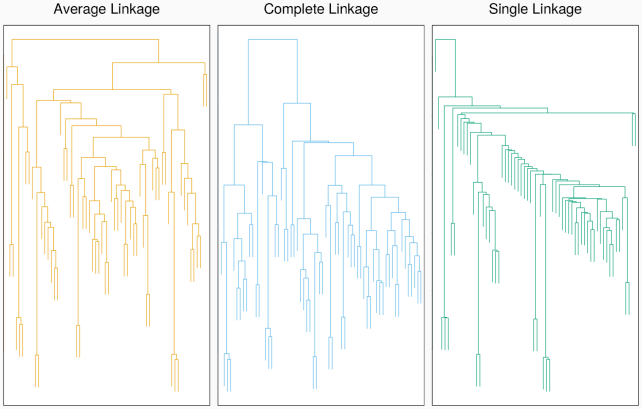


- Divisive clustering is similar to decision tree classification and works as follows:
  1. Initially, all the observations are grouped into a single cluster (i.e. $K = 1$).
  2. The cluster is split into two subclusters according to some splitting criterion. Usually, this involves maximising the dissimilarity between the resulting subclusters.
  3. Step 2 is repeated for each new subcluster until each observation occupies a unique cluster (i.e. $K = N$).
- Divisive clustering can be useful when the desired number of clusters is small, as only a small number of divisions need be performed.

- Agglomerative clustering works as follows:
  1. Each observation is assigned to its own cluster (i.e. $K = N$).
  2. At each level, the two least dissimilar clusters are merged until just one cluster remains (i.e. $K = 1$).
- Dissimilarity between clusters is typically measured using a distance metric, *e.g.* Manhattan, Euclidean.
- Dissimilarity between clusters is also characterised by a *linkage* property, *i.e.*
  1. Single linkage.
  2. Complete linkage.
  3. Average linkage.

1. Single linkage:
   - The dissimilarity of clusters is measured by the closest pair of members from each cluster.
   - Can lead to the clustering of dissimilar groups, where a single pair of observations in each group are close.
2. Complete linkage:
   - The dissimilarity of clusters is measured by the furthest pair of members from each cluster.
   - Can result in similar groups being rejected because one pair of observations are dissimilar.
3. Group average:
   - The dissimilarity of clusters is measured by the average dissimilarity of the members of each.
   - Blends the advantages and disadvantages of single and complete linkage.

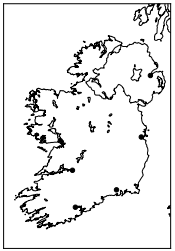
- The use of different linkage techniques can lead to completely different outcomes!



Credit: Hastie et al.

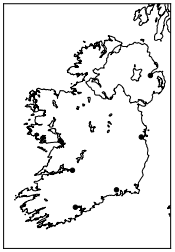
Q. A large retailer wishes to open two warehouses to support their operations on the island of Ireland.

Where should it locate the warehouses in order to best serve the six largest cities?



A. We can solve this problem using agglomerative clustering: starting with as many clusters as we have observations (*i.e.* one cluster per city), we can merge them until we have just two clusters left.

If we use single linkage merging, then we will create clusters based on the *closest* pair of points in each cluster. This way, we can find the pair of clusters that represent the best choice to locate the warehouses.



	BELFAST	CORK	DUBLIN	GALWAY	LIMERICK	WATERFORD
BELFAST	-	422	167	369	364	331
CORK	422	-	261	198	102	**84**
DUBLIN	167	261	-	208	197	165
GALWAY	369	198	208	-	99	231
LIMERICK	364	102	197	99	-	128
WATERFORD	331	**84**	165	231	128	-

- According to the data in the table, the closest pair of cities are Cork and Waterford, so we will merge these first.
- The distances between the new cluster and the existing clusters are given by their distances from Cork or Waterford, whichever is nearer (due to single linkage).

	BELFAST	CORK/WATERFORD	DUBLIN	GALWAY	LIMERICK
BELFAST	-	331	167	369	364
CORK/WATERFORD	331	-	165	198	102
DUBLIN	167	165	-	208	197
GALWAY	369	198	208	-	**99**
LIMERICK	364	102	197	**99**	-

- The next closest pair of clusters is now Galway and Limerick, so we will merge these next.
- Again, the distances between the new cluster and each of the existing clusters are given by the distances of those clusters from Galway or Limerick, whichever is nearer.

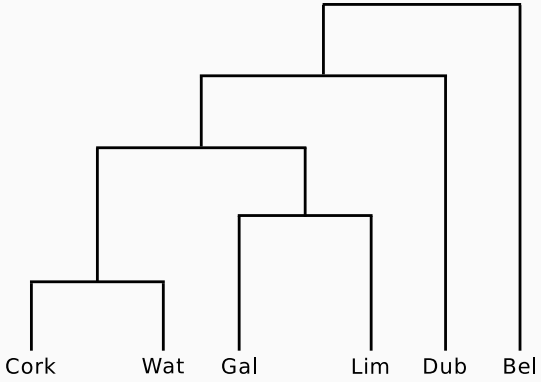
	BELFAST	CORK/WATERFORD	DUBLIN	GALWAY/LIMERICK
BELFAST	-	331	167	364
CORK/WATERFORD	331	-	165	**102**
DUBLIN	167	165	-	197
GALWAY/LIMERICK	364	**102**	197	-

- The next closest pair of clusters is now Cork/Waterford and Galway/Limerick, so we will merge these next.

	BELFAST	CORK/WATERFORD/GALWAY/LIMERICK	DUBLIN
BELFAST	-	331	167
CORK/WATERFORD/GALWAY/LIMERICK	331	-	**165**
DUBLIN	167	**165**	-

- The next closest pair of clusters is now Cork/Waterford/Galway/Limerick and Dublin, so we will merge these next.
- As there are just two clusters left after this merge – Belfast and Cork/Waterford/Galway/Limerick/Dublin – we merge them to create the final unified cluster.

- The dendrogram corresponding to the merges is shown opposite.
- As can be seen, there are five (*i.e.* $N - 1$) levels.



Summary

- Clustering:
  - A analysis technique for partitioning data into a number of distinct groups.
  - Combinatorial clustering: clusters are assigned by trying different combinations of assignments and selecting the best.
  - Hierarchical clustering: clusters are assigned in a hierarchical manner.
- Lab work:
  - Cluster data using the *K*-means algorithm.
  - Plot the sum of distances for different values of *K*, so that you can visualise the "elbow" of the curve and pick an appropriate value of *K*.
- Next week: association rule mining!

1. Hastie et al. *The Elements of Statistical Learning: Data mining, Inference and Prediction.* 2nd edition, February 2009. (stanford.io/2i1T6fN)
2. Ullman et al. *Mining of Massive Data Sets.* Cambridge University Press, 2014. (stanford.io/1qtgAYh)