COSC 3337 : Data Science I

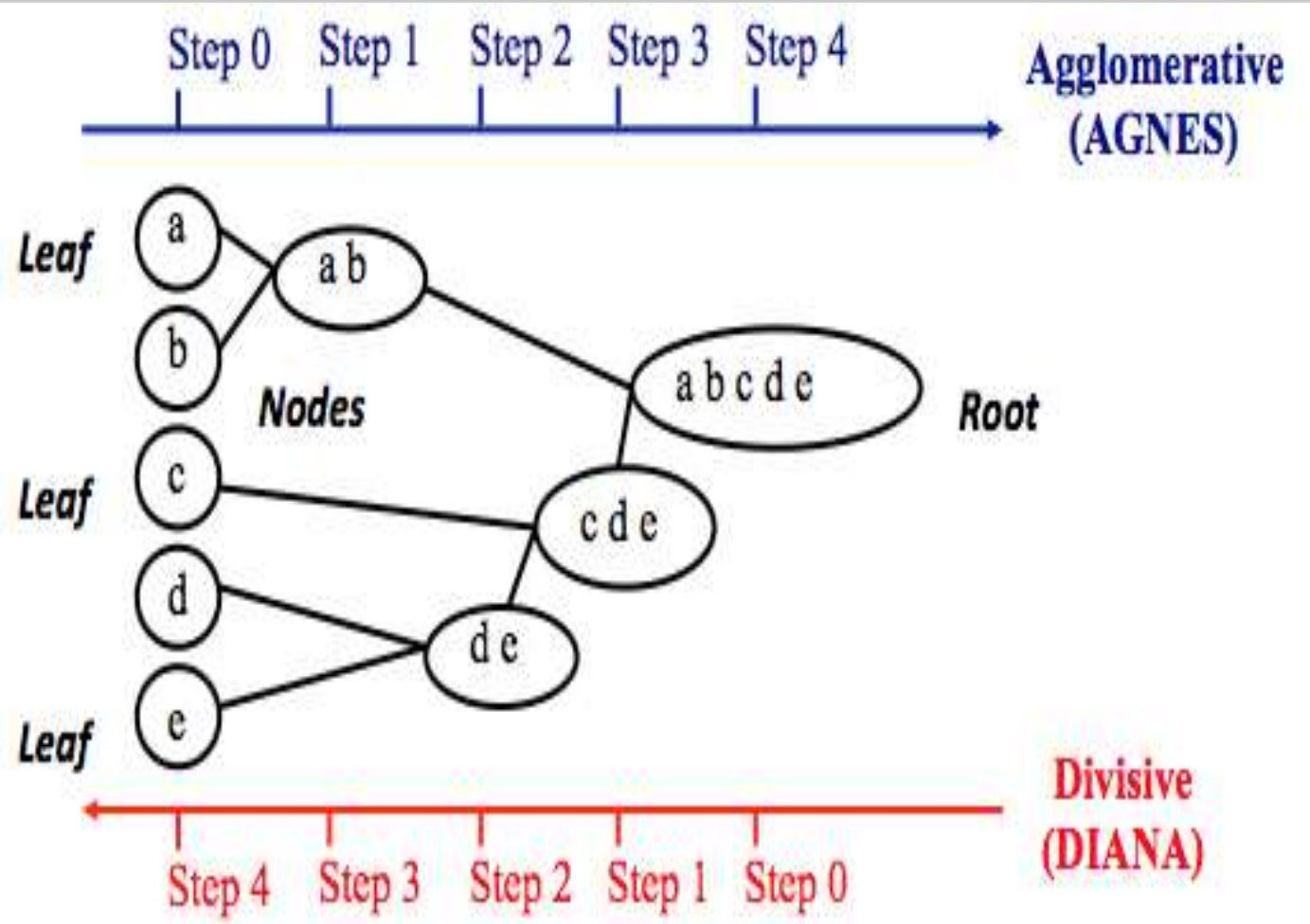
N. Rizk

College of Natural and Applied Sciences

Department of Computer Science

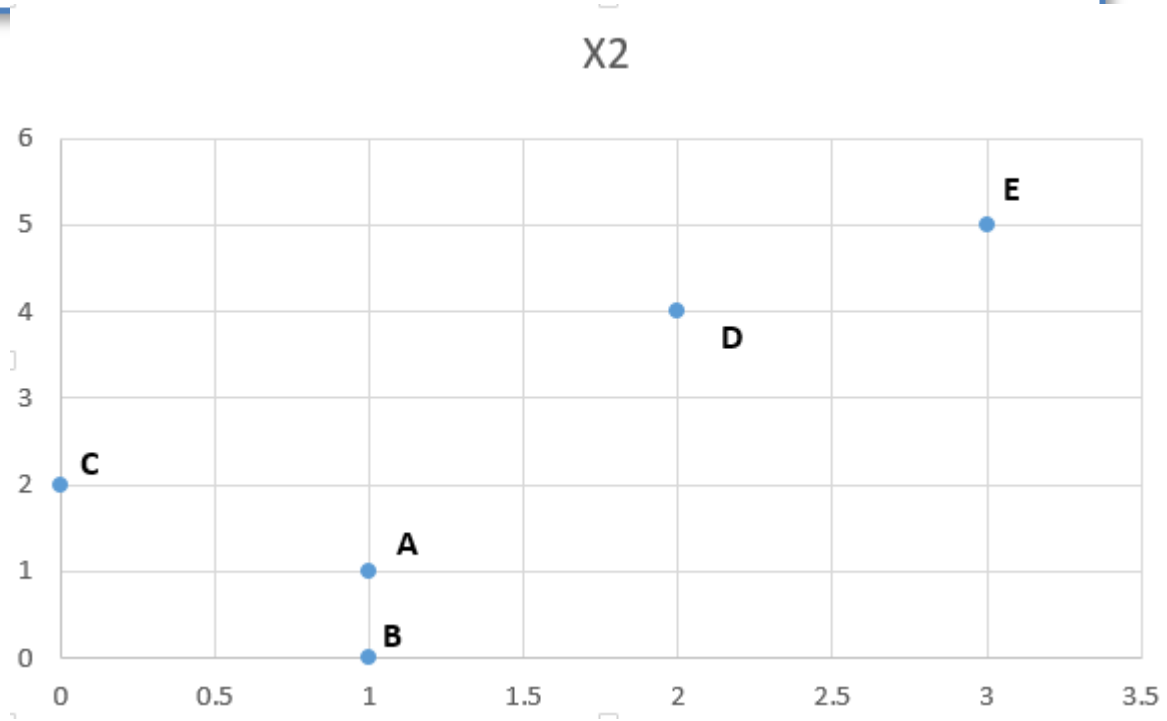
University of Houston

Hierarchical *Agglomerative* Clustering



Example

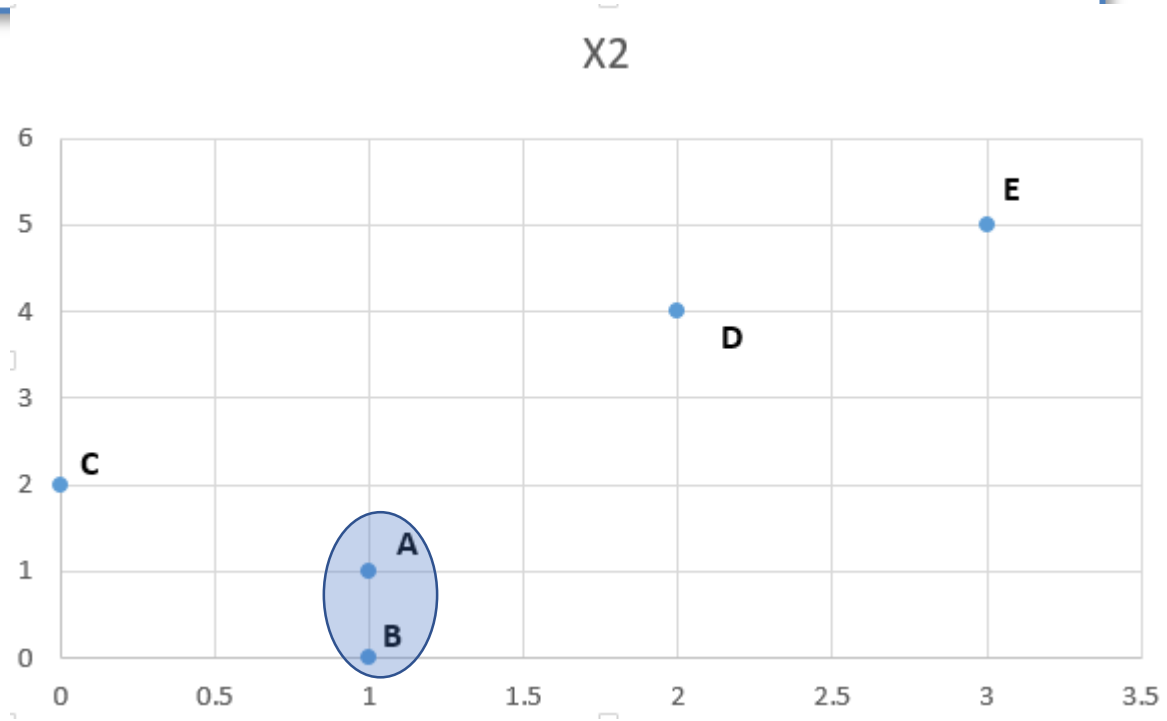
	X1	X2
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5



	A	B	C	D	E
A	0.0	1.0	1.4	3.2	4.5
B	1.0	0.0	2.2	4.1	5.4
C	1.4	2.2	0.0	2.8	4.2
D	3.2	4.1	2.8	0.0	1.4
E	4.5	5.4	4.2	1.4	0.0

Step1 find the smallest distance

	A	B	C	D	E
A	0.0	1.0	1.4	3.2	4.5
