dm25s1

Topic 11 : Clustering

Part 01 : Overview Hierarchical

Dr Bernard Butler

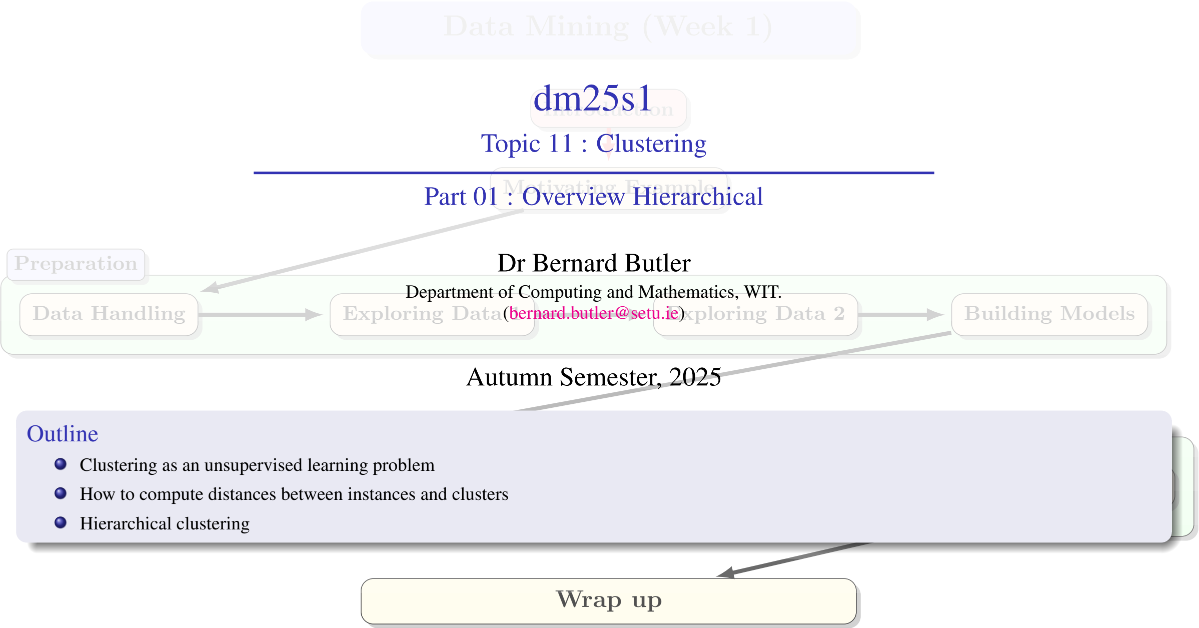
Department of Computing and Mathematics, WIT.

(bernard.butler@setu.ie)

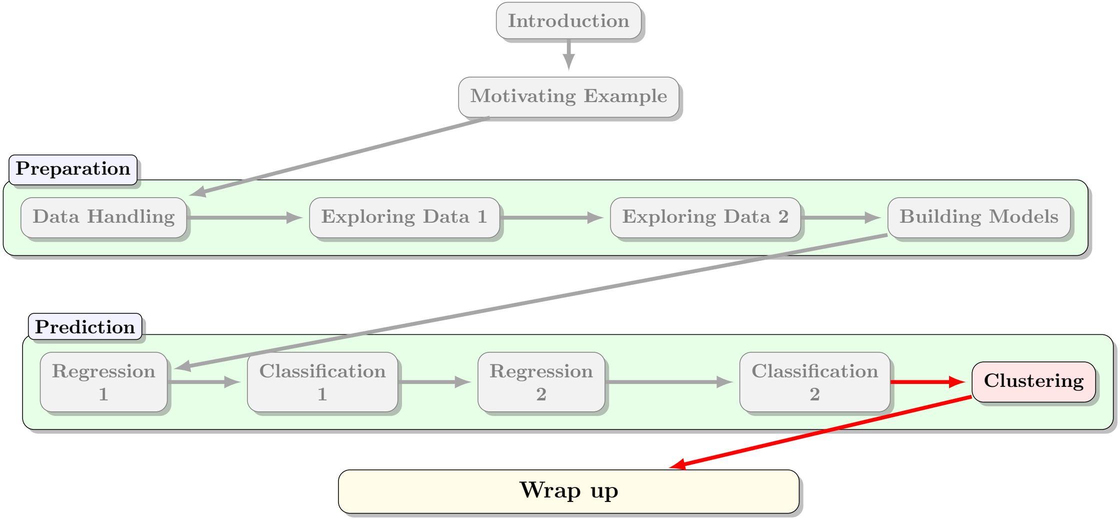
Autumn Semester, 2025

Outline

- Clustering as an unsupervised learning problem
- How to compute distances between instances and clusters
- Hierarchical clustering



