1495 UNIVERSITY OF ABERDEEN

CELEBRATING 525 YEARS 1495 – 2020

ABERDEEN 2040

Revision – Week 3

Data Mining & Visualisation

2025

Today...

Revision questions on:

- K-Means Clustering
- Hierarchical Clustering
- Association Rule Mining

K-Means Clustering

K-Means: Revision

Let's say we have the following six data objects (a—f) in two-dimensional Euclidean space.

Using K-Means clustering, let's see how we would cluster these objects into ***three*** clusters.

Let's say that our initial centroids are *b*, *d*, and *f*.

Point	x_1	x_2
a	2	1
b	2	2
c	4	1
d	4	2
e	7	1
f	7	2

Cluster Centroids:

C1: 2, 2

C2: 4, 2

C3: 7, 2

K-Means: Revision

We start by working out the distance between each point and each cluster centroid.

$$dist(i,j)^2 = (i_{x_1} - j_{x_1})^2 + (i_{x_2} - j_{x_2})^2$$

Let's start with point a (2, 1):

$\text{dist}(a, c1)^2 = (2 - 2)^2 + (1 - 2)^2 = (0)^2 + (-1)^2 = 1$

$\text{dist}(a, c2)^2 = (2 - 4)^2 + (1 - 2)^2 = (-2)^2 + (-1)^2 = 5$

$\text{dist}(a, c3)^2 = (2 - 7)^2 + (1 - 2)^2 = (-5)^2 + (-1)^2 = 26$

In this case, a is closest to C1, so let's allocate a to C1.

Point	x_1	x_2
a	2	1
b	2	2
c	4	1
d	4	2
e	7	1
f	7	2

Cluster Centroids:

C1: 2, 2

C2: 4, 2

C3: 7, 2

K-Means: Revision

We start by working out the distance between each point and each cluster centroid.

$$dist(i,j)^2 = (i_{x_1} - j_{x_1})^2 + (i_{x_2} - j_{x_2})^2$$

We know that point b is our initial cluster centroid for C1.

So let's just allocate b to C1 without calculating!

Point	x_1	x_2
a	2	1
b	2	2
c	4	1
d	4	2
e	7	1
f	7	2

Cluster Centroids:

C1: 2, 2 {a}

C2: 4, 2

C3: 7, 2

K-Means: Revision

We start by working out the distance between each point and each cluster centroid.

$$dist(i,j)^2 = (i_{x_1} - j_{x_1})^2 + (i_{x_2} - j_{x_2})^2$$

Let's move on to point c (4, 1):

$\text{dist}(c, c1)^2 = (4 - 2)^2 + (1 - 2)^2 = (2)^2 + (-1)^2 = 5$

$\text{dist}(c, c2)^2 = (4 - 4)^2 + (1 - 2)^2 = (0)^2 + (-1)^2 = 1$

$\text{dist}(c, c3)^2 = (4 - 7)_2 + (1 - 2)^2 = (-3)^2 + (-1)^2 = 10$

In this case, c is closest to C2, so let's allocate c to C2.

Point	x_1	x_2
a	2	1
b	2	2
c	4	1
d	4	2
e	7	1
f	7	2

Cluster Centroids:

C1: 2, 2 {a, b}

C2: 4, 2

C3: 7, 2

K-Means: Revision

We start by working out the distance between each point and each cluster centroid.

$$dist(i,j)^2 = (i_{x_1} - j_{x_1})^2 + (i_{x_2} - j_{x_2})^2$$

We know that point d is our initial cluster centroid for C2.

So let's just allocate d to C2 without calculating!

Point	x_1	x_2
a	2	1
b	2	2
c	4	1
d	4	2
e	7	1
f	7	2

Cluster Centroids:

C1: 2, 2 {a, b}

C2: 4, 2 {c}

C3: 7, 2

K-Means: Revision

Point	x_1	x_2
a	2	1
b	2	2
c	4	1
d	4	2
e	7	1
f	7	2

We start by working out the distance between each point and each cluster centroid.

$$dist(i,j)^2 = (i_{x_1} - j_{x_1})^2 + (i_{x_2} - j_{x_2})^2$$

Let's move on to point e (7, 1):

Cluster Centroids:

C1: 2, 2 {a, b}

C2: 4, 2 {c, d}

C3: 7, 2

$dist(e, c1)^2 = (7 - 2)^2 + (1 - 2)^2 = (5)^2 + (-1)^2 = 26$

$dist(e, c2)^2 = (7 - 4)^2 + (1 - 2)^2 = (-3)^2 + (-1)^2 = 10$

$dist(e, c3)^2 = (7 - 7)^2 + (1 - 2)^2 = (0)^2 + (-1)^2 = 1$

In this case, e is closest to C3, so let's allocate e to C3.