B	1.0	0.0	2.2	4.1	5.4
C	1.4	2.2	0.0	2.8	4.2
D	3.2	4.1	2.8	0.0	1.4
E	4.5	5.4	4.2	1.4	0.0

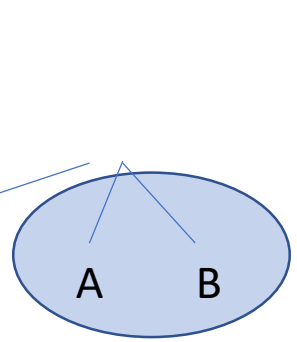


	A	B	C	D	E
A	0.0	1.0	1.4	3.2	4.5
B	1.0	0.0	2.2	4.1	5.4
C	1.4	2.2	0.0	2.8	4.2
D	3.2	4.1	2.8	0.0	1.4
E	4.5	5.4	4.2	1.4	0.0

	AB	C	D	E
AB	1.0	1.8	3.6	4.9
C	1.8	0.0	2.8	4.2
D	3.6	2.8	0.0	1.4
E	4.9	4.2	1.4	0.0

Linkage by average

Dendrogram



Minimum distance at 1

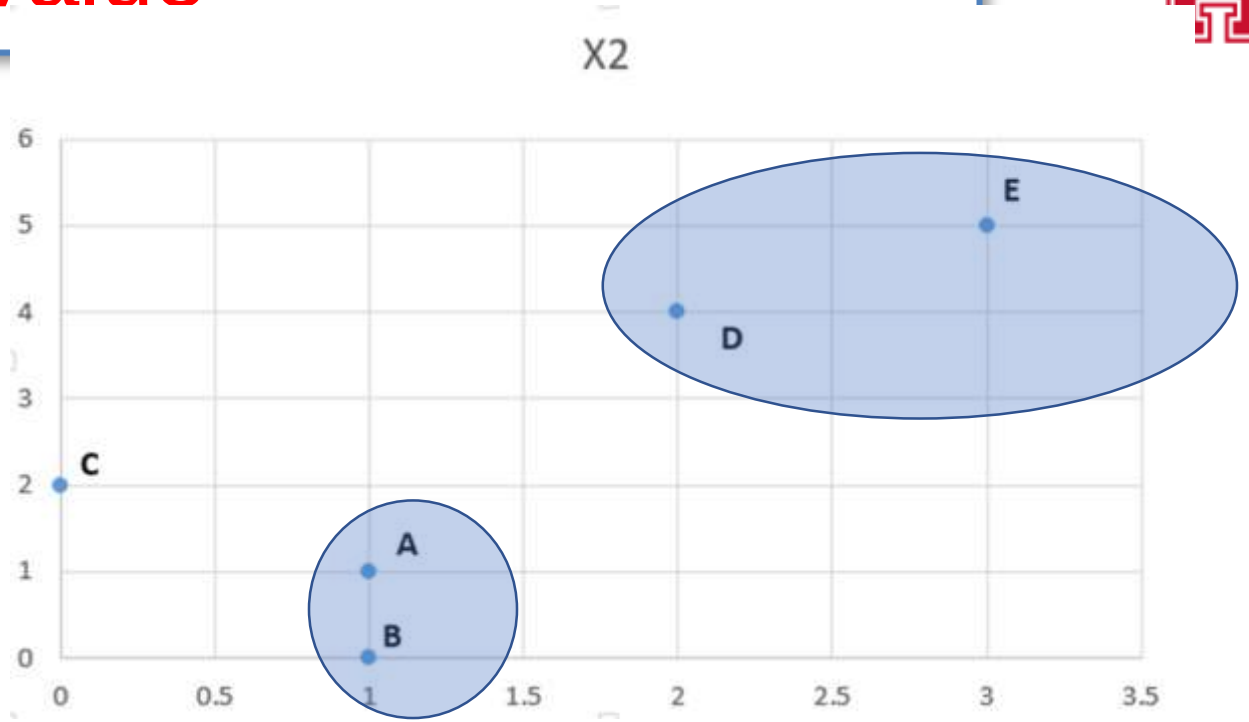
	A	B	C	D	E
A	0.0	1.0	1.4	3.2	4.5
B	1.0	0.0	2.2	4.1	5.4
C	1.4	2.2	0.0	2.8	4.2
D	3.2	4.1	2.8	0.0	1.4
E	4.5	5.4	4.2	1.4	0.0