Data Mining (Week 11)



Introduction

Motivating Example

Preparation

Data Handling

Exploring Data 1

Exploring Data 2

Building Models

Prediction

Regression 1

Classification 1

Regression 2

Classification 2

Clustering

Wrap up

Outline

1. Introduction 3

2. Distance Measures 8

3. Hierarchical Clustering 13

This Week's Aim

This week's aim is to introduce the main concepts and representative algorithms used in cluster analysis.

- Introduction to unsupervised learning

This Week's Aim

This week's aim is to introduce the main concepts and representative algorithms used in cluster analysis.

- Introduction to unsupervised learning
- Clustering as a means of understanding data

This Week's Aim

This week's aim is to introduce the main concepts and representative algorithms used in cluster analysis.

- Introduction to unsupervised learning
- Clustering as a means of understanding data
- Choice of distance function and metaparameters

This Week’s Aim

This week’s aim is to introduce the main concepts and representative algorithms used in cluster analysis.

- Introduction to unsupervised learning
- Clustering as a means of understanding data
- Choice of distance function and metaparameters
- Clusters that partition the data

This Week's Aim

This week's aim is to introduce the main concepts and representative algorithms used in cluster analysis.

- Introduction to unsupervised learning
- Clustering as a means of understanding data
- Choice of distance function and metaparameters
- Clusters that partition the data
    - Iris data: predicting which of three species

This Week’s Aim

This week’s aim is to introduce the main concepts and representative algorithms used in cluster analysis.

- Introduction to unsupervised learning
- Clustering as a means of understanding data
- Choice of distance function and metaparameters
- Clusters that partition the data
    - Iris data: predicting which of three species

Clustering is a long-established form of analysis, having much in common with exploratory data analysis. We look at the main concepts and algorithms today.

Background: Unsupervised Learning

Definition 1 (Unsupervised Learnng)

With unsupervised learning, the system receives input instances $x_1, x_2, \ldots$ but obtains neither target outputs, nor rewards from its environment. Its goal is to build representations of the input that can be used for decision making, predicting future inputs, efficiently communicating the inputs to another machine, etc. It does this by finding patterns in the data beyond what would be considered pure unstructured noise… [Ghahramani]

Regression and classification are examples of supervised learning because they require labeled *training* data.

We saw *dimensionality reduction* in Topic 8 (Regression 2). Other unsupervised learning techniques include *anomaly detection* and the very “hot” *Generative Adversarial Networks* of deep learning. We look at *clustering* today.

Introduction to Clustering

Definition 2 (Clustering)

Clustering is the operation of grouping objects into a smaller number of clusters (or segments), which have two properties. Firstly, they are not defined in advance by an analyst, but are discovered during the operation, unlike the classes used in classification. Secondly, the clusters combine objects having similar characteristics, which are separated from objects having different characteristics (resulting in *internal homogeneity* and *external heterogeneity* [Tuffery]

Clustering is usually called *segmentation* in marketing studies.

Usually clustering is not an end in itself: it generates insights that are used to motivate and inform other analyses.

The "quality" of a cluster analysis is difficult to determine objectively - there is no equivalent of *recall*, say.

Hierarchical versus Partitional techniques

Hierarchical

- Intermediate steps are interpretable

Partitional

- Interpret the final clustering only

Hierarchical versus Partitional techniques

Hierarchical

- Intermediate steps are interpretable
- More than one clustering generated - see *dendrogram*

Partitional

- Interpret the final clustering only
- Single clustering returned; repeat with different conditions to improve it

Hierarchical versus Partitional techniques

Hierarchical

- Intermediate steps are interpretable
- More than one clustering generated - see *dendrogram*
- Given choice of linkage, algorithm proceeds *deterministically*

Partitional

- Interpret the final clustering only
- Single clustering returned; repeat with different conditions to improve it
- Optimisation by gradient descent, so

Hierarchical versus Partitional techniques

Hierarchical

- Intermediate steps are interpretable
- More than one clustering generated - see *dendrogram*
- Given choice of linkage, algorithm proceeds *deterministically*
    - no concept of starting values

Partitional

- Interpret the final clustering only
- Single clustering returned; repeat with different conditions to improve it
- Optimisation by gradient descent, so
    - result depends on starting values