K-Means: Revision

We start by working out the distance between each point and each cluster centroid.

$$dist(i,j)^2 = (i_{x_1} - j_{x_1})^2 + (i_{x_2} - j_{x_2})^2$$

We know that point f is our initial cluster centroid for C3.

So let's just allocate f to C3 without calculating!

Point	x_1	x_2
a	2	1
b	2	2
c	4	1
d	4	2
e	7	1
f	7	2

Cluster Centroids:

C1: 2, 2 {a, b}

C2: 4, 2 {c, d}

C3: 7, 2 {e}

K-Means: Revision

So after 1 iteration of assignments, our current clusters are:

C1 = {a, b}

C2 = {c, d}

C3 = {e, f}

Point	x_1	x_2
a	2	1
b	2	2
c	4	1
d	4	2
e	7	1
f	7	2

Cluster Centroids:

C1: 2, 2 {a, b}

C2: 4, 2 {c, d}

C3: 7, 2 {e, f}

K-Means: Revision

We then need to recompute our centroids:

C1 centroid = ((2 + 2) / 2, (1 + 2) / 2) = (2, 1.5)

C2 centroid = ((4 + 4) / 2, (1 + 2) / 2) = (4, 1.5)

C3 centroid = ((7 + 7) / 2, (1 + 2) / 2) = (7, 1.5)

Point	x_1	x_2
a	2	1
b	2	2
c	4	1
d	4	2
e	7	1
f	7	2

Cluster Centroids:

C1: 2, 2 {a, b}

C2: 4, 2 {c, d}

C3: 7, 2 {e, f}

K-Means: Revision

Point	x_1	x_2
a	2	1
b	2	2
c	4	1
d	4	2
e	7	1
f	7	2

Now, we need to repeat the process of allocating our points to clusters, based on their new centroids

$$dist(i,j)^2 = (i_{x_1} - j_{x_1})^2 + (i_{x_2} - j_{x_2})^2$$

Let's start with point a (2, 1):

Cluster Centroids:

C1: 2, 1.5

C2: 4, 1.5

C3: 7, 1.5

$\text{dist}(a, c1)^2 = (2 - 2)^2 + (1 - 1.5)^2 = (0)^2 + (-0.5)^2 = 0.25$

$\text{dist}(a, c2)^2 = (2 - 4)^2 + (1 - 1.5)^2 = (-2)^2 + (-0.5)^2 = 4.25$

$\text{dist}(a, c3)^2 = (2 - 7)^2 + (1 - 1.5)^2 = (-5)^2 + (-0.5)^2 = 25.25$

In this case, a is still closest to C1, so let's re-allocate a to C1.

K-Means: Revision

Point	x_1	x_2
a	2	1
b	2	2
c	4	1
d	4	2
e	7	1
f	7	2

Now, we need to repeat the process of allocating our points to clusters, based on their new centroids

$$dist(i,j)^2 = (i_{x_1} - j_{x_1})^2 + (i_{x_2} - j_{x_2})^2$$

Let's move on to point b (2, 2):

Cluster Centroids:

C1: 2, 1.5 {a}

C2: 4, 1.5

C3: 7, 1.5

$\text{dist}(b, c1)^2 = (2 - 2)^2 + (2 - 1.5)^2 = (0)^2 + (0.5)^2 = 0.25$

$\text{dist}(b, c2)^2 = (2 - 4)^2 + (2 - 1.5)^2 = (-2)^2 + (0.5)^2 = 4.25$

$\text{dist}(b, c3)^2 = (2 - 7)^2 + (2 - 1.5)^2 = (-5)^2 + (0.5)^2 = 25.25$

In this case, b is still closest to C1, so let's re-allocate b to C1.

