UNIVERSITY OF ABERDEEN

CELEBRATING 525 YEARS 1495 – 2020

ABERDEEN 2040

Hierarchical Clustering

Data Mining & Visualisation

Lecture 18

2025

Today...

- Hierarchical Clustering
- Dendrograms

Types of Clustering

Last lecture, we outlined two broad categories that clustering approaches can fall under:

Partitional Clustering: A division of data objects into non-overlapping subsets (clusters) such that <u>each datapoint is in exactly one subset</u>.

Hierarchical Clustering: A set of nested clusters organized as a hierarchical tree, where <u>each cluster can be a subset of another cluster</u>.

Types of Clustering

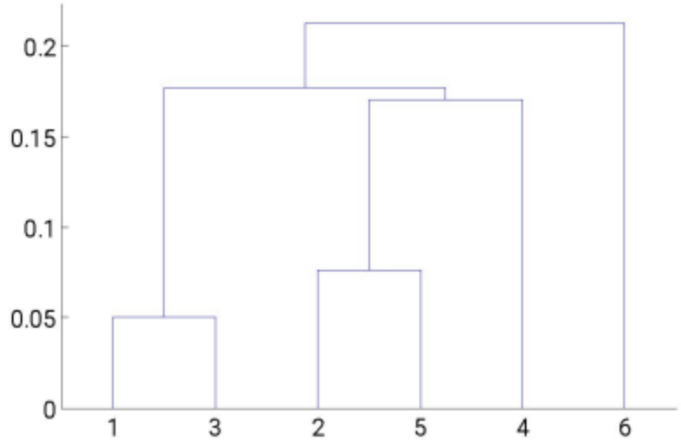
K-Means and K-Means ++ are both examples of **Partitional Clustering**, since each datapoint is assigned to one, and only one, cluster.

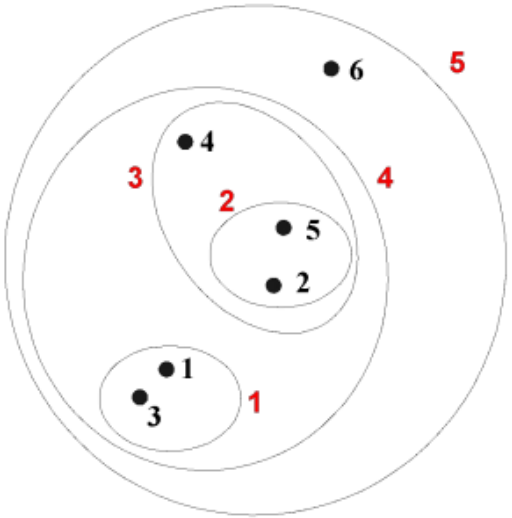
In this lecture, we're going to explore **Hierarchical Clustering**, and how it can be used to have data points assigned to *multiple* nested clusters.

Hierarchical Clustering

Hierarchical Clustering

Hierarchical clustering produces a set of nested clusters organized as a hierarchical tree:





Hierarchical Clustering – Strengths

With hierarchical clustering, we do not have to assume any particular number of clusters, like with K-Means.

We can start off with each datapoint as its own cluster, then start to merge similar clusters together, until just one remains.

As such, any number of desired clusters can be obtained by simply working backward through these cluster merges.

Hierarchical Clustering – Strengths

Another strength of hierarchical clustering is that these hierarchies may correspond to meaningful taxonomies.

In areas such as biological science, there are several such taxonomies (e.g. animal kingdom, phylogeny reconstruction).

By using hierarchical clustering in such taxonomic cases, our findings can more accurately map to the real-world.

Hierarchical Clustering

There are two main types of hierarchical clustering:

- Agglomerative:
  - Start with each datapoint as its own cluster
  - At each step, merge the closest pair of clusters until only one (or K) remains.
- Divisive:
  - Start with one all-inclusive cluster
  - At each step, split a cluster until each cluster contains an individual point (or there are K clusters left)

Defining Cluster Similarity

Like with K-Means, we need a way to evaluate cluster similarity, to determine the order in which clusters merge or split.

Traditional hierarchical algorithms will typically use a *proximity* or *distance matrix*.

These tell us how close, or distant, each cluster is from every other cluster. In this course, we will use a *proximity matrix*.

Defining Cluster Similarity

P*roximity and distance matrices are inversions of each other*.

	a	b	c
a	1.00	0.90	0.10
b	0.90	1.00	0.70
c	0.10	0.70	1.00

Proximity Matrix

	a	b	c
a	0.00	0.10	0.90
b	0.10	0.00	0.30
c	0.90	0.30	0.00

Distance Matrix

Both are valid approaches and both are commonly used. In this course, we will focus on the *proximity matrix* for consistency.

Dendrograms

Dendrograms

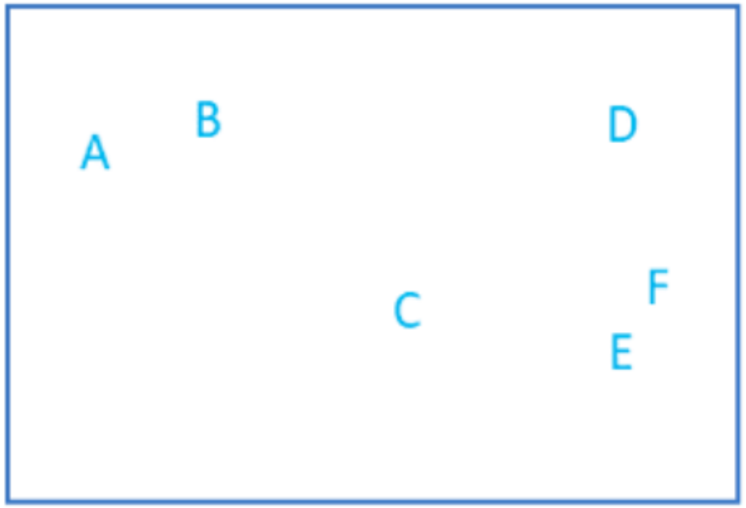
A dendrogram is a diagram that shows the hierarchical relationship between objects.

The main use of a dendrogram is to work out the best way to allocate objects to clusters.

It is most commonly created as an output from hierarchical clustering.