Hierarchical versus Partitional techniques

Hierarchical

- Intermediate steps are interpretable
- More than one clustering generated - see *dendrogram*
- Given choice of linkage, algorithm proceeds *deterministically*
    - no concept of starting values
    - no concept of local versus global optimum

Partitional

- Interpret the final clustering only
- Single clustering returned; repeat with different conditions to improve it
- Optimisation by gradient descent, so
    - result depends on starting values
    - might find local rather than global optimum

Hierarchical versus Partitional techniques

Hierarchical

- Intermediate steps are interpretable
- More than one clustering generated - see *dendrogram*
- Given choice of linkage, algorithm proceeds *deterministically*
    - no concept of starting values
    - no concept of local versus global optimum
- relatively few parameters to specify

Partitional

- Interpret the final clustering only
- Single clustering returned; repeat with different conditions to improve it
- Optimisation by gradient descent, so
    - result depends on starting values
    - might find local rather than global optimum
- more parameters to specify

Hierarchical versus Partitional techniques

Hierarchical

- Intermediate steps are interpretable
- More than one clustering generated - see *dendrogram*
- Given choice of linkage, algorithm proceeds *deterministically*
  - no concept of starting values
  - no concept of local versus global optimum
- relatively few parameters to specify
- more complex interpretation

Partitional

- Interpret the final clustering only
- Single clustering returned; repeat with different conditions to improve it
- Optimisation by gradient descent, so
  - result depends on starting values
  - might find local rather than global optimum
- more parameters to specify
- interpretation is relatively easy

Outline

1. Introduction 3

2. Distance Measures 8

3. Hierarchical Clustering 13

Distance Measures and their role in clustering

Definition 3 (Distance Measure)

A *distance measure* (c.f., its complement, a *similarity measure*) is a scalar number $d(x_1, x_2)$ that quantifies the degree of agreement between two (usually vector-valued) observations x_1 and x_2. When $x_1 = x_2$, $d(x_1, x_2) = 0$ and $d(x_1, x_2) > 0$ otherwise. It increases as the difference in the observations increases.

By definition, clustering is based on within-cluster homogeneity (measured by small d) versus large d between clusters. Thus choice of distance measure plays a critical part in generating useful clusters.

Distance Measures for numeric data

Definition 4 (Minkowski $p-$norm)

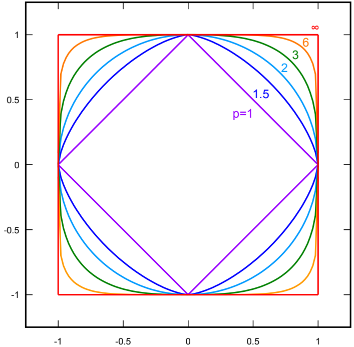
For a real number $1 \leq p < \infty$, the p-norm of $\boldsymbol{x}$ is defined by

$$\|\boldsymbol{x}\|_p \equiv \left(|x_1|^p + |x_2|^p + \ldots + |x_n|^p\right)^{\frac{1}{p}}.$$

The limiting case of $p = \infty$ is defined as

$$\|\boldsymbol{x}\|_\infty \equiv \max\{|x_1|, |x_2|, \ldots, |x_n|\}.$$

See the visualisation of the “unit balls” alongside, for $p = 1, 1.5, 2, 3, 6, \infty$.



Source: wikipedia

Distance Measures for numeric data

Definition 4 (Minkowski $p-$norm)

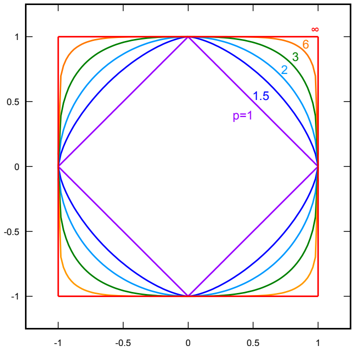
For a real number $1 \leq p < \infty$, the p-norm of $\boldsymbol{x}$ is defined by

$$\|\boldsymbol{x}\|_p \equiv \left(|x_1|^p + |x_2|^p + \ldots + |x_n|^p\right)^{\frac{1}{p}}.$$

The limiting case of $p = \infty$ is defined as

$$\|\boldsymbol{x}\|_\infty \equiv \max\{|x_1|, |x_2|, \ldots, |x_n|\}.$$

See the visualisation of the "unit balls" alongside, for $p = 1, 1.5, 2, 3, 6, \infty$.