K-Means: Revision

Point	x_1	x_2
a	2	1
b	2	2
c	4	1
d	4	2
e	7	1
f	7	2

Now, we need to repeat the process of allocating our points to clusters, based on their new centroids

$$dist(i,j)^2 = (i_{x_1} - j_{x_1})^2 + (i_{x_2} - j_{x_2})^2$$

Let's move on to point c (4, 1):

Cluster Centroids:

C1: 2, 1.5 {a, b}

C2: 4, 1.5

C3: 7, 1.5

$dist(c, c1)^2 = (4 - 2)^2 + (1 - 1.5)^2 = (2)^2 + (-0.5)^2 = 4.25$

$dist(c, c2)^2 = (4 - 4)^2 + (1 - 1.5)^2 = (0)^2 + (-0.5)^2 = 0.25$

$dist(c, c3)^2 = (4 - 7)^2 + (1 - 1.5)^2 = (-3)^2 + (-0.5)^2 = 9.25$

In this case, c is still closest to C2, so let's re-allocate c to C2.

K-Means: Revision

Point	x_1	x_2
a	2	1
b	2	2
c	4	1
d	4	2
e	7	1
f	7	2

Now, we need to repeat the process of allocating our points to clusters, based on their new centroids

$$dist(i, j)^2 = (i_{x_1} - j_{x_1})^2 + (i_{x_2} - j_{x_2})^2$$

Cluster Centroids:

C1: 2, 1.5 {a, b}

C2: 4, 1.5 {c}

C3: 7, 1.5

Let's move on to point d (4, 2):

dist(d, c1)^2 = $(4 - 2)^2 + (2 – 1.5)^2$ = $(2)^2 + (0.5)^2$ = 4.25

dist(d, c2)^2 = $(4 - 4)^2 + (2 – 1.5)^2$ = $(0)^2 + (0.5)^2$ = 0.25

dist(d, c3)^2 = $(4 - 7)^2 + (2 – 1.5)^2$ = $(-3)^2 + (0.5)^2$ = 9.25

In this case, d is still closest to C2, so let's re-allocate d to C2.

K-Means: Revision

Point	x_1	x_2
a	2	1
b	2	2
c	4	1
d	4	2
e	7	1
f	7	2

Now, we need to repeat the process of allocating our points to clusters, based on their new centroids

$$dist(i,j)^2 = (i_{x_1} - j_{x_1})^2 + (i_{x_2} - j_{x_2})^2$$

Let's move on to point e (7, 1):

Cluster Centroids:

C1: 2, 1.5 {a, b}

C2: 4, 1.5 {c, d}

C3: 7, 1.5

$dist(e, c1)^2 = (7 - 2)^2 + (1 – 1.5)^2 = (5)^2 + (-0.5)^2 = 25.25$

$dist(e, c2)^2 = (7 - 4)^2 + (1 – 1.5)^2 = (3)^2 + (-0.5)^2 = 9.25$

$dist(e, c3)^2 = (7 - 7)^2 + (1 – 1.5)^2 = (0)^2 + (-0.5)^2 = 0.25$

In this case, e is still closest to C3, so let's re-allocate e to C3.

K-Means: Revision

Point	x_1	x_2
a	2	1
b	2	2
c	4	1
d	4	2
e	7	1
f	7	2

Now, we need to repeat the process of allocating our points to clusters, based on their new centroids

$$dist(i,j)^2 = (i_{x_1} - j_{x_1})^2 + (i_{x_2} - j_{x_2})^2$$

Let's move on to point f (7, 2):

Cluster Centroids:

C1: 2, 1.5 {a, b}

C2: 4, 1.5 {c, d}

C3: 7, 1.5 {e}

$dist(f, c1)^2 = (7 - 2)^2 + (2 - 1.5)^2 = (5)^2 + (0.5)^2 = 25.25$

$dist(f, c2)^2 = (7 - 4)^2 + (2 - 1.5)^2 = (3)^2 + (0.5)^2 = 9.25$

$dist(f, c3)^2 = (7 - 7)^2 + (2 - 1.5)^2 = (0)^2 + (0.5)^2 = 0.25$

In this case, f is still closest to C3, so let's re-allocate f to C3.