Dendrograms

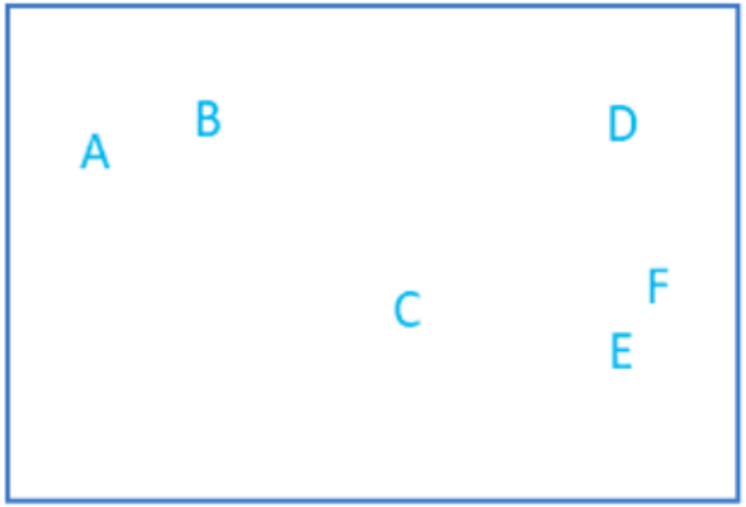
This dendrogram shows the hierarchical clustering of six observations shown on the scatterplot to the left.



Dendrograms

The key to interpreting a dendrogram is to focus on the **height** at which any two objects are joined together.

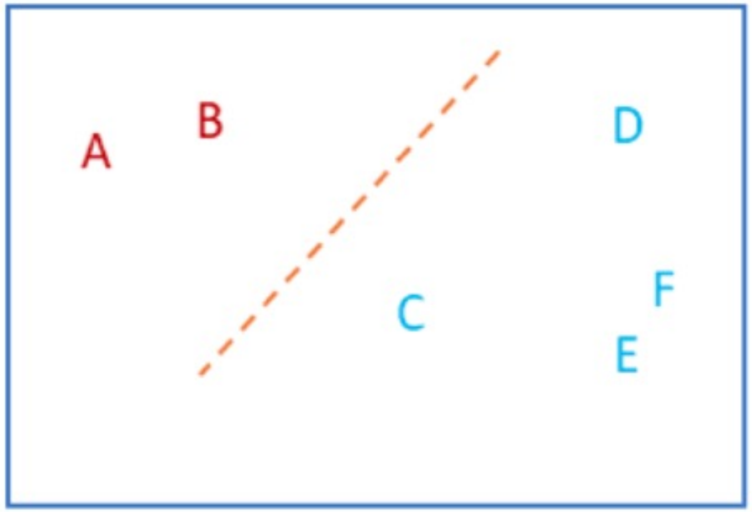
Here, E and F are most similar, as the height of the link that joins them together is the *lowest*. The next two most similar objects are A and B.

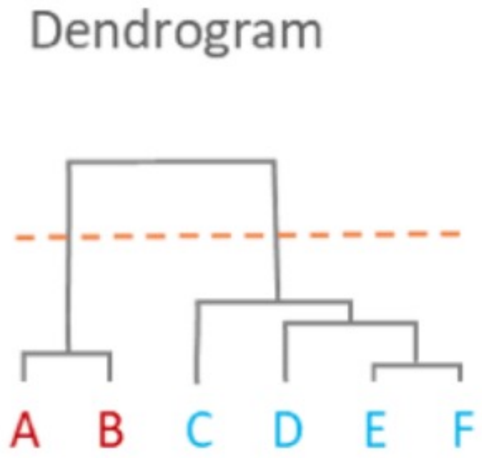


Dendrograms

Observations are allocated to clusters by drawing a horizontal line through the dendrogram.

Observations that are joined together below the line are in clusters.





Agglomerative Hierarchical Clustering

Agglomerative Clustering Algorithm

The agglomerative clustering algorithm is one of the two main types of hierarchical clustering.

Here, we treat every datapoint as its own cluster, and then repeatedly merge the two clusters with the highest proximity.

The key idea is to successively merge the closest clusters.

Agglomerative Clustering Algorithm

The basic algorithm is:

1: Compute the proximity matrix
2: Let each data point be a cluster
3: Repeat:
4:     Merge the two closest clusters
5:     Update the proximity matrix
6: Until only a single cluster remains

Defining Inter-Cluster Similarity

The missing piece of the puzzle so far is how we update our proximity matrix, when two clusters merge.

There are several ways to do this, but we will focus on three:

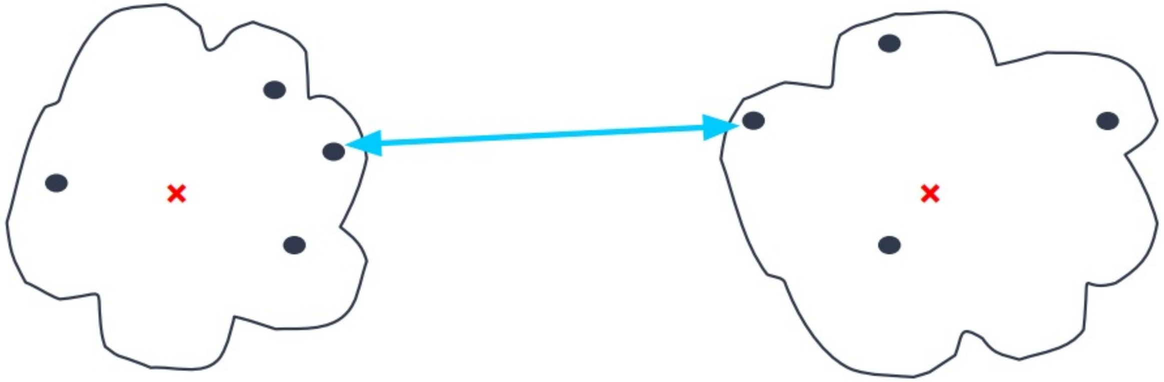
- Single Linkage (MIN)
- Complete Linkage (MAX)
- Group Average Linkage (AVG)

Defining Inter-Cluster Similarity – MIN

The **Single Linkage** method is based on the minimum distance, or the nearest neighbour rule.

At every stage, the distance between two clusters is the distance between their two closest points (**MIN**).

Note: **MIN** distance corresponds to a higher proximity matrix value.