Source: wikipedia

The most common norms are when $p = 1, 2, \text{or}, \infty$. Choice of p depends on the application scenario. Can you think of when you would use each?

(Selected) Distance Measures for categorical data

Let $\boldsymbol{x}_1 = [e_{1,1}, e_{1,2}, \dots e_{1,k}]^T$ and $\boldsymbol{x}_2 = [e_{2,1}, e_{2,2}, \dots e_{2,k}]^T$. Furthermore let $e_{1,j}e_{2,j} = 1$ if $e_{1,j} = e_{2,j}$ and $e_{1,j}e_{2,j} = 0$ otherwise. To compute s, the number of matching attributes between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, we can just compute the dot product:

$$s = \boldsymbol{x}_1^T \boldsymbol{x}_2$$

and the number of mismatches is $d = k - s$, where k is the number of attributes in $\boldsymbol{x}$.

(Selected) Distance Measures for categorical data

Let $\boldsymbol{x}_1 = [e_{1,1}, e_{1,2}, \ldots e_{1,k}]^T$ and $\boldsymbol{x}_2 = [e_{2,1}, e_{2,2}, \ldots e_{2,k}]^T$. Furthermore let $e_{1,j}e_{2,j} = 1$ if $e_{1,j} = e_{2,j}$ and $e_{1,j}e_{2,j} = 0$ otherwise. To compute s, the number of matching attributes between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, we can just compute the dot product:

$$s = \boldsymbol{x}_1^T\boldsymbol{x}_2$$

and the number of mismatches is $d = k - s$, where k is the number of attributes in $\boldsymbol{x}$.

Definition 5 (Euclidean distance for categorical observations)

$\|\boldsymbol{x}_1 - \boldsymbol{x}_2\| = \sqrt{\boldsymbol{x}_1^T\boldsymbol{x}_1 - 2\boldsymbol{x}_1^T\boldsymbol{x}_2 + \boldsymbol{x}_2^T\boldsymbol{x}_2} = \sqrt{2(k-s)}$. So the maximum distance occurs when $s = 0$ ($\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ share no attribute values in common, as expected).

(Selected) Distance Measures for categorical data

Let $\boldsymbol{x}_1 = [e_{1,1}, e_{1,2}, \dots e_{1,k}]^T$ and $\boldsymbol{x}_2 = [e_{2,1}, e_{2,2}, \dots e_{2,k}]^T$. Furthermore let $e_{1,j}e_{2,j} = 1$ if $e_{1,j} = e_{2,j}$ and $e_{1,j}e_{2,j} = 0$ otherwise. To compute s, the number of matching attributes between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, we can just compute the dot product:

$$s = \boldsymbol{x}_1^T \boldsymbol{x}_2$$

and the number of mismatches is $d = k - s$, where k is the number of attributes in $\boldsymbol{x}$.

Definition 5 (Euclidean distance for categorical observations)

$\|\boldsymbol{x}_1 - \boldsymbol{x}_2\| = \sqrt{\boldsymbol{x}_1^T \boldsymbol{x}_1 - 2\boldsymbol{x}_1^T \boldsymbol{x}_2 + \boldsymbol{x}_2^T \boldsymbol{x}_2} = \sqrt{2(k - s)}$. So the maximum distance occurs when $s = 0$ ($\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ share no attribute values in common, as expected).

Definition 6 (Hamming Distance)

This is the number of mismatched values $k - s$.

(Selected) Distance Measures for categorical data - ratios

Definition 7 (Cosine similarity)

$$\cos\theta_{1,2} = \frac{\boldsymbol{x}_1^T \boldsymbol{x}_2}{\|\boldsymbol{x}_1\| \|\boldsymbol{x}_2\|} = \frac{s}{\sqrt{k}\sqrt{k}} = \frac{s}{k}.$$

because $\|\boldsymbol{x}\| \equiv \sqrt{\boldsymbol{x}^T \boldsymbol{x}} = \sqrt{k}$.

(Selected) Distance Measures for categorical data - ratios

Definition 7 (Cosine similarity)

$\cos\theta_{1,2} = \frac{\boldsymbol{x}_1^T \boldsymbol{x}_2}{\|\boldsymbol{x}_1\| \|\boldsymbol{x}_2\|} = \frac{s}{\sqrt{k}\sqrt{k}} = \frac{s}{k}$.

because $\|\boldsymbol{x}\| \equiv \sqrt{\boldsymbol{x}^T \boldsymbol{x}} = \sqrt{k}$.

