Assignment 3 - TDT4300

Hermann Owren Elton, Olaf Rosendahl

March 6, 2023

2 Hierarchical Agglomerative Clustering (HAC)

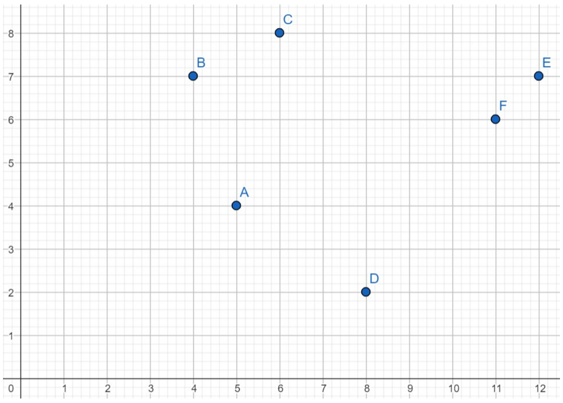
a)

With Hierarchical Agglomerative Clustering, one start with each point as an individual cluster. Then, at each step, the closest pair of clusters is mergeed until only one cluster (or k clusters) is left.

When using MIN-link, the distance between the points in each of the clusters that are closest to each other is used. When using MAX-link it is the distance between the points which are most apart that are used.

b)

When plotting the dataset in Table 1 in a graph, it looks like this:



Then we created a table showing the eucledian distance between all pairs of points:

	A	B	C	D	E	F
A	0					
B	3.16	0				
C	4.12	2.24	0			
D	3.61	6.40	6.32	0		
E	7.62	8.0	6.08	6.40	0	
F	6.32	7.07	5.39	5.0	1.41	0

Starting with MIN-link:

Step 1: Merges EF since it has the lowest distance and then recomputes the table of distances:

	A	B	C	D	EF
A	0				
B	3.16	0			
C	4.12	2.24	0		
D	3.61	6.40	6.32	0	
EF	6.32	7.07	5.39	5.0	0

Step 2: Merges BC:

	A	BC	D	EF
A	0			
BC	3.16	0		
D	3.61	6.32	0	
EF	6.32	5.39	5.0	0

Step 3: Merges ABC:

	ABC	D	EF
ABC	0		
D	3.61	0	
EF	5.39	5.0	0

Step 4: Merges ABCD:

	ABCD	EF
ABCD	0	
EF	5.0	0

Step 5: Merges ABCDEF and do now have only 1 cluster left.

Then with MAX-link:

Step 1: Merges EF since it has the lowest distance and then recomputes the table of distances, but this time using the largest distance in the cluster:

	A	B	C	D	EF
A	0				
B	3.16	0			
C	4.12	2.24	0		
D	3.61	6.40	6.32	0	
EF	7.62	8.0	6.08	6.40	0

Step 2: Merges BC:

	A	BC	D	EF
A	0			
BC	4.12	0		
D	3.61	6.40	0	
EF	7.62	8.0	6.40	0

Step 3: Merges AD:

	AD	BC	EF
AD	0		
BC	6.40	0	
EF	8.0	6.40	0

Step 4: Merges ADBC (equal distance between AD-BC and BC-EF, picks AD-BC):

	ADBC	EF
ADBC	0	
EF	8.0	0

Step 5: Merges ADBCEF and do now have only 1 cluster left.

c)

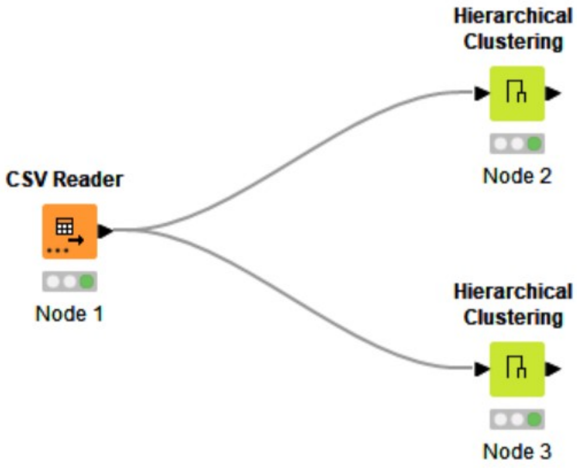
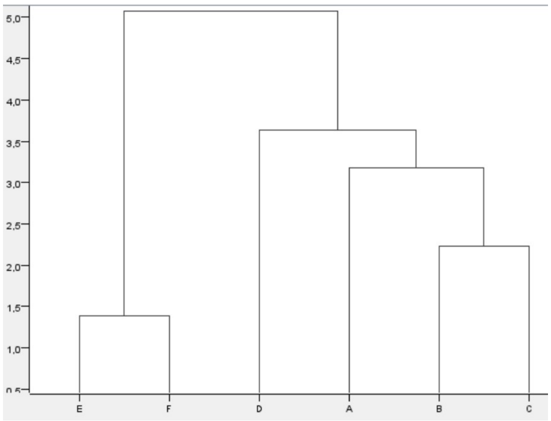
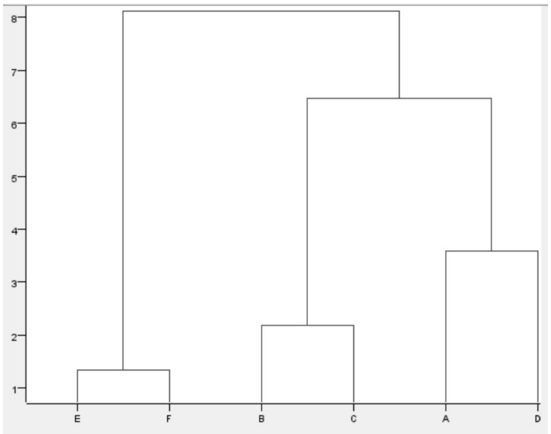