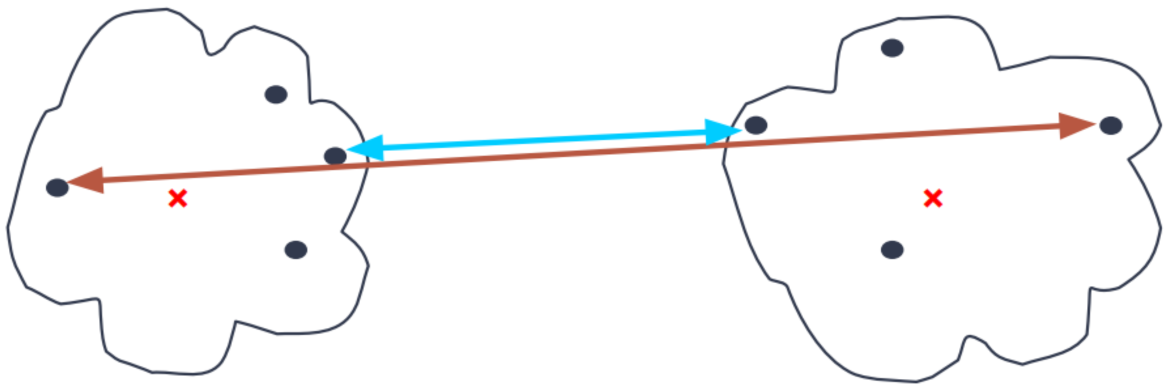


Defining Inter-Cluster Similarity – MAX

The **Complete Linkage** method is based on the maximum distance, or the furthest neighbour approach.

Here, the distance between two clusters is calculated as the distance between their two furthest points (**MAX**).

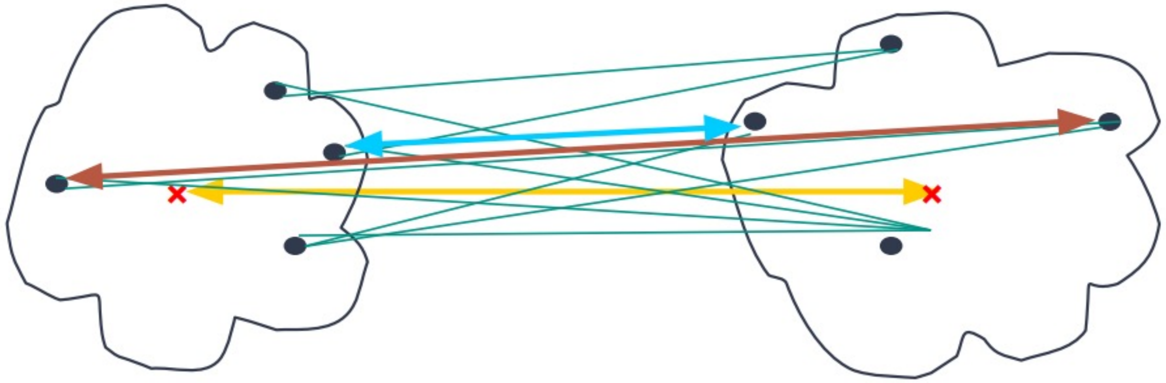
Note: **MAX** distance corresponds to a lower proximity matrix value.



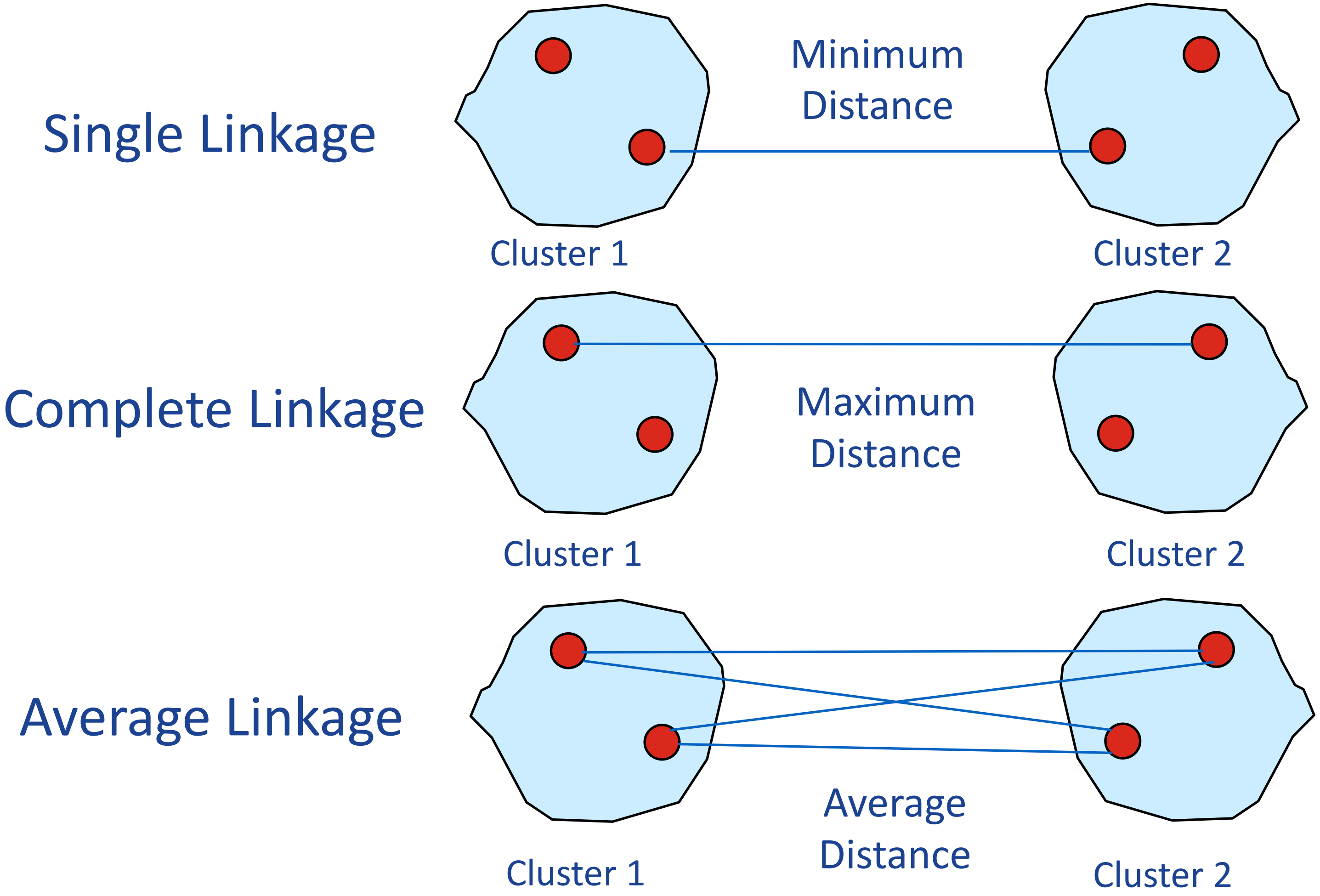
Defining Inter-Cluster Similarity – AVG

The **Group Average Linkage** method works similarly.