K-Means: Revision

So after 2 iterations of assignments, our current clusters are:

C1 = {a, b}

C2 = {c, d}

C3 = {e, f}

Point	x_1	x_2
a	2	1
b	2	2
c	4	1
d	4	2
e	7	1
f	7	2

Cluster Centroids:

C1: 2, 1.5 {a, b}

C2: 4, 1.5 {c, d}

C3: 7, 1.5 {e, f}

K-Means: Revision

Note that, at this point, our cluster allocations have not changed since the last iteration.

Therefore, re-calculating the centroids will result in the same coordinates.

At this point, we have reached convergence, and our clusters will not change anymore.

Point	x_1	x_2
a	2	1
b	2	2
c	4	1
d	4	2
e	7	1
f	7	2

Cluster Centroids:

C1: 2, 1.5 {a, b}

C2: 4, 1.5 {c, d}

C3: 7, 1.5 {e, f}

K-Means: Revision

Last step we need to do is to calculate the SSE.

Our clusters: C1 = {a, b}; C2 = {c, d}; C3 = {e, f}

And we've already calculated dist^2 for each of these values. <u>This is our squared error!</u>

To calculate SSE for each cluster:

$$\text{SSE}_{C_k} = \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$

Point	x_1	x_2
a	2	1
b	2	2
c	4	1
d	4	2
e	7	1
f	7	2

Cluster Centroids:

C1: 2, 1.5 {a, b}

C2: 4, 1.5 {c, d}

C3: 7, 1.5 {e, f}

K-Means: Revision

Point	x_1	x_2
a	2	1
b	2	2
c	4	1
d	4	2
e	7	1
f	7	2

Last step we need to do is to calculate the SSE.

$$\mathrm{SSE}_{C_k} = \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$

$SSE_{c1} = dist(a, c1)^2 + dist(b, c1)^2 = .25 + .25 = .5$

$SSE_{c2} = dist(c, c2)^2 + dist(d, c2)^2 = .25 + .25 = .5$

$SSE_{c3} = dist(e, c3)^2 + dist(f, c3)^2 = .25 + .25 = .5$

Cluster Centroids:

C1: 2, 1.5 {a, b}

C2: 4, 1.5 {c, d}

C3: 7, 1.5 {e, f}

Our overall SSE = SSE_{c1} + SSE_{c2} + SSE_{c3} = 1.5

Hierarchical Clustering

Hierarchical Clustering: Revision

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** and **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

Let’s also draw dendrograms for these.

Hierarchical Clustering: Revision

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 i, j 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 any member of the new cluster and each remaining (unchanged) clusters.

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

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 closest 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	0.80
e	0.50	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**.

	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 <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 {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 <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>

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 closest 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

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

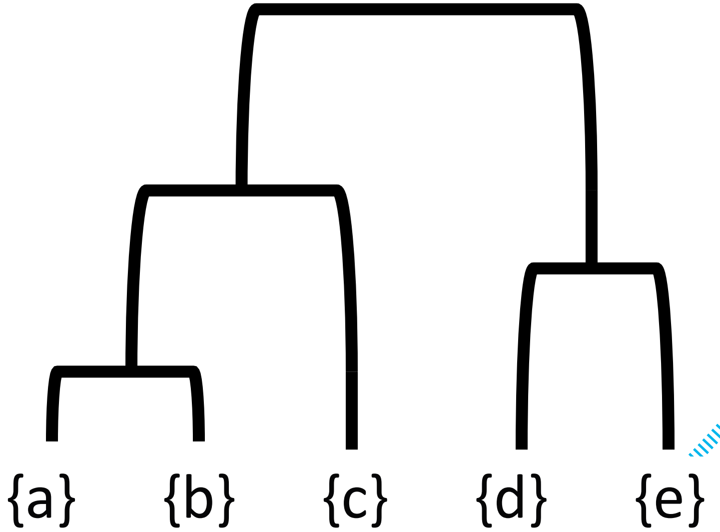
	a, b, c	d, e
a, b, c	1.00	
d, e		1.00

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

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

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).

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 <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	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	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		
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 <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 {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 closest 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 any member of the new cluster and each remaining (unchanged) clusters.