	AB	C	D	E
AB	0	1.8	3.6	4.9
C	1.8	0.0	2.8	4.2
D	3.6	2.8	0.0	1.4
E	4.9	4.2	1.4	0.0

Linkage by average

Next Step Smallest Value

	AB	C	D	E
AB	0	1.8	3.6	4.9
C	1.8	0.0	2.8	4.2
D	3.6	2.8	0.0	1.4
E	4.9	4.2	1.4	0.0

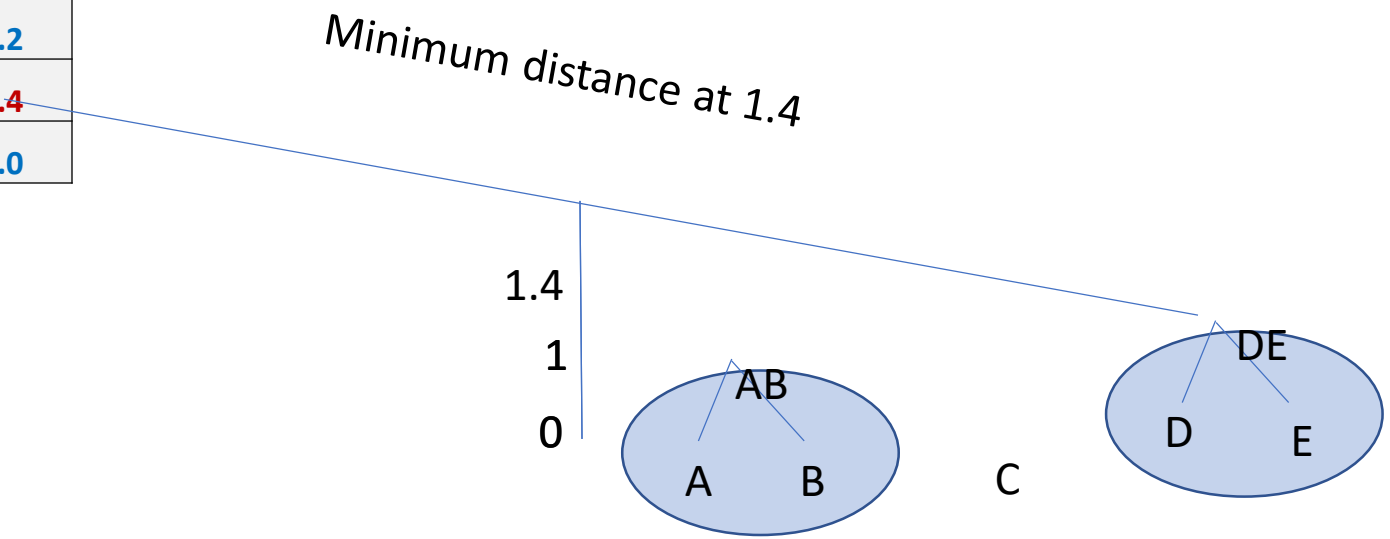


	AB	C	D	E
AB	0	1.8	3.6	4.9
C	1.8	0.0	2.8	4.2
D	3.6	2.8	0.0	1.4
E	4.9	4.2	1.4	0.0

Average 3.6 and 4.9 is the distance between DE and AB

Dendrogram

	AB	C	D	E
AB	0	1.8	3.6	4.9
C	1.8	0.0	2.8	4.2
D	3.6	2.8	0.0	1.4
E	4.9	4.2	1.4	0.0



	AB	C	DE
AB	0	1.8	4.3
C	1.8	0.0	3.5
DE	4.3	3.5	0.0

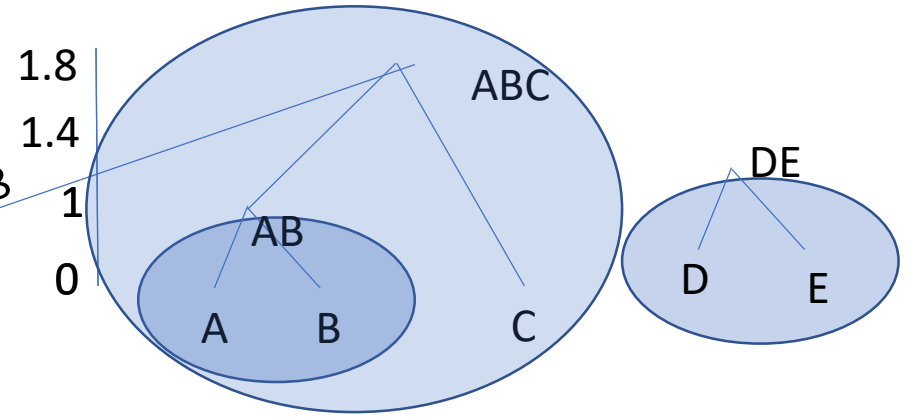
FIND THE SMALLEST

	AB	C	DE
AB	0	1.8	4.3
C	1.8	0.0	3.5
DE	4.3	3.5	0.0

	AB	C	DE
AB	0	1.8	4.3
C	1.8	0.0	3.5
DE	4.3	3.5	0.0

	ABC	DE
ABC	0	4
DE	4	0.0

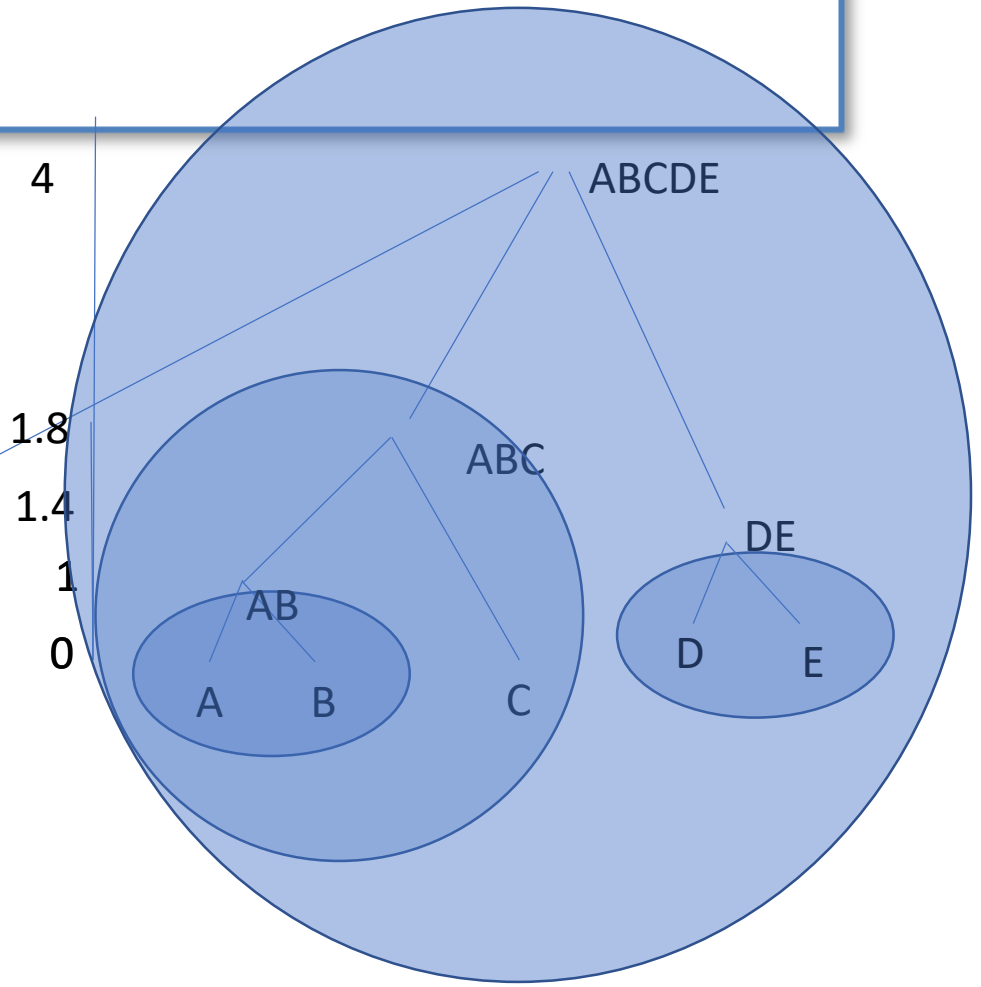
Dendrogram



	AB	C	DE
AB	0	1.8	4.3
C	1.8	0.0	3.5
DE	4.3	3.5	0.0

	ABC	DE
ABC	0	4
DE	4	0.0

Dendrogram



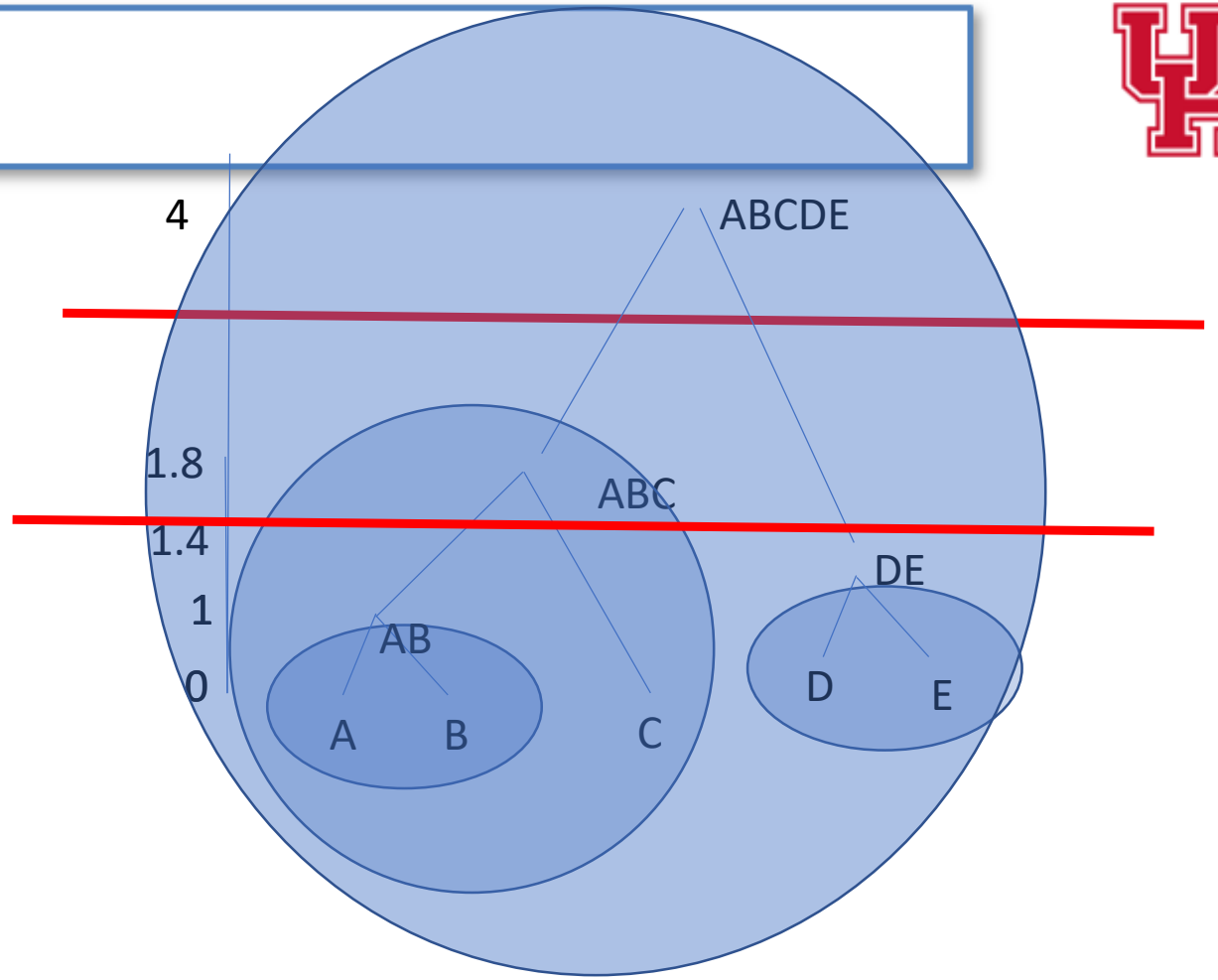
Minimum distance at 4

	ABC	DE
ABC	0	4
DE	4	0.0

Threshold