The distance between two clusters is defined as <u>the average of the distances between all pairs of objects</u>, where one member of the pair is from each of the clusters (**AVG**).



Defining Inter-Cluster Similarity



Hierarchical Clustering: Step-by-Step

Hierarchical Clustering: Step-by-Step

Let’s say we are given a proximity matrix for data objects (a—e).

Using hierarchical clustering, let’s see how we would cluster these objects using **MIN**, **MAX**, and **AVG**.

Let’s also draw dendrograms for these.

	a	b	c	d	e
a	1.00	0.90	0.10	0.65	0.20
b	0.90	1.00	0.70	0.60	0.50
c	0.10	0.70	1.00	0.40	0.30
d	0.65	0.60	0.40	1.00	0.80
e	0.20	0.50	0.30	0.80	1.00

Hierarchical Clustering: Step-by-Step

Note that we will use *sim(i, j)* to represent similarity between i and j, where i and j are points or clusters.

For instance, sim(a, b) = 0.90.

We will also use ij to represent a cluster containing points i and j.

	a	b	c	d	e
a	1.00	0.90	0.10	0.65	0.20
b	0.90	1.00	0.70	0.60	0.50
c	0.10	0.70	1.00	0.40	0.30
d	0.65	0.60	0.40	1.00	0.80
e	0.20	0.50	0.30	0.80	1.00

Hierarchical Clustering: Using MIN

Let's start with **MIN**.

We initialise each point as its own cluster:
{a}; {b}; {c}; {d}; {e}

	a	b	c	d	e
a	1.00	0.90	0.10	0.65	0.20
b	0.90	1.00	0.70	0.60	0.50
c	0.10	0.70	1.00	0.40	0.30
d	0.65	0.60	0.40	1.00	0.80
e	0.20	0.50	0.30	0.80	1.00

We then find the two clusters that are the **closest** together (highest proximity).

We can see from the proximity matrix, that our two closest clusters are {a} and {b}, since sim(a, b) = 0.90.

As such, we merge them into {a, b} (and keep a record that we merged these first).

Hierarchical Clustering: Using MIN

Now we need to update our proximity matrix.

Since we're using **MIN**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **MIN distance** between any member of the new cluster and each remaining (unchanged) clusters.

Note: MIN distance corresponds to a **higher proximity value**.

For {a, b} and {c}? 0.70 is the MIN distance.

	a	b	c	d	e
a	1.00	0.90	0.10	0.65	0.20
b	0.90	1.00	0.70	0.60	0.50
c	0.10	0.70	1.00	0.40	0.30
d	0.65	0.60	0.40	1.00	0.80
e	0.20	0.50	0.30	0.80	1.00

	a, b	c	d	e
a, b	1.00			
c		1.00	0.40	0.30
d		0.40	1.00	0.80
e		0.30	0.80	1.00

Hierarchical Clustering: Using MIN

Now we need to update our proximity matrix.

Since we're using **MIN**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **MIN distance** between any member of the new cluster and each remaining (unchanged) clusters.

Note: MIN distance corresponds to a **higher proximity value**.

For {a, b} and {d}? 0.65 is the MIN distance.

	a	b	c	d	e
a	1.00	0.90	0.10	0.65	0.20
b	0.90	1.00	0.70	0.60	0.50
c	0.10	0.70	1.00	0.40	0.30
d	0.65	0.60	0.40	1.00	0.80
e	0.20	0.50	0.30	0.80	1.00

	a, b	c	d	e
a, b	1.00	0.70		
c	0.70	1.00	0.40	0.30
d		0.40	1.00	0.80
e		0.30	0.80	1.00

Hierarchical Clustering: Using MIN

Now we need to update our proximity matrix.

Since we're using **MIN**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **MIN distance** between <u>any member of the new cluster</u> and <u>each remaining (unchanged) clusters</u>.

<u>**Note: MIN distance** corresponds to a **higher proximity value**.</u>

For {a, b} and {e}? 0.50 is the MIN distance.

	a	b	c	d	e
a	1.00	0.90	0.10	0.65	0.20
b	0.90	1.00	0.70	0.60	0.50
c	0.10	0.70	1.00	0.40	0.30
d	0.65	0.60	0.40	1.00	0.80
e	0.20	0.50	0.30	0.80	1.00

	a, b	c	d	e
a, b	1.00	0.70	0.65	
c	0.70	1.00	0.40	0.30
d	0.65	0.40	1.00	0.80
e		0.30	0.80	1.00

Hierarchical Clustering: Using MIN

Now we need to update our proximity matrix.

Since we're using **MIN**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **MIN distance** between any member of the new cluster and each remaining (unchanged) clusters.

Note: MIN distance corresponds to a **higher proximity value**.

We have now updated our confusion matrix with {a, b}.

	a	b	c	d	e
a	1.00	0.90	0.10	0.65	0.20
b	0.90	1.00	0.70	0.60	0.50
c	0.10	0.70	1.00	0.40	0.30
d	0.65	0.60	0.40	1.00	0.80
e	0.20	0.50	0.30	0.80	1.00

	a, b	c	d	e
a, b	1.00	0.70	0.65	0.50
c	0.70	1.00	0.40	0.30
d	0.65	0.40	1.00	0.80
e	0.50	0.30	0.80	1.00

Hierarchical Clustering: Using MIN

Now we merge the <u>closest</u> clusters again.

We can see from the proximity matrix, that our two closest clusters are {d} and {e}, since sim(d, e) = 0.80.

As such, we merge them into {d, e}

(and keep a record that we merged these second).

	a, b	c	d	e
a, b	1.00	0.70	0.65	0.50
c	0.70	1.00	0.40	0.30
d	0.65	0.40	1.00	$\boxed{0.80}$
e	0.50	0.30	$\boxed{0.80}$	1.00

Hierarchical Clustering: Using MIN

Now we need to update our proximity matrix.

Since we're using **MIN**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **MIN distance** between <u>any member of the new cluster</u> and <u>each remaining (unchanged) clusters</u>.

<u>**Note: MIN distance** corresponds to a **higher proximity value**</u>.

	a, b	c	d	e
a, b	1.00	0.70	0.65	0.50
c	0.70	1.00	0.40	0.30
d	0.65	0.40	1.00	0.80
e	0.50	0.30	0.80	1.00

	a, b	c	d, e
a, b	1.00	0.70	
c	0.70	1.00	
d, e			1.00

Hierarchical Clustering: Using MIN

Now we need to update our proximity matrix.

