COMPSCI 361 (2024 Semester 1)
Tutorial Week 9

Sam Thompson, Jackson Ayling-Campbell

The University of Auckland

May 8, 2024