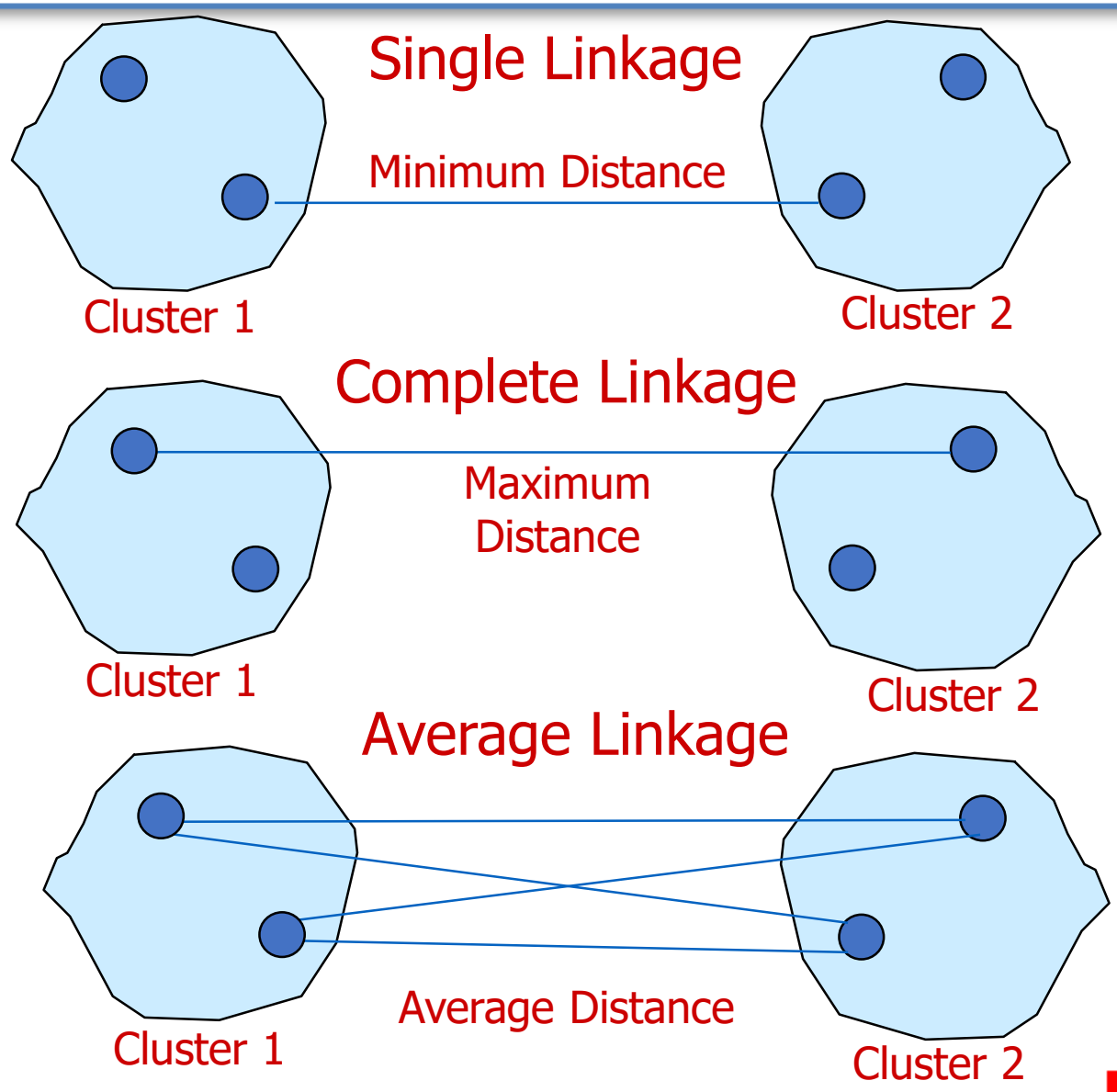
2 clusters at threshold 3

2 clusters and one outlier at threshold 1.5



Optimal threshold can be the mid point of the longest branch 4/2=2

Linkage Methods of Clustering



Repeat this example with Single linkage (select the minimum distance instead of average)

➔

Euclidean , Single Linkage

	A	B	C	D	E
A	0.0	1.0	1.4	3.2	4.5
B	1.0	0.0	2.2	4.1	5.4
C	1.4	2.2	0.0	2.8	4.2
D	3.2	4.1	2.8	0.0	1.4
E	4.5	5.4	4.2	1.4	0.0

	AB	C	D	E
AB	1.0	*1.4*	3.6	4.9
C	1.8	0.0	2.8	4.2
D	3.6	2.8	0.0	1.4
E	4.9	4.2	1.4	0.0

Repeat this example with complete linkage (select the maximum distance instead of average)

Euclidean , complete Linkage

	A	B	C	D	E
A	0.0	1.0	1.4	3.2	4.5
B	1.0	0.0	2.2	4.1	5.4
C	1.4	2.2	0.0	2.8	4.2
D	3.2	4.1	2.8	0.0	1.4
E	4.5	5.4	4.2	1.4	0.0

	AB	C	D	E
AB	1.0	***2.2***	3.6	4.9
C	1.8	0.0	2.8	4.2
D	3.6	2.8	0.0	1.4
E	4.9	4.2	1.4	0.0

Hierarchical *Divisive* Clustering example

Hierarchical K-Means =top down

- Run K-means on the original set of n objects