Since we're using **MIN**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **MIN distance** between any member of the new cluster and each remaining (unchanged) clusters.

Note: MIN distance corresponds to a **higher proximity value**.

For {d, e} and {a, b}? 0.65 is the MIN distance.

	a, b	c	d	e
a, b	1.00	0.70	0.65	0.50
c	0.70	1.00	0.40	0.30
d	0.65	0.40	1.00	0.80
e	0.50	0.30	0.80	1.00

	a, b	c	d, e
a, b	1.00	0.70	
c	0.70	1.00	
d, e			1.00

Hierarchical Clustering: Using MIN

Now we need to update our proximity matrix.

Since we're using **MIN**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **MIN distance** between any member of the new cluster and each remaining (unchanged) clusters.

Note: MIN distance corresponds to a **higher proximity value**.

For {d, e} and {c}? 0.40 is the MIN distance.

	a, b	c	d	e
a, b	1.00	0.70	0.65	0.50
c	0.70	1.00	0.40	0.30
d	0.65	0.40	1.00	0.80
e	0.50	0.30	0.80	1.00

	a, b	c	d, e
a, b	1.00	0.70	0.65
c	0.70	1.00	
d, e	0.65		1.00

Hierarchical Clustering: Using MIN

Now we need to update our proximity matrix.

Since we're using **MIN**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **MIN distance** between any member of the new cluster and each remaining (unchanged) clusters.

Note: MIN distance corresponds to a **higher proximity value**.

We have now updated our confusion matrix with {d, e}.

	a, b	c	d	e
a, b	1.00	0.70	0.65	0.50
c	0.70	1.00	0.40	0.30
d	0.65	0.40	1.00	0.80
e	0.50	0.30	0.80	1.00

	a, b	c	d, e
a, b	1.00	0.70	0.65
c	0.70	1.00	0.40
d, e	0.65	0.40	1.00

Hierarchical Clustering: Using MIN

Now we merge the <u>closest</u> clusters again.

We can see from the proximity matrix, that our two closest clusters are {a, b} and {c}, since sim(ab, c) = 0.70.

As such, we merge them into {a, b, c}

(and keep a record that we merged these third).

	a, b	c	d, e
a, b	1.00	0.70	0.65
c	0.70	1.00	0.40
d, e	0.65	0.40	1.00

Hierarchical Clustering: Using MIN

And, again, we only have two clusters left to merge at this point, so we merge {a, b, c} with {d, e}.

	a, b	c	d, e
a, b	1.00	0.70	0.65
c	0.70	1.00	0.40
d, e	0.65	0.40	1.00

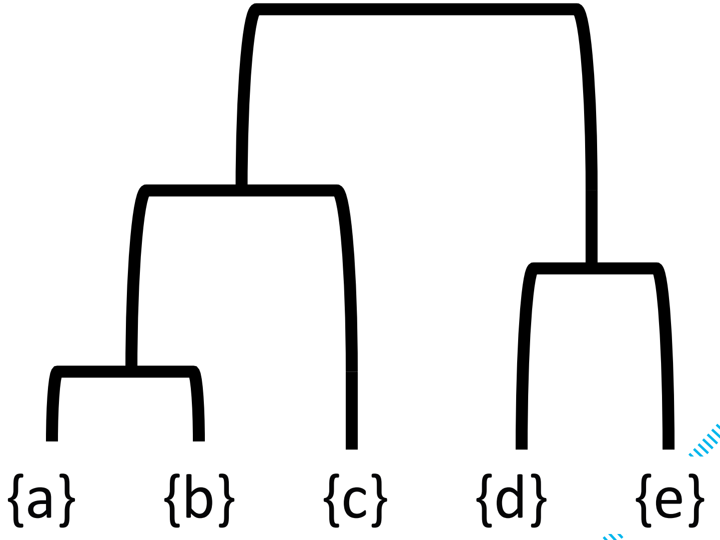
	a, b, c	d, e
a, b, c	1.00	
d, e		1.00

Hierarchical Clustering: Using MIN

So at the end of the MIN process, our merge order was:

1. {a} & {b} → {a, b}
2. {d} & {e} → {d, e}
3. {a, b} & {c} → {a, b, c}
4. {a, b, c} & {d, e} → {a, b, c, d, e}

Therefore, our dendrogram would look like this:



Hierarchical Clustering: Using MAX

	a	b	c	d	e
a	1.00	0.90	0.10	0.65	0.20
b	0.90	1.00	0.70	0.60	0.50
c	0.10	0.70	1.00	0.40	0.30
d	0.65	0.60	0.40	1.00	0.80
e	0.20	0.50	0.30	0.80	1.00

Now let's go back to the question, and focus on the key differences for **MAX**.

We initialise each point as its own cluster:
{a}; {b}; {c}; {d}; {e}

We then find the two clusters that are the **closest** together (highest proximity).

But, again, we see from the proximity matrix, that our two closest clusters are {a} and {b}, since sim(a, b) = 0.90.

As such, we merge them into {a, b} (and keep a record that we merged these first).

Hierarchical Clustering: Using MAX

This time, we'll use MAX to update our proximity matrix.

Since we're using **MAX**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **MAX distance** between any member of the new cluster and each remaining (unchanged) clusters.

Note: MAX distance corresponds to a **lower proximity value**.

	a	b	c	d	e
a	1.00	0.90	0.10	0.65	0.20
b	0.90	1.00	0.70	0.60	0.50
c	0.10	0.70	1.00	0.40	0.30
d	0.65	0.60	0.40	1.00	0.80
e	0.20	0.50	0.30	0.80	1.00

	a, b	c	d	e
a, b	1.00			
c		1.00	0.40	0.30
d		0.40	1.00	0.80
e		0.30	0.80	1.00

Hierarchical Clustering: Using MAX

This time, we'll use MAX to update our proximity matrix.

Since we're using **MAX**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **MAX distance** between any member of the new cluster and each remaining (unchanged) clusters.

Note: MAX distance corresponds to a **lower proximity value**.

For {a, b} and {c}? 0.10 is the MAX distance.

	a	b	c	d	e
a	1.00	0.90	0.10	0.65	0.20
b	0.90	1.00	0.70	0.60	0.50
c	0.10	0.70	1.00	0.40	0.30
d	0.65	0.60	0.40	1.00	0.80
e	0.20	0.50	0.30	0.80	1.00

	a, b	c	d	e
a, b	1.00			
c		1.00	0.40	0.30
d		0.40	1.00	0.80
e		0.30	0.80	1.00

Hierarchical Clustering: Using MAX

This time, we'll use MAX to update our proximity matrix.