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

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	$\boxed{0.30}$
d, e	0.20	$\boxed{0.30}$	1.00

Hierarchical Clustering: Using MAX

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

	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

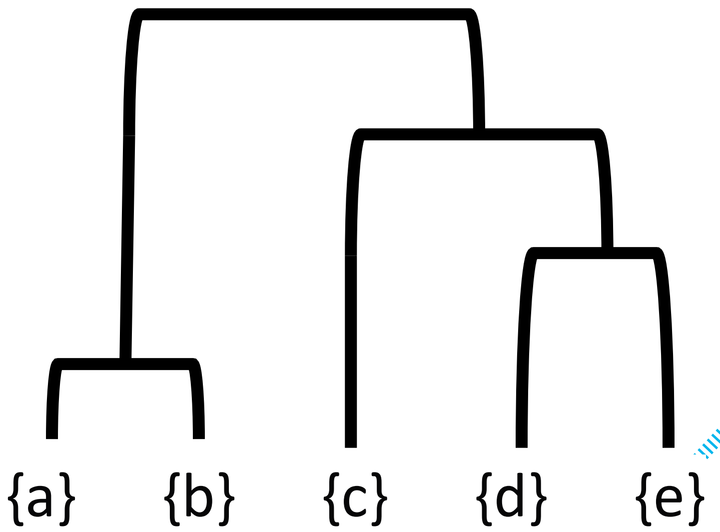
	a, b	c, d, e
a, b	1.00	
c, d, e		1.00

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:



Association Rule Mining

Association Rule Mining: Revision

Consider the following transactions involving five items. You have been asked to produce association rules for these items using the Apriori algorithm:

Transaction ID	Item List
1	Apple, Broccoli, Durian, Eggplant
2	Broccoli, Carrot, Durian
3	Apple, Broccoli, Durian, Eggplant
4	Apple, Carrot, Durian, Eggplant

Association Rule Mining: Revision

ID	Item List
1	Apple, Broccoli, Durian, Eggplant
2	Broccoli, Carrot, Durian
3	Apple, Broccoli, Durian, Eggplant
4	Apple, Carrot, Durian, Eggplant

i) Using a minimum support of 0.75, generate the frequent itemsets for the above data showing clearly the application of Apriori principle in pruning infrequent itemsets.

ii) Using a minimum confidence of 0.75, generate the association rules generated from the frequent itemsets computed in (i) showing clearly the application of Apriori principle in pruning low confidence rules.

Association Rule Mining: Revision

ID	Item List
1	Apple, Broccoli, Durian, Eggplant
2	Broccoli, Carrot, Durian
3	Apple, Broccoli, Durian, Eggplant
4	Apple, Carrot, Durian, Eggplant

i) **Using a minimum support of 0.75, generate the frequent itemsets for the above data showing clearly the application of Apriori principle in pruning infrequent itemsets.**

ii) Using a minimum confidence of 0.75, generate the association rules generated from the frequent itemsets computed in (i) showing clearly the application of Apriori principle in pruning low confidence rules.

Association Rule Mining:

ID	Item List	
1	Apple, Broccoli, Durian, Eggplant	←
2	Broccoli, Carrot, Durian	
3	Apple, Broccoli, Durian, Eggplant	←
4	Apple, Carrot, Durian, Eggplant	←

minsup = 0.75

First step, let's identify *frequent* 1-itemsets

$$Support = \frac{X \cap Y.count}{n}$$

{Apple}: 3 / 4 = 0.75 ✅
{Broccoli}:
{Carrot}:
{Durian}:
{Eggplant}:

We see that Apple appears in 3 transactions ($X.count$)

We also see that we have 4 total transactions (n)

Therefore, the support of our {Apple} 1-itemset is 0.75.

Since Support ≥ minsup, {Apple} is a frequent 1-itemset.

Association Rule Mining:

ID	Item List
1	Apple, Broccoli, Durian, Eggplant
2	Broccoli, Carrot, Durian
3	Apple, Broccoli, Durian, Eggplant
4	Apple, Carrot, Durian, Eggplant

minsup = 0.75

First step, let's identify *frequent* 1-itemsets

$$Support = \frac{X \cap Y.count}{n}$$

{Apple}: 3 / 4 = 0.75 ✅
{Broccoli}: 3 / 4 = 0.75 ✅
{Carrot}: 2 / 4 = 0.5 ❌
{Durian}: 4 / 4 = 1.0 ✅
{Eggplant}: 3 / 4 = 0.75 ✅

After checking the support of each of our 1-itemsets, we can see that 4 of our itemsets are frequent.