- Apply k means , split recursively (same K) until a singleton cluster
- Complexity O(KndLog $_k$ n)➔fast
- Disadvantage nearest points may end up in different clusters

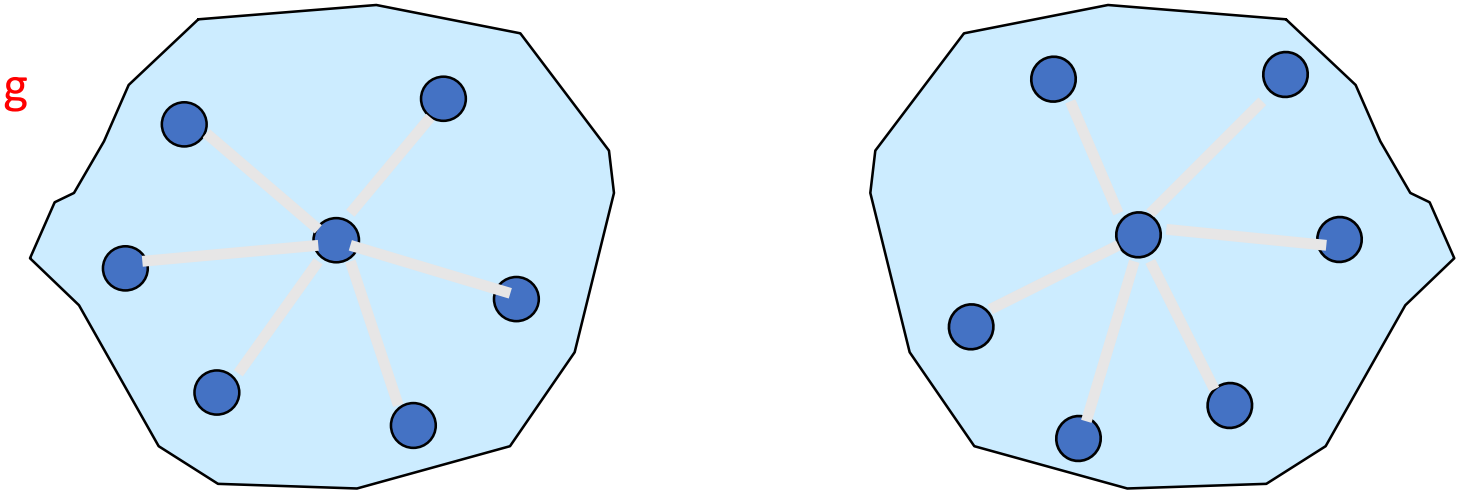
Hierarchical Agglomerative Clustering- Variance and Centroid Method

- **Variance methods** generate clusters to minimize the within-cluster variance.
- **Ward's procedure** is commonly used. For each cluster, the sum of squares is calculated. The two clusters with the smallest increase in the overall sum of squares within cluster distances are combined.
- In the **centroid methods**, the distance between two clusters is the distance between their centroids (means for all the variables),
- Of the hierarchical methods, average linkage and Ward's methods have been shown to perform better than the other procedures.

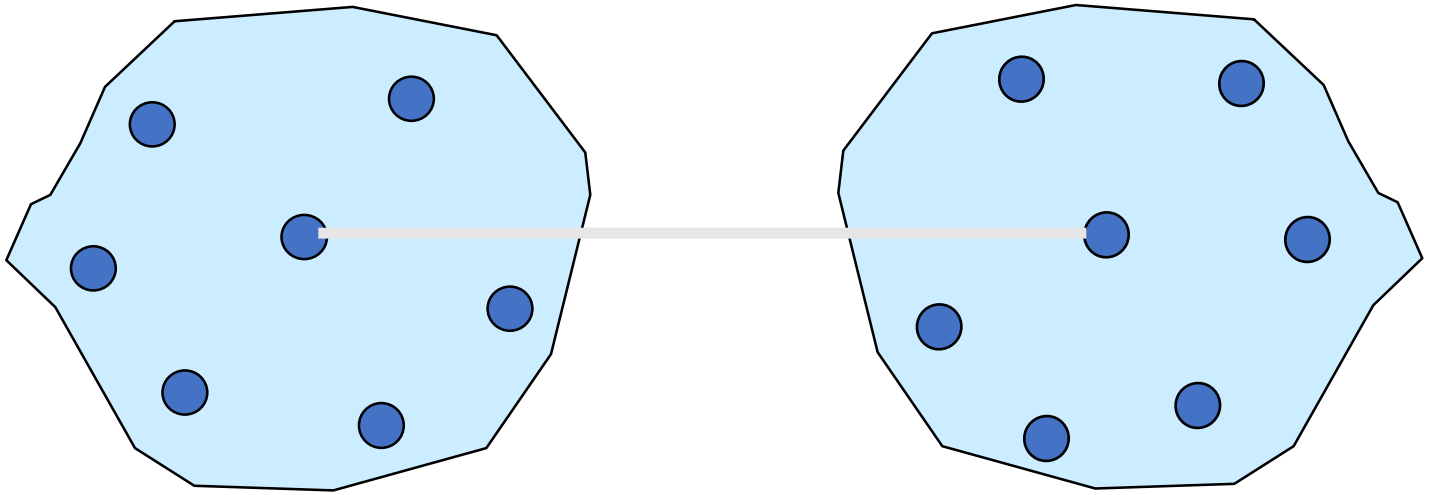
Other Agglomerative Clustering Methods

Repeat the example using Ward's procedure!