Since we're using **MAX**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **MAX distance** between any member of the new cluster and each remaining (unchanged) clusters.

Note: MAX distance corresponds to a **lower proximity value**.

For {a, b} and {d}? 0.60 is the MAX distance.

	a	b	c	d	e
a	1.00	0.90	0.10	0.65	0.20
b	0.90	1.00	0.70	0.60	0.50
c	0.10	0.70	1.00	0.40	0.30
d	$\boxed{0.65}$	$\boxed{0.60}$	0.40	1.00	0.80
e	0.20	0.50	0.30	0.80	1.00

	a, b	c	d	e
a, b	1.00	0.10		
c	0.10	1.00	0.40	0.30
d		0.40	1.00	0.80
e		0.30	0.80	1.00

Hierarchical Clustering: Using MAX

This time, we'll use MAX to update our proximity matrix.

Since we're using **MAX**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **MAX distance** between any member of the new cluster and each remaining (unchanged) clusters.

Note: MAX distance corresponds to a **lower proximity value**.

For {a, b} and {e}? 0.20 is the MAX distance.

	a	b	c	d	e
a	1.00	0.90	0.10	0.65	0.20
b	0.90	1.00	0.70	0.60	0.50
c	0.10	0.70	1.00	0.40	0.30
d	0.65	0.60	0.40	1.00	0.80
e	0.20	0.50	0.30	0.80	1.00

	a, b	c	d	e
a, b	1.00	0.10	0.60	
c	0.10	1.00	0.40	0.30
d	0.60	0.40	1.00	0.80
e		0.30	0.80	1.00

Hierarchical Clustering: Using MAX

This time, we'll use MAX to update our proximity matrix.

Since we're using **MAX**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **MAX distance** between any member of the new cluster and each remaining (unchanged) clusters.

Note: MAX distance corresponds to a **lower proximity value**.

We have now updated our confusion matrix with {a, b}.

	a	b	c	d	e
a	1.00	0.90	0.10	0.65	0.20
b	0.90	1.00	0.70	0.60	0.50
c	0.10	0.70	1.00	0.40	0.30
d	0.65	0.60	0.40	1.00	0.80
e	0.20	0.50	0.30	0.80	1.00

	a, b	c	d	e
a, b	1.00	0.10	0.60	0.20
c	0.10	1.00	0.40	0.30
d	0.60	0.40	1.00	0.80
e	0.20	0.30	0.80	1.00

Hierarchical Clustering: Using MAX

Now we merge the <u>closest</u> clusters again.

And, again, our two closest clusters are {d} and {e}, since sim(d, e) = 0.80.

As such, we merge them into {d, e}
(and keep a record that we merged these second).

	a, b	c	d	e
a, b	1.00	0.10	0.60	0.20
c	0.10	1.00	0.40	0.30
d	0.60	0.40	1.00	0.80
e	0.20	0.30	0.80	1.00

Hierarchical Clustering: Using MAX

This time, we'll use MAX to update our proximity matrix.

Since we're using **MAX**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **MAX distance** between <u>any member of the new cluster</u> and <u>each remaining (unchanged) clusters</u>.

<u>**Note: MAX distance** corresponds to a **lower proximity value**.</u>

	a, b	c	d	e
a, b	1.00	0.10	0.60	0.20
c	0.10	1.00	0.40	0.30
d	0.60	0.40	1.00	0.80
e	0.20	0.30	0.80	1.00

	a, b	c	d, e
a, b	1.00	0.10	
c	0.10	1.00	
d, e			1.00

Hierarchical Clustering: Using MAX

This time, we'll use MAX to update our proximity matrix.

Since we're using **MAX**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **MAX distance** between <u>any member of the new cluster</u> and <u>each remaining (unchanged) clusters</u>.

<u>**Note: MAX distance** corresponds to a **lower proximity value**.</u>

For {d, e} and {a, b}? 0.20 is the MAX distance.

	a, b	c	d	e
a, b	1.00	0.10	0.60	0.20
c	0.10	1.00	0.40	0.30
d	0.60	0.40	1.00	0.80
e	0.20	0.30	0.80	1.00

	a, b	c	d, e
a, b	1.00	0.10	
c	0.10	1.00	
d, e			1.00

Hierarchical Clustering: Using MAX

This time, we'll use MAX to update our proximity matrix.

Since we're using **MAX**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **MAX distance** between any member of the new cluster and each remaining (unchanged) clusters.

Note: MAX distance corresponds to a **lower proximity value**.

For {d, e} and {c}?   0.30 is the MAX distance.

	a, b	c	d	e
a, b	1.00	0.10	0.60	0.20
c	0.10	1.00	0.40	0.30
d	0.60	0.40	1.00	0.80
e	0.20	0.30	0.80	1.00

	a, b	c	d, e
a, b	1.00	0.10	0.20
c	0.10	1.00	
d, e	0.20		1.00

Hierarchical Clustering: Using MAX

This time, we'll use MAX to update our proximity matrix.

Since we're using **MAX**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **MAX distance** between <u>any member of the new cluster</u> and <u>each remaining (unchanged) clusters</u>.

<u>**Note: MAX distance** corresponds to a **lower proximity value**.</u>

We have now updated our confusion matrix with {d, e}.

	a, b	c	d	e
a, b	1.00	0.10	0.60	0.20
c	0.10	1.00	0.40	0.30
d	0.60	0.40	1.00	0.80
e	0.20	0.30	0.80	1.00

	a, b	c	d, e
a, b	1.00	0.10	0.20
c	0.10	1.00	0.30
d, e	0.20	0.30	1.00

Hierarchical Clustering: Using MAX

Now we merge the closest clusters again.

We can see from the proximity matrix, that our two closest clusters are {c} and {d, e}, since sim(c, de) = 0.30.

As such, we merge them into {c, d, e}

(and keep a record that we merged these third).

Note: this is a different ordering than we had for MIN!

	a, b	c	d, e
a, b	1.00	0.10	0.20
c	0.10	1.00	0.30
d, e	0.20	0.30	1.00

Hierarchical Clustering: Using MAX

At this point, we only have two clusters left to merge.