Definition 8 (Jaccard Coefficient)

This is the ratio of the number of matching values s to the number of distinct values that appear in $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, across the d *distinct* attributes of both. It is $J(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{s}{2(k-s)+s} = \frac{s}{2k-s}$.

(Selected) Distance Measures for categorical data - ratios

Definition 7 (Cosine similarity)

$\cos\theta_{1,2} = \frac{\boldsymbol{x}_1^T \boldsymbol{x}_2}{\|\boldsymbol{x}_1\| \|\boldsymbol{x}_2\|} = \frac{s}{\sqrt{k}\sqrt{k}} = \frac{s}{k}$.

because $\|\boldsymbol{x}\| \equiv \sqrt{\boldsymbol{x}^T \boldsymbol{x}} = \sqrt{k}$.

Definition 8 (Jaccard Coefficient)

This is the ratio of the number of matching values s to the number of distinct values that appear in $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, across the d *distinct* attributes of both. It is $J(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{s}{2(k-s)+s} = \frac{s}{2k-s}$.

Note that all these distance measures are functions of s and k, where k is a constant and s is a count of the number of matching attribute values across the two observations in question.

Outline

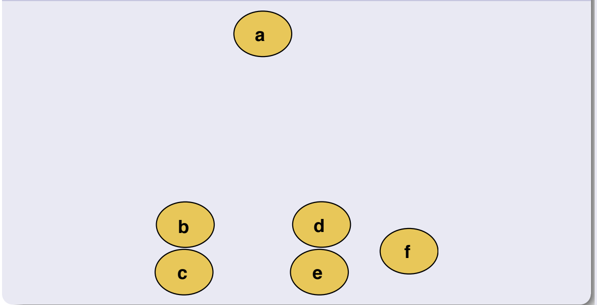
1. Introduction 3

2. Distance Measures 8

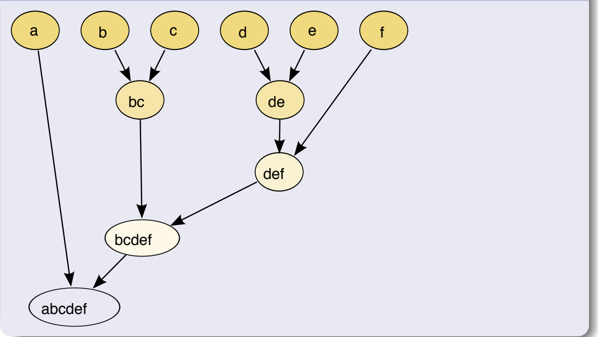
3. Hierarchical Clustering 13

Simple example of data and its dendrogram

Data



Dendrogram



This illustrates how *agglomerative* clustering works. *Divisive* clustering works in the opposite direction, but the (flipped) dendrogram is the same in this case.

Sometimes, the vertical separation between Level i and $i + 1$ of the tree indicates the distance between clusters that are merged at Level $i + 1$.

Diagram Source: wikipedia

Dendrograms

Definition 9 (dendrogram)

The dendrogram shows clusters and their subclusters, in the form of a tree. The root cluster contains all elements; each leaf contains a single element. Clusters are merged in order of their similarity.

Dendrograms

Definition 9 (dendrogram)

The dendrogram shows clusters and their subclusters, in the form of a tree. The root cluster contains all elements; each leaf contains a single element. Clusters are merged in order of their similarity.

The dendrogram makes the internal similarity structure of the data more visible.

Dendrograms

Definition 9 (dendrogram)

The dendrogram shows clusters and their subclusters, in the form of a tree. The root cluster contains all elements; each leaf contains a single element. Clusters are merged in order of their similarity.

The dendrogram makes the internal similarity structure of the data more visible.
Sometimes, the vertical separation between Level i and $i + 1$ of the tree is proportional to the distance between clusters that are merged at Level $i + 1$.

Dendrograms

Definition 9 (dendrogram)

The dendrogram shows clusters and their subclusters, in the form of a tree. The root cluster contains all elements; each leaf contains a single element. Clusters are merged in order of their similarity.

The dendrogram makes the internal similarity structure of the data more visible.
Sometimes, the vertical separation between Level i and $i+1$ of the tree is proportional to the distance between clusters that are merged at Level $i+1$.
An alternative representation is to display the data as a point cloud and to overlay nested clusters over the data.