Ward's Procedure



Centroid Method



Ward's Method (Minimum variance method)

Like other clustering methods, Ward's method starts with *n* clusters, each containing a single object. These *n* clusters are combined to make one cluster containing all objects. At each step, the process makes a new cluster that minimizes variance , measured by an index called *E* (also called the *sum of squares index*).

ESS? Sum of squares

At each step, the following calculations are made to find *E*:

- **Find the mean of each cluster.**
- **Calculate the distance between each object in a particular cluster, and that cluster's mean.**
- **Square the differences from Step 2.**
- **Sum (add up) the squared values from Step 3.**
- **Add up all the sums of squares from Step 4.**

Let X_{ijk} denote the value for variable k in observation j belonging to cluster i:

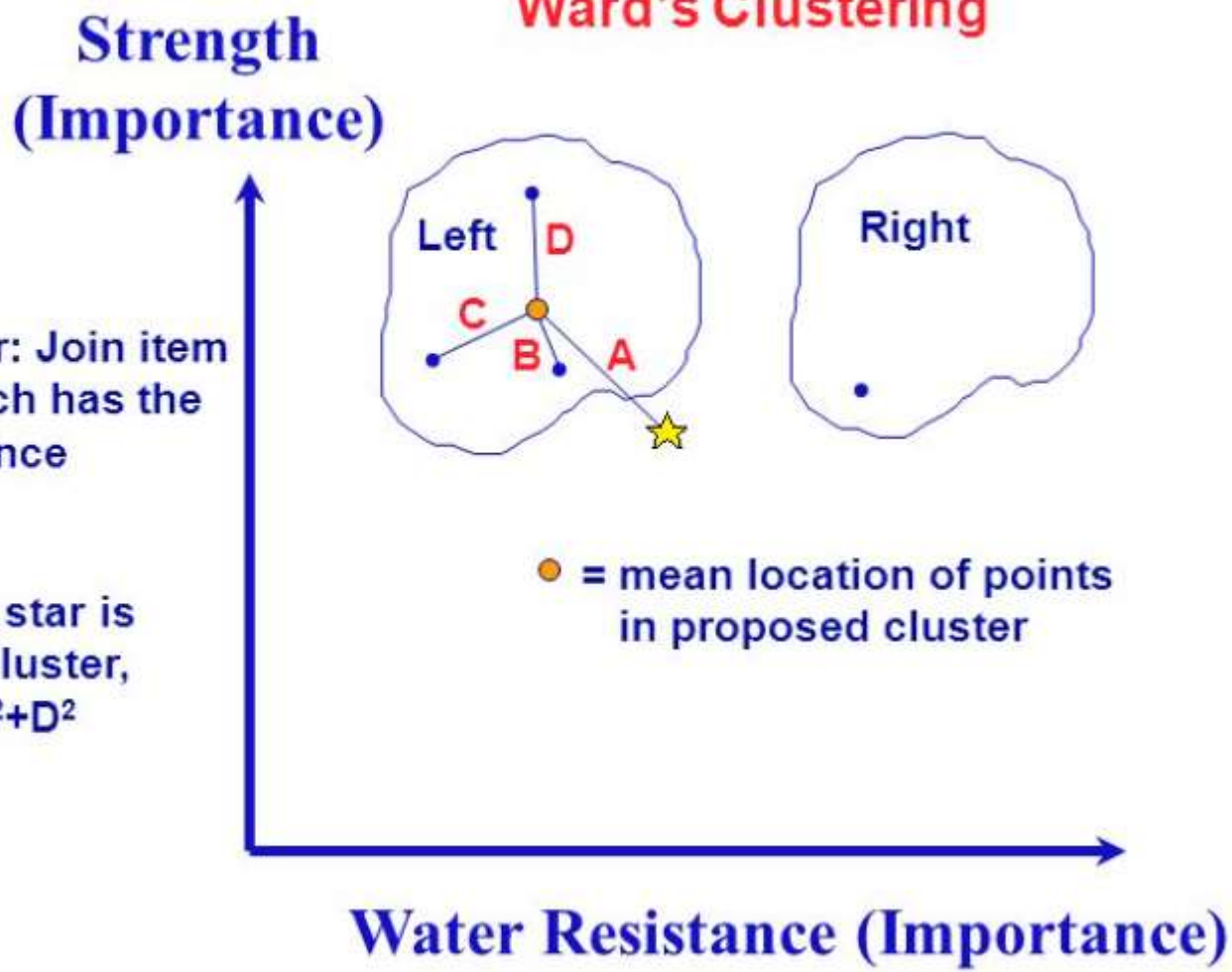
$$ESS(X) = \sum_i \sum_j \sum_k (X_{ijk} - \bar{X}_{i \cdot k})^2$$

$$= \sum_{clusters} \sum_{cases} (X_{ij} - centroid_i)^2$$

- At the beginning each case is a cluster.
- In the first step of the algorithm, $n - 1$ clusters are formed, one of size two and the remaining of size 1. The error sum of squares is computed. The pair of sample units that yield the smallest *ESS* will form the first cluster.

- Then, in the second step of the algorithm n – 2 clusters are formed from that $n - 1$ clusters defined in step 2. These may include two clusters of size 2, or a single cluster of size 3 including the two items clustered in step 1.
- Again, the value of ESS is minimized.
- Thus, at each step of the algorithm clusters or observations are combined in such a way as to minimize the results of error from the squares.
- The algorithm stops when all sample units are combined into a single large cluster of size n.

Ward's Clustering



Ward's Cluster: Join item to cluster which has the smallest distance ESS.