We could repeat the process and find sim(ab, cde), which is 0.10, or we can just merge them.

	a, b	c	d, e
a, b	1.00	0.10	0.20
c	0.10	1.00	0.30
d, e	0.20	0.30	1.00

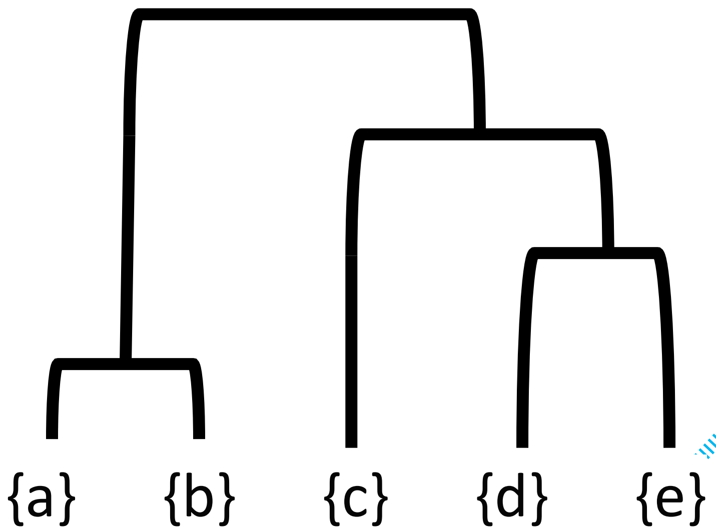
	a, b	c, d, e
a, b	1.00	
c, d, e		1.00

Hierarchical Clustering: Using MAX

So at the end of the MAX process, our merge order was:

1. {a} & {b} → {a, b}
2. {d} & {e} → {d, e}
3. {c} & {d, e} → {c, d, e}
4. {a, b} & {c, d, e} → {a, b, c, d, e}

Therefore, our dendrogram would look like this:



Hierarchical Clustering: Using AVG

AVG is much the same!

Since we're using **AVG**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **AVG** distance between <u>any member of the new cluster</u> and <u>each remaining (unchanged) clusters</u>.

For {a, b} and {c}? 0.40 is the AVG distance.

	a	b	c	d	e
a	1.00	0.90	0.10	0.65	0.20
b	0.90	1.00	0.70	0.60	0.50
c	0.10	0.70	1.00	0.40	0.30
d	0.65	0.60	0.40	1.00	0.80
e	0.20	0.50	0.30	0.80	1.00

	a, b	c	d	e
a, b	1.00			
c		1.00	0.40	0.30
d		0.40	1.00	0.80
e		0.30	0.80	1.00

Hierarchical Clustering: Using AVG

AVG is much the same!

Since we're using **AVG**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **AVG** distance between any member of the new cluster and each remaining (unchanged) clusters.

For {a, b} and {c}? 0.625 is the AVG distance.

	a	b	c	d	e
a	1.00	0.90	0.10	0.65	0.20
b	0.90	1.00	0.70	0.60	0.50
c	0.10	0.70	1.00	0.40	0.30
d	0.65	0.60	0.40	1.00	0.80
e	0.20	0.50	0.30	0.80	1.00

	a, b	c	d	e
a, b	1.00	0.40		
c	0.40	1.00	0.40	0.30
d		0.40	1.00	0.80
e		0.30	0.80	1.00

Hierarchical Clustering: Using AVG

AVG is much the same!

Since we're using **AVG**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **AVG** distance between any member of the new cluster and each remaining (unchanged) clusters.

For {a, b} and {c}? 0.35 is the AVG distance.

	a	b	c	d	e
a	1.00	0.90	0.10	0.65	0.20
b	0.90	1.00	0.70	0.60	0.50
c	0.10	0.70	1.00	0.40	0.30
d	0.65	0.60	0.40	1.00	0.80
e	0.20	0.50	0.30	0.80	1.00

	a, b	c	d	e
a, b	1.00	0.40	0.625	
c	0.40	1.00	0.40	0.30
d	0.625	0.40	1.00	0.80
e		0.30	0.80	1.00

Hierarchical Clustering: Using AVG

AVG is much the same!

Since we're using **AVG**, the distance between the new cluster {a, b} and the old clusters {c}, {d}, and {e} is:

…the **AVG** distance between any member of the new cluster and each remaining (unchanged) clusters.

We have now updated our confusion matrix with {a, b}.

	a	b	c	d	e
a	1.00	0.90	0.10	0.65	0.20
b	0.90	1.00	0.70	0.60	0.50
c	0.10	0.70	1.00	0.40	0.30
d	0.65	0.60	0.40	1.00	0.80
e	0.20	0.50	0.30	0.80	1.00

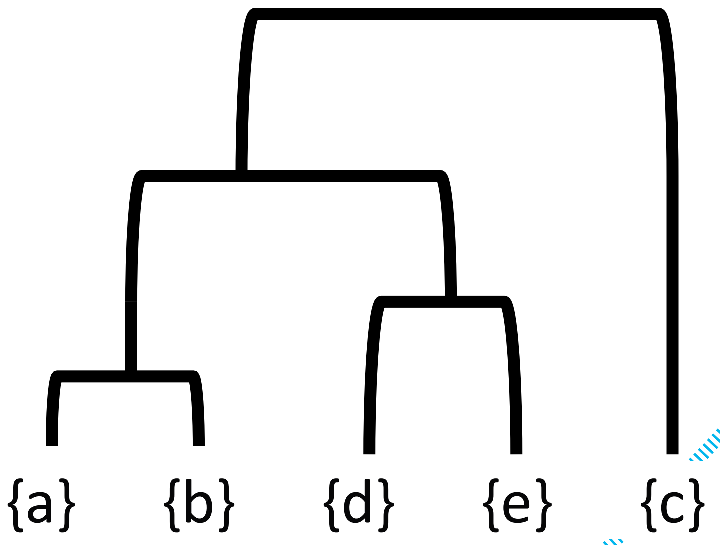
	a, b	c	d	e
a, b	1.00	0.40	0.625	0.35
c	0.40	1.00	0.40	0.30
d	0.625	0.40	1.00	0.80
e	0.35	0.30	0.80	1.00

Hierarchical Clustering: Using AVG

Using the AVG process, our merge order would be:

1. {a} & {b} → {a, b}
2. {d} & {e} → {d, e}
3. {a, b} & {d, e} → {a, b, d, e}
4. {a, b, d, e} & {c} → {a, b, c, d, e}