Carrot does not reach the minimum level of support. Therefore, we use the Apriori principle to prune it.

Association Rule Mining:

ID	Item List
1	Apple, Broccoli, Durian, Eggplant
2	Broccoli, Carrot, Durian
3	Apple, Broccoli, Durian, Eggplant
4	Apple, Carrot, Durian, Eggplant

minsup = 0.75

So far, our *frequent* itemsets are:

{Apple}, {Broccoli}, {Durian}, {Eggplant}

From these, our *candidate* 2- itemsets are:

{Apple, Broccoli}, {Apple, Durian}, {Apple, Eggplant},
{Broccoli, Durian}, {Broccoli, Eggplant}, {Durian, Eggplant}

Note that we do not make any candidate 2-itemsets using Carrot, since the Apriori principle tells us that all supersets containing Carrot will also be infrequent!

Association Rule Mining:

ID	Item List
1	Apple, Broccoli, Durian, Eggplant
2	Broccoli, Carrot, Durian
3	Apple, Broccoli, Durian, Eggplant
4	Apple, Carrot, Durian, Eggplant

Next, lets go back and count the number of transactions that include our *candidate* 2-itemsets:

minsup = 0.75

{A, B}: 2 / 4 = 0.5 ❌
{A, D}: 3 / 4 = 0.75 ✅
{A, E}: 3 / 4 = 0.75 ✅
{B, D}: 3 / 4 = 0.75 ✅
{B, E}: 2 / 4 = 0.5 ❌
{D, E}: 3 / 4 = 0.75 ✅

After checking the support of each of our 2-itemsets, we can see that 4 of our itemsets are frequent.

{Apple, Broccoli} and {Broccoli, Eggplant} do not reach the minimum level of support. Therefore, we use the Apriori principle to prune them.

Association Rule Mining:

ID	Item List
1	Apple, Broccoli, Durian, Eggplant
2	Broccoli, Carrot, Durian
3	Apple, Broccoli, Durian, Eggplant
4	Apple, Carrot, Durian, Eggplant

minsup = 0.75

So far, our *frequent* itemsets are:

{A}, {B}, {D}, {E}, {A, D}, {A, E}, {B, D}, {D, E}

From these, our *candidate* 3- itemsets are:

{A, D, E}

From our frequent 2-itemsets, we can only produce one candidate 3-itemset, since the Apriori principle tells us that any other combinations will be infrequent!

E.g. since {A, B} is infrequent, we know that {A, B, D} will also be infrequent.

Association Rule Mining:

ID	Item List
1	Apple, Broccoli, Durian, Eggplant
2	Broccoli, Carrot, Durian
3	Apple, Broccoli, Durian, Eggplant
4	Apple, Carrot, Durian, Eggplant

minsup = 0.75

So, lets go back and count the number of transactions that include our *candidate* 3-itemset:

{A, D, E}: 3 / 4 = 0.75 ✅

Association Rule Mining:

ID	Item List
1	Apple, Broccoli, Durian, Eggplant
2	Broccoli, Carrot, Durian
3	Apple, Broccoli, Durian, Eggplant
4	Apple, Carrot, Durian, Eggplant

minsup = 0.75

So far, our *frequent* itemsets are:

{A}, {B}, {D}, {E}, {A, D}, {A, E}, {B, D}, {D, E}, {A, D, E}

However, since we only have one frequent 3- itemset, we cannot produce any candidate 4-itemsets.

Therefore, we are done finding frequent itemsets!

Association Rule Mining: Revision

ID	Item List
1	Apple, Broccoli, Durian, Eggplant
2	Broccoli, Carrot, Durian
3	Apple, Broccoli, Durian, Eggplant
4	Apple, Carrot, Durian, Eggplant

i) Using a minimum support of 0.75, generate the frequent itemsets for the above data showing clearly the application of Apriori principle in pruning infrequent itemsets.

ii) Using a minimum confidence of 0.75, generate the association rules generated from the frequent itemsets computed in (i) showing clearly the application of Apriori principle in pruning low confidence rules.

Association Rule Mining:

ID	Item List
1	Apple, Broccoli, Durian, Eggplant
2	Broccoli, Carrot, Durian
3	Apple, Broccoli, Durian, Eggplant
4	Apple, Carrot, Durian, Eggplant

minsup = 0.75

So far, our *frequent* itemsets are:

{A}, {B}, {D}, {E}, {A, D}, {A, E}, {B, D}, {D, E}, {A, D, E}

Let's now generate some *candidate* association rules.