In this case, if star is joined to left cluster, $ESS=A^2+B^2+C^2+D^2$

Ward's Minimum Variance Agglomerative Clustering Procedure

First Stage:	A= 2	B=5	C=9	D=10	E=15
Second Stage:		AB= 4.5	BD=12.5		
		AC=24.5	BE=50.0		
		AD=32.0	CD= 0.5		
		AE=84.5	CE=18.0		
		BC= 8.0	DE=12.5		
Third Stage:	CDA=38.0	CDB=14	CDE=20.66	AB= 5.0	
	AE =85	BE =50.5			
Fourth Stage:		ABCD=41.0	ABE=93.17	CDE=25.18	
Fifth Stage:			ABCDE=98.8		

one of size two and the remaining of size 1
CD, A,B,E

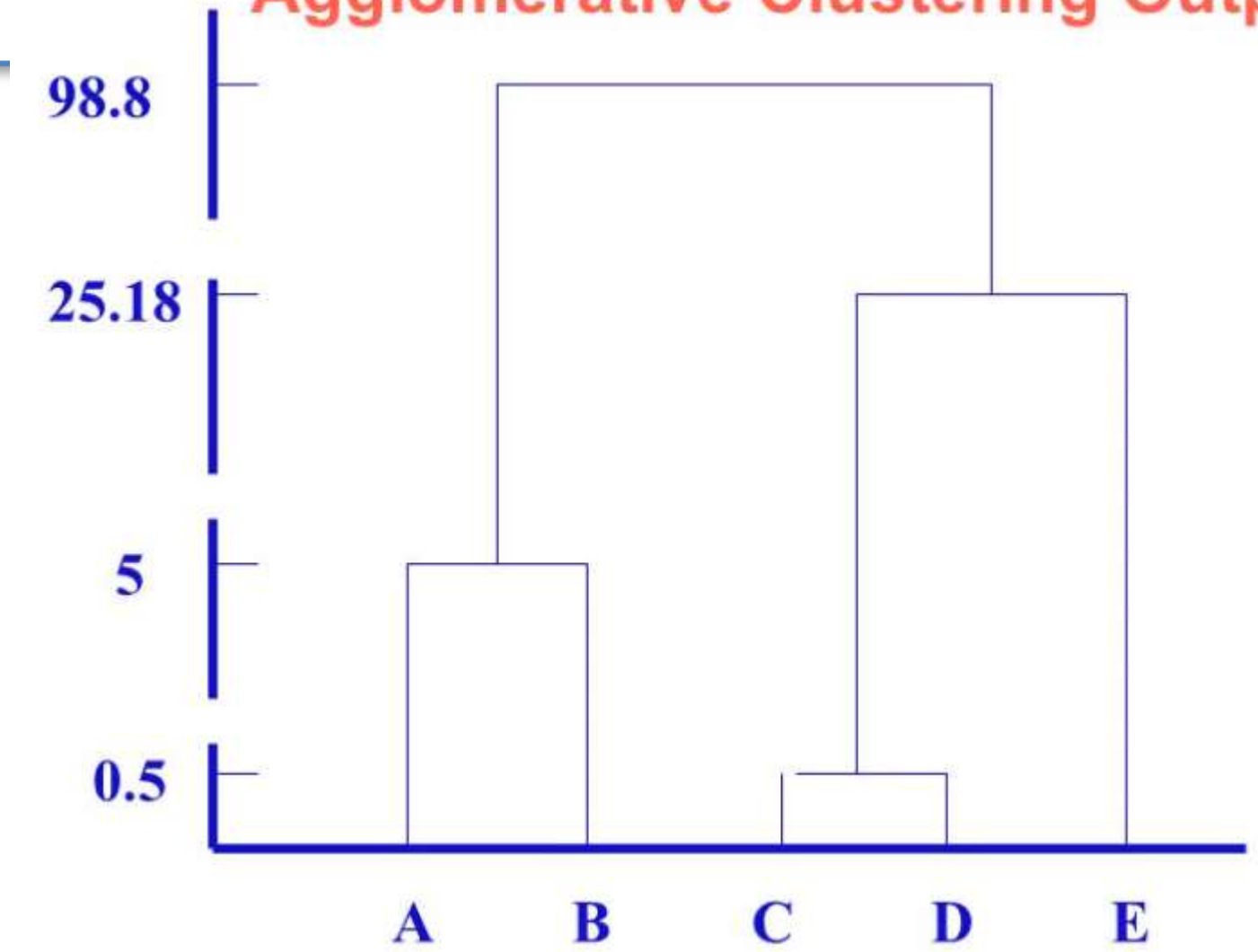
CD =(10-9)^2 /2 = 1/2=0.5

two clusters of size 2 (CD,AB, a single cluster of size 1

AB, CDE,

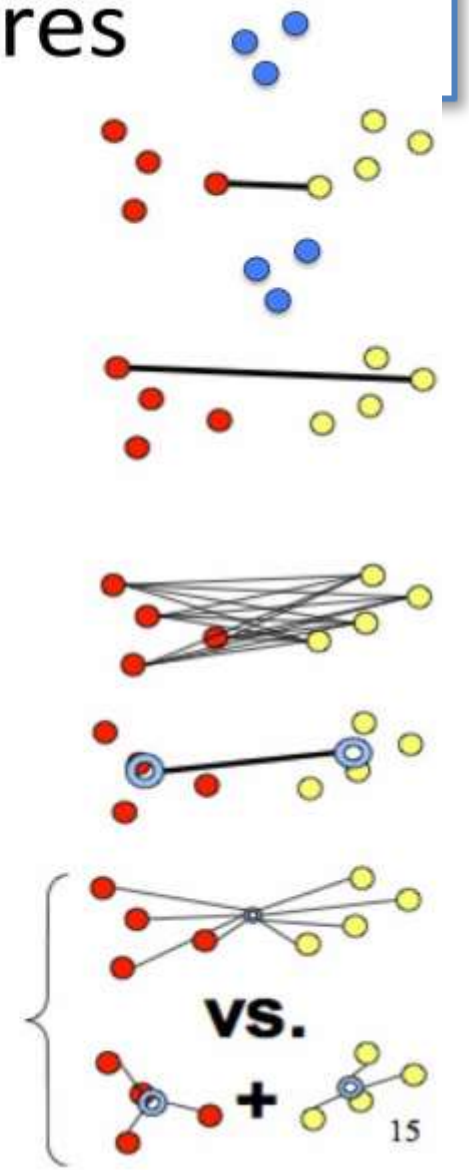
ABCDE

Agglomerative Clustering Output

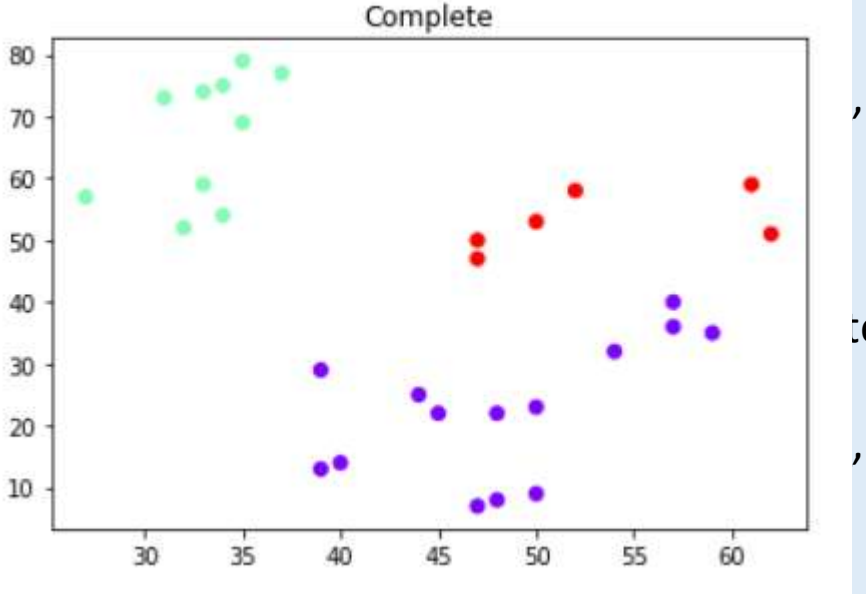


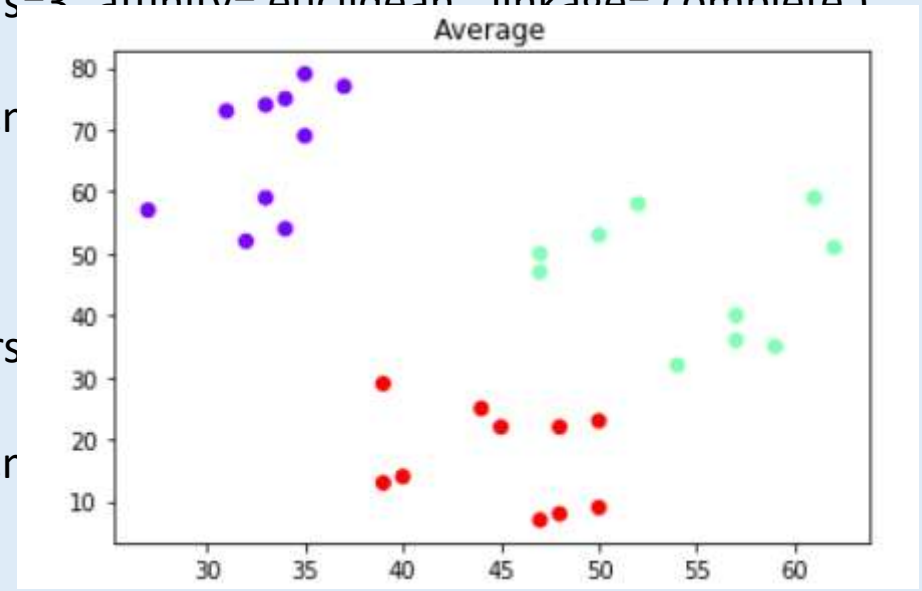
Cluster distance measures

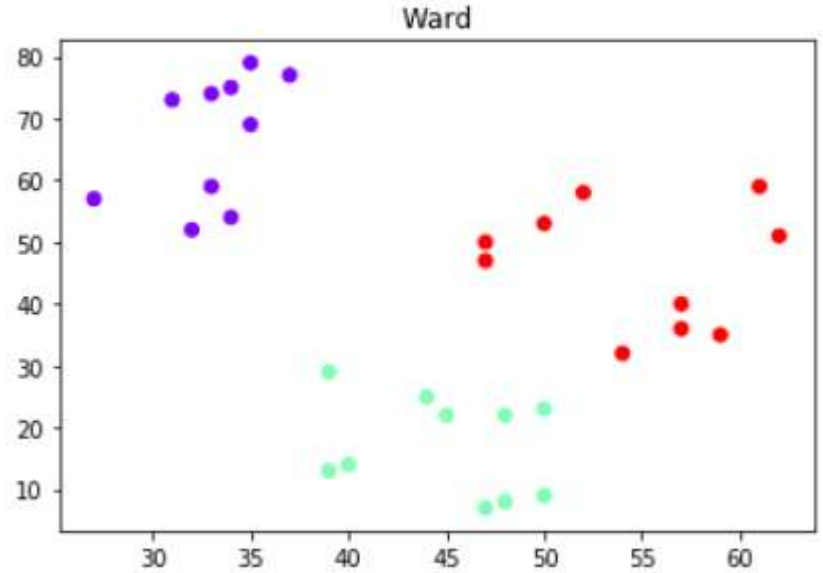
- Single link: $D(c_1,c_2) = \min_{x_1 \in c_1, x_2 \in c_2} D(x_1,x_2)$
  - distance between closest elements in clusters
  - produces long chains a→b→c→…→z
- Complete link: $D(c_1,c_2) = \max_{x_1 \in c_1, x_2 \in c_2} D(x_1,x_2)$
  - distance between farthest elements in clusters
  - forces "spherical" clusters with consistent "diameter"
- Average link: $D(c_1,c_2) = \frac{1}{|c_1|}\frac{1}{|c_2|}\sum_{x_1 \in c_1}\sum_{x_2 \in c_2} D(x_1,x_2)$
  - average of all pairwise distances