Overview of Hierarchical Clustering Algorithm

Method (Agglomerative hierarchical clustering (AGNES))

```
Initialise the Cluster set $C = \{\boldsymbol{x}_i\}, i = 1, \ldots, n$;
$q \leftarrow |\{c_i\}| = n$;
Compute the $n \times n$ proximity matrix $D$ where $D_{ij} = d(c_i, c_j)$;
repeat
    Find $i, j$ associated with $\min_{i,j} D$, where $i, j$ are indices of clusters that are nearest each other;
    Create the merged cluster $c_i'$ containing the elements of cluster $c_i$ and $c_j$;
    Record the merge operation so the dendrogram data structure can be built;
    Drop the old $c_j$ cluster since it is not needed any more;
    Delete row $D(j, :)$ and column $D(:, j)$ from $D$
    $q \leftarrow q - 1$;
    Update row $D(i, :)$ and column $D(:, i)$ to compute distance between new cluster $c_i'$ and remaining $q - 2$ clusters;
until $q = 1$ and hence only one cluster remains;
```

Overview of Hierarchical Clustering Algorithm

Method (Agglomerative hierarchical clustering (AGNES))

```
Initialise the Cluster set $C = \{\boldsymbol{x}_i\}, i = 1, \ldots, n$;
$q \leftarrow |\{c_i\}| = n$;
Compute the $n \times n$ proximity matrix $D$ where $D_{ij} = d(c_i, c_j)$;
repeat
    Find $i, j$ associated with $\min_{i,j} D$, where $i, j$ are indices of clusters that are nearest each other;
    Create the merged cluster $c_i'$ containing the elements of cluster $c_i$ and $c_j$;
    Record the merge operation so the dendrogram data structure can be built;
    Drop the old $c_j$ cluster since it is not needed any more;
    Delete row $D(j, :)$ and column $D(:, j)$ from $D$
    $q \leftarrow q - 1$;
    Update row $D(i, :)$ and column $D(:, i)$ to compute distance between new cluster $c_i'$ and remaining $q - 2$ clusters;
until $q = 1$ and hence only one cluster remains;
```

As can be seen, this is a deterministic search algorithm.

However, there is scope for different definitions of the distance function $d(c_i, c_j)$ betwen clusters c_i and c_j.

Distance between clusters: linkage

Earlier, we looked at different ways of computing the *distance between two points*.
For hierarchical clustering, we need to compute the *distance between two clusters*.

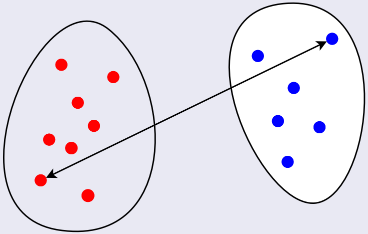
Definition 10 (Linkage function)

For Complete Linkage: $D(X, Y) = \max_{x \in X, y \in Y} d(x, y)$.
For Single Linkage: $D(X, Y) = \min_{x \in X, y \in Y} d(x, y)$.
For Average Linkage: $D(X, Y) = \frac{1}{|X||Y|} \sum_{x \in X, y \in Y} d(x, y)$. This is also known as Unweighted Pair Group Method with Arithmetic Mean (UPGMA) linkage.
For Ward linkage: the initial (single-point) cluster distances are simply the Euclidean distances between the points. The clusters are merged based on a minimum variance criterion. The distance between any point and a merged cluster is calculated using a recursive formula of Lance-Williams type

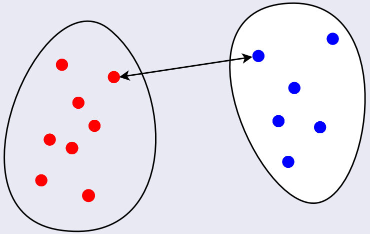
Generally, Complete Linkage and Ward's minimum variance linkage give the most balanced and useful clusters.