Association Rule Mining: Revision

Itemset	Support
A	3 / 4 = 0.75
B	3 / 4 = 0.75
D	4 / 4 = 1
E	3 / 4 = 0.75
A, D	3 / 4 = 0.75
A, E	3 / 4 = 0.75
B, D	3 / 4 = 0.75
D, E	3 / 4 = 0.75
A, D, E	3 / 4 = 0.75

Minconf = 0.75

So far, our *frequent* itemsets are:

{A}, {B}, {D}, {E}, {A, D}, {A, E}, {B, D}, {D, E}, {A, D, E}

Our *candidate* association rules are:

$\{A \Longrightarrow D\}$	$\{D \Longrightarrow B\}$	$\{D \Longrightarrow A, E\}$
$\{D \Longrightarrow A\}$	$\{D \Longrightarrow E\}$	$\{A, E \Longrightarrow D\}$
$\{A \Longrightarrow E\}$	$\{E \Longrightarrow D\}$	$\{E \Longrightarrow A, D\}$
$\{E \Longrightarrow A\}$	$\{A \Longrightarrow D, E\}$	$\{A, D \Longrightarrow E\}$
$\{B \Longrightarrow D\}$	$\{D, E \Longrightarrow A\}$	

Association Rule Mining: Revision

Itemset	Support
A	3 / 4 = 0.75
B	3 / 4 = 0.75
D	4 / 4 = 1
E	3 / 4 = 0.75
A, D	3 / 4 = 0.75
A, E	3 / 4 = 0.75
B, D	3 / 4 = 0.75
D, E	3 / 4 = 0.75
A, D, E	3 / 4 = 0.75

Our *candidate* association rules are:

$\{A \Longrightarrow D\}$	0.75 / 0.75 = 1
$\{D \Longrightarrow A\}$	0.75 / 1 = 0.75
$\{A \Longrightarrow E\}$	0.75 / 0.75 = 1
$\{E \Longrightarrow A\}$	0.75 / 0.75 = 1
$\{B \Longrightarrow D\}$	0.75 / 0.75 = 1
$\{D \Longrightarrow B\}$	0.75 / 1 = 0.75
$\{D \Longrightarrow E\}$	0.75 / 1 = 0.75
$\{E \Longrightarrow D\}$	0.75 / 0.75 = 1
$\{A \Longrightarrow D, E\}$	0.75 / 0.75 = 1
$\{D, E \Longrightarrow A\}$	0.75 / 0.75 = 1
$\{D \Longrightarrow A, E\}$	0.75 / 1 = 0.75
$\{A, E \Longrightarrow D\}$	0.75 / 0.75 = 1
$\{E \Longrightarrow A, D\}$	0.75 / 0.75 = 1
$\{A, D \Longrightarrow E\}$	0.75 / 0.75 = 1

Minconf = 0.75

$$Confidence(X \Rightarrow Y) = \frac{Supp(X \cap Y)}{Supp(X)}$$

Association Rule Mining: Revision

Itemset	Support
A	3 / 4 = 0.75
B	3 / 4 = 0.75
D	4 / 4 = 1
E	3 / 4 = 0.75
A, D	3 / 4 = 0.75
A, E	3 / 4 = 0.75
B, D	3 / 4 = 0.75
D, E	3 / 4 = 0.75
A, D, E	3 / 4 = 0.75

Minconf = 0.75

In this case, all of our *candidate* association rules are valid, with confidence ≥ minconf.

Our *final* association rules are:

$\{A \Longrightarrow D\}$	$\{D \Longrightarrow B\}$	$\{D \Longrightarrow A, E\}$
$\{D \Longrightarrow A\}$	$\{D \Longrightarrow E\}$	$\{A, E \Longrightarrow D\}$
$\{A \Longrightarrow E\}$	$\{E \Longrightarrow D\}$	$\{E \Longrightarrow A, D\}$
$\{E \Longrightarrow A\}$	$\{A \Longrightarrow D, E\}$	$\{A, D \Longrightarrow E\}$
$\{B \Longrightarrow D\}$	$\{D, E \Longrightarrow A\}$	