  - less affected by outliers
- Centroids: $D(c_1,c_2) = D\left(\left(\frac{1}{|c_1|}\sum_{x \in c_1}\vec{x}\right),\left(\frac{1}{|c_2|}\sum_{x \in c_2}\vec{x}\right)\right)$
  - distance between centroids (means) of two clusters
- Ward's method: $TD_{c_1 \cup c_2} = \sum_{x \in c_1 \cup c_2} D(x,\mu_{c_1 \cup c_2})^2$
  - consider joining two clusters, how does it change the total distance (TD) from centroids?

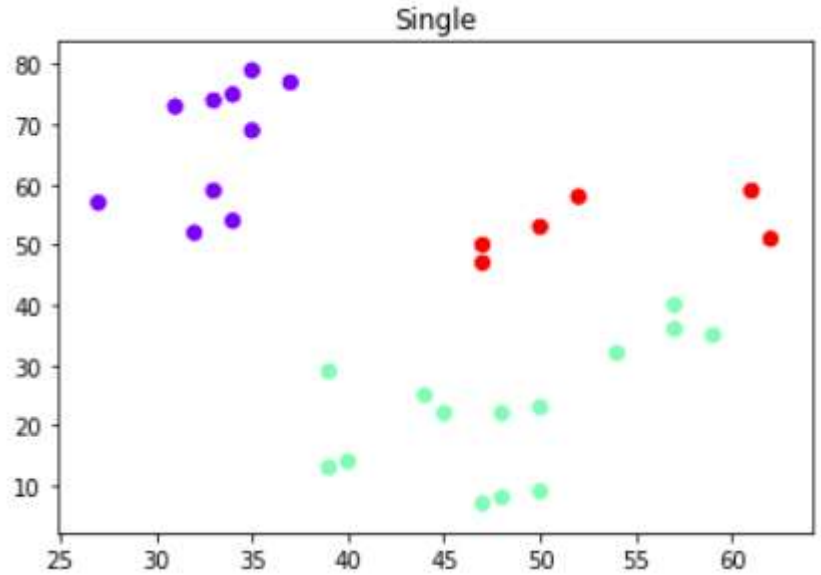


```
import matplotlib.pyplot as plt
from pandas import DataFrame
from sklearn.cluster import AgglomerativeClustering

Data = {
            'x': [35,34,32,37,33,33,31,27,35,34,62,54,57,47,50,57,59,52,61,47,50,48,39,40,45,47,39,44,50,48],
     'y': [79,54,52,77,59,74,73,57,69,75,51,32,40,47,53,36,35,58,59,50,23,22,13,14,22,7,29,25,9,8]
    }

df = DataFrame(Data,columns=['x','y'])

cluster = AgglomerativeClustering(n_clusters=3, affinity='euclidean', linkage='ward')
cluster.fit_predict(df)
plt.scatter(df['x'], df['y'], c=cluster.labels_, cmap='rainbow')
plt.title('Ward');
plt.show()

cluster = AgglomerativeClustering(n_clusters=3, affinity='euclidean', linkage='single')
cluster.fit_predict(df)
plt.scatter(df['x'], df['y'], c=cluster.labels_, cmap='rainbow')
plt.title('Single')
plt.scatter(df['x'], df['y'], c=cluster.labels_, cmap='rainbow')
plt.show()
```


Complete


Average


Ward


Single