Figure 1: KNIME workflow



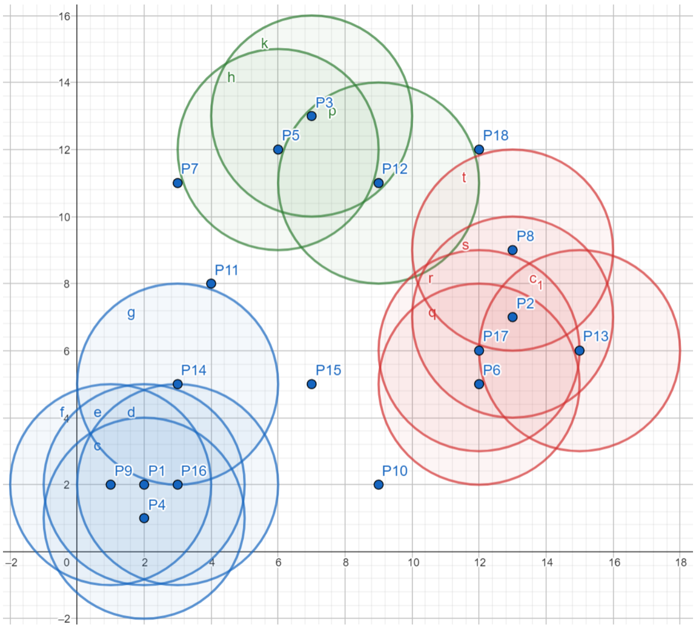
(a) KNIME Dendrogram with MIN-link



(b) KNIME Dendrogram with MAX-link

3 DBSCAN Clustering

a)



We dicided to use Geogebra for this task. We plotted all points and created circles around each point with a radius of 3. The circles with less then $MinPts = 2$ around the center was then removed. Then the circles which formed clusters was colored with the same color. We ended up with 3 clusters in total.

Core points: $\{P1, P2, P3, P4, P5, P6, P8, P9, P12, P13, P14, P16, P17\}$

Border points: $\{\}$

Noise points: $\{P7, P10, P11, P15, P18\}$

b)

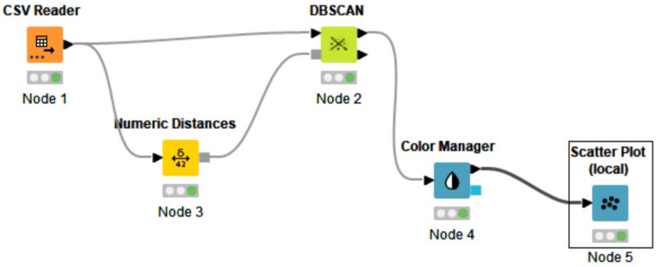


Figure 3: KNIME workflow

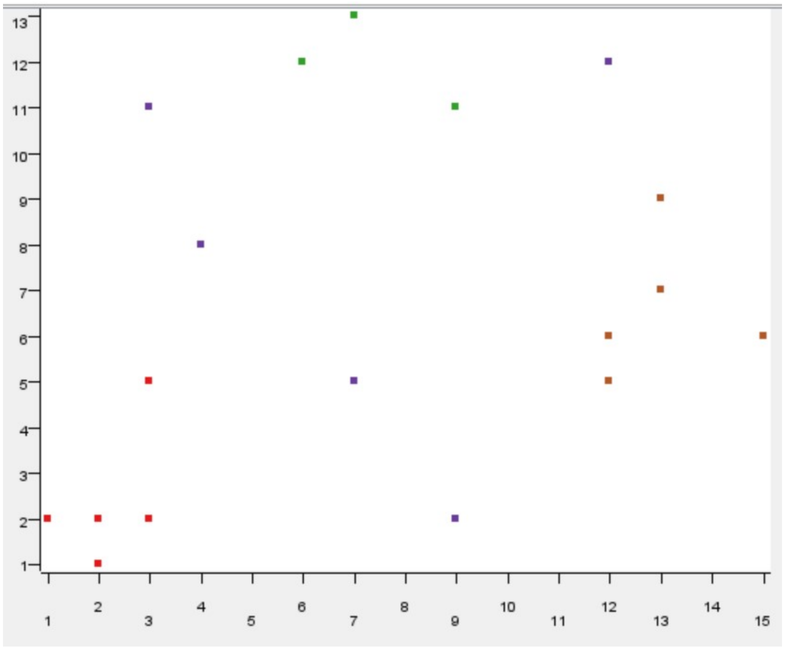


Figure 4: KNIME Scatterplot, purple points are noise

As seen in Figure 4, the results from KNIME corresponds with our results from our manual calculations in Geogebra.