Distance between clusters: linkage visualisation

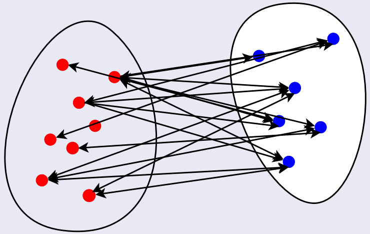
Complete Linkage



Single Linkage



Average Linkage

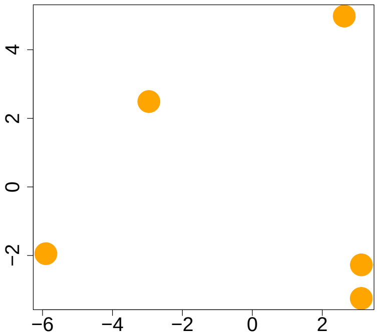


AGNES worked example: setting the scene

Distance Matrix, Step 0

	A	B	C	D	E
A	0				
B	9	0			
C	3	7	0		
D	6	5	9	0	
E	11	10	2	8	0

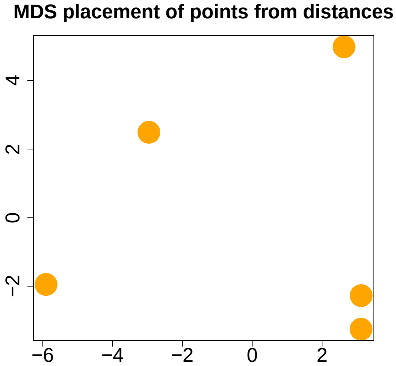
MDS placement of points from distances



AGNES worked example: setting the scene

Distance Matrix, Step 0

	A	B	C	D	E
A	0				
B	9	0			
C	3	7	0		
D	6	5	9	0	
E	11	10	2	8	0



Use of distance matrices

Many algorithms in data mining either start from a distance matrix representation, or need to create it themselves. Reversing the process(from distances to locations) is not unique, but MultiDimensional Scaling often gives an attractive placement (as above, centred on origin).

AGNES worked example: Initial iterations

First clustering: CE, A, B, D

The smallest distance is 2 between C-E. We cluster these points and compute the distance of the remaining points from the CE cluster. Because of single linkage, we store the minimum such distance in the revised table beside.

Distance Matrix, Step 1

	CE	A	B	D
CE	0			
A	3	0		
B	7	9	0	
D	8	6	5	0

AGNES worked example: Initial iterations

First clustering: CE, A, B, D

The smallest distance is 2 between C-E. We cluster these points and compute the distance of the remaining points from the CE cluster. Because of single linkage, we store the minimum such distance in the revised table beside.

Distance Matrix, Step 1

	CE	A	B	D
CE	0			
A	3	0		
B	7	9	0	
D	8	6	5	0

Second clustering: ACE, B, D

The smallest distance is 3 between A and CE. We cluster these points and compute the distance of the remaining points from the ACE cluster. For example d(B,A) = 9, d(B,C) = 7, d(B,E) = 10, so by single linkage d(B,ACE) = 7 as in the revised table beside.

Distance Matrix, Step 2

	ACE	B	D
ACE	0		
B	7	0	
D	6	5	0

Minimum distances so far: 2,3

AGNES worked example: Final iterations

Third clustering: CE, A, B, D

The smallest distance is 5, between B and D, so we create a BD cluster. The new distance matrix is shown alongside. The next step after this would be to merge ACE with BD, creating a single ABCDE cluster. The algorithm ends…

Distance Matrix, Step 3

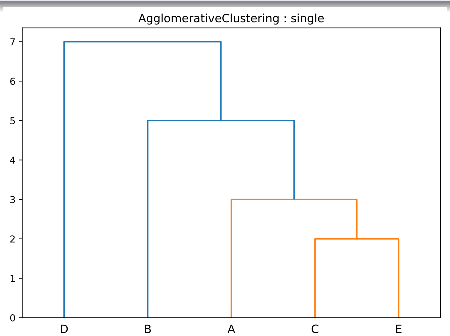
	ACE	BD
ACE	0	
BD	6	0

Minimum distances so far: 2,3,5,(6)

AGNES worked example: Final iterations

Third clustering: CE, A, B, D

The smallest distance is 5, between B and D, so we create a BD cluster. The new distance matrix is shown alongside. The next step after this would be to merge ACE with BD, creating a single ABCDE cluster. The algorithm ends…



Distance Matrix, Step 3

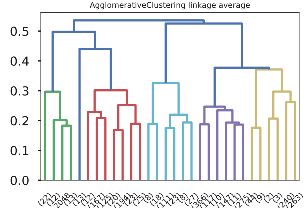
	ACE	BD
ACE	0	
BD	6	0

Minimum distances so far: 2,3,5,(6)

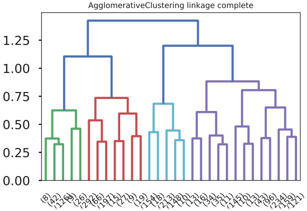
Resulting Dendrogram

The resulting dendrogram summarises the hierarchical clustering. Note that the joins/splits occur at distances 2, 3, 5 and 6, as noted above.