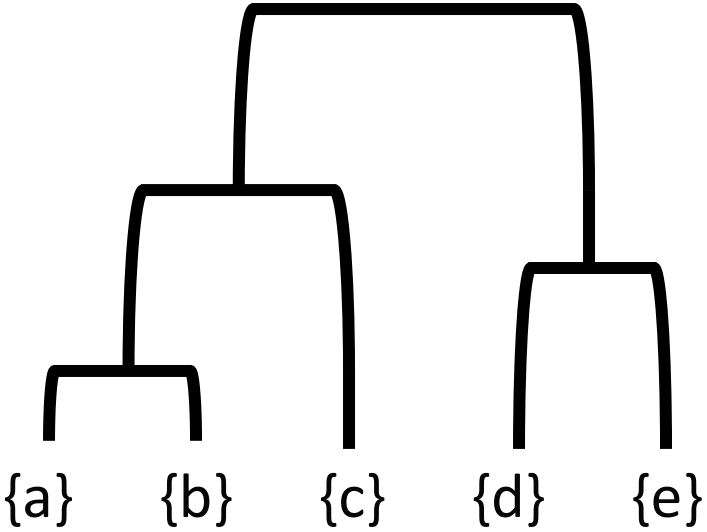
Therefore, our dendrogram would look like this:



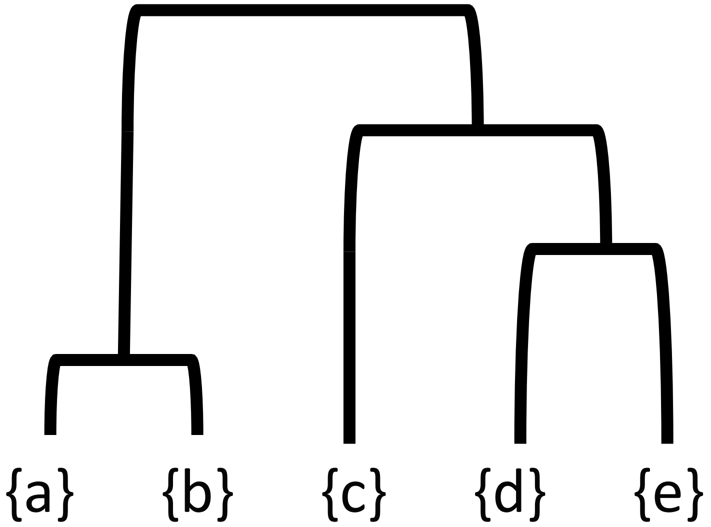
Hierarchical Clustering: MIN vs. MAX vs. AVG

Note that using MIN, using MAX, and using AVG all resulted in different orderings.

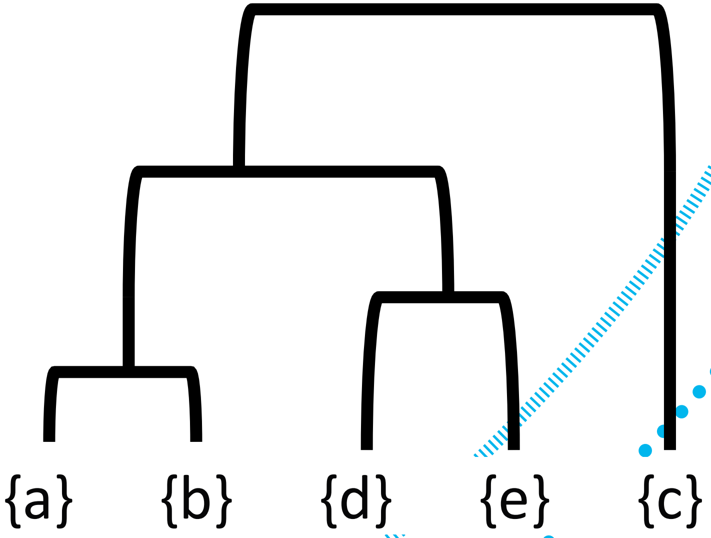
Therefore, the dendograms are all different!



Dendrogram using **MIN**



Dendrogram using **MAX**



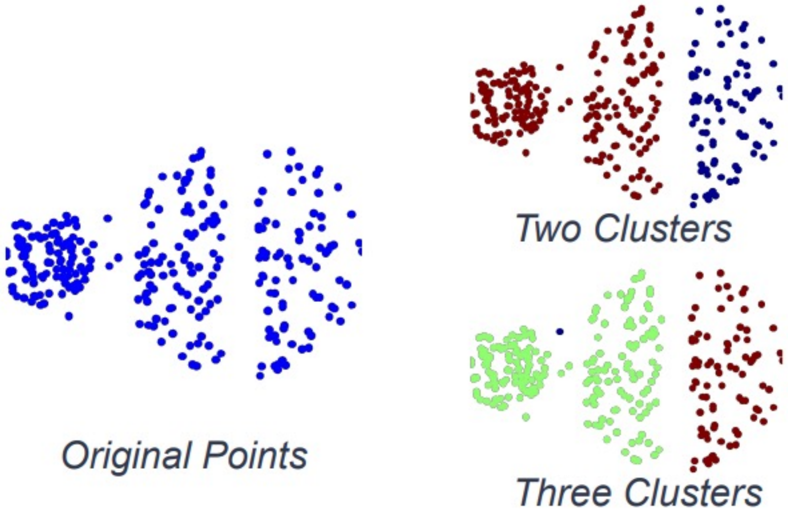
Dendrogram using **AVG**

Summary

Summary: MIN

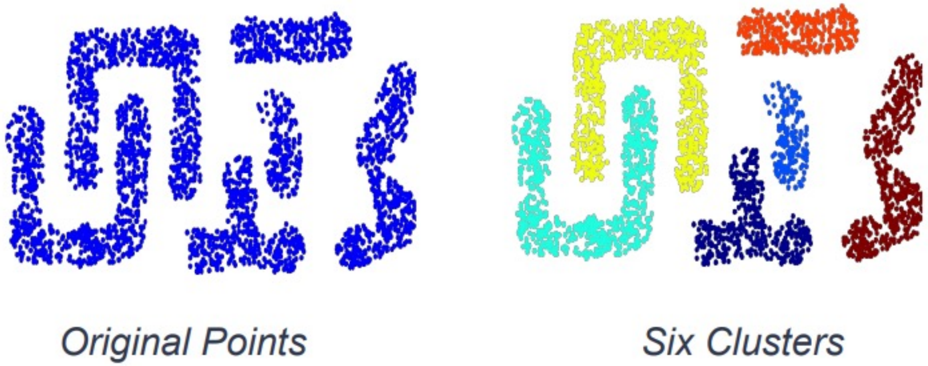
Weaknesses:

- Sensitive to noise
- Biased to chain-like shapes



Strengths:

- Can handle non-elliptical shapes



Summary: MAX

Weaknesses:

- Tends to break large clusters
- Biased towards globular shapes

Strengths:

- Less susceptible to noise