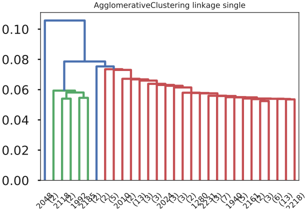
Comparison of Dendrograms



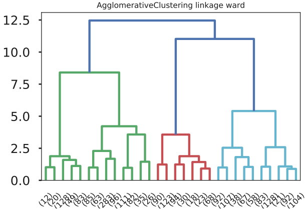
AgglomerativeClustering linkage average



AgglomerativeClustering linkage complete



AgglomerativeClustering linkage single



AgglomerativeClustering linkage ward

As can be seen, the choice of linkage function has a dramatic effect on cluster membership.
The underlying data in this instance appeared to have 6 clusters. Can you see this in these dendrograms?

Uses of hierarchical clustering

- Hierarchical clustering can be very helpful for looking at data at a variety of scales, and hence for seeing hierarchical structure in a data set. This can lead to insights that other techniques, which focus on finding a single cluster mapping, cannot offer.

Uses of hierarchical clustering

- Hierarchical clustering can be very helpful for looking at data at a variety of scales, and hence for seeing hierarchical structure in a data set. This can lead to insights that other techniques, which focus on finding a single cluster mapping, cannot offer.
- Hierarchical clustering offers a rich variety of objective functions (primarily relating to linkage), some of which might suit a specific scenario.

Uses of hierarchical clustering

- Hierarchical clustering can be very helpful for looking at data at a variety of scales, and hence for seeing hierarchical structure in a data set. This can lead to insights that other techniques, which focus on finding a single cluster mapping, cannot offer.
- Hierarchical clustering offers a rich variety of objective functions (primarily relating to linkage), some of which might suit a specific scenario.
- It can be used as a means of estimating parameters for other, perhaps more focused techniques, e.g., to estimate the number of clusters/components in the data.

Uses of hierarchical clustering

- Hierarchical clustering can be very helpful for looking at data at a variety of scales, and hence for seeing hierarchical structure in a data set. This can lead to insights that other techniques, which focus on finding a single cluster mapping, cannot offer.
- Hierarchical clustering offers a rich variety of objective functions (primarily relating to linkage), some of which might suit a specific scenario.
- It can be used as a means of estimating parameters for other, perhaps more focused techniques, e.g., to estimate the number of clusters/components in the data.
- Since hierarchical clustering provides more than one candidate clustering, it can be require more system resources (computation and memory) than other techniques. Thus it might not scale very well.

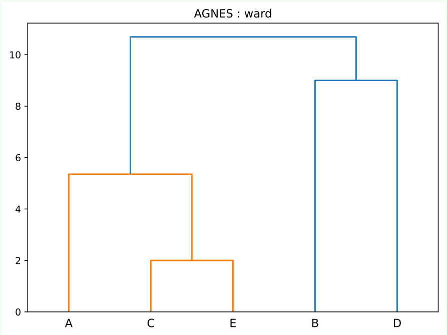
Uses of hierarchical clustering

- Hierarchical clustering can be very helpful for looking at data at a variety of scales, and hence for seeing hierarchical structure in a data set. This can lead to insights that other techniques, which focus on finding a single cluster mapping, cannot offer.
- Hierarchical clustering offers a rich variety of objective functions (primarily relating to linkage), some of which might suit a specific scenario.
- It can be used as a means of estimating parameters for other, perhaps more focused techniques, e.g., to estimate the number of clusters/components in the data.
- Since hierarchical clustering provides more than one candidate clustering, it can be require more system resources (computation and memory) than other techniques. Thus it might not scale very well.
- We have seen agglomerative (bottom-up) clustering. Divisive (top-down) clustering (DIANA) is also available, but has worse scalability.

Hierarchical clustering using scipy

```python
# imports to generate the linkages and plot dendrogram
from scipy.cluster.hierarchy import linkage, dendrogram
link = 'ward'
# Print the condensed distance matrix
print(condensed)
# Cluster the condensed distances using Ward linkage
Z = linkage(condensed, method=link)
# Prepare the plot axes and the plot title
fig, ax = plt.subplots(layout='constrained')
ax.set_title("AGNES : "+link)
# Plot dendrogram showing the hierarchical clusters
labels = ('A','B','C','D','E')
R = dendrogram(Z, ax=ax, orientation='top',
    labels=labels, truncate_mode=None)
plt.show()
```

```
[9, 3, 7, 6, 5, 9, 11, 10, 2, 8]
```


AGNES : ward
A
C
E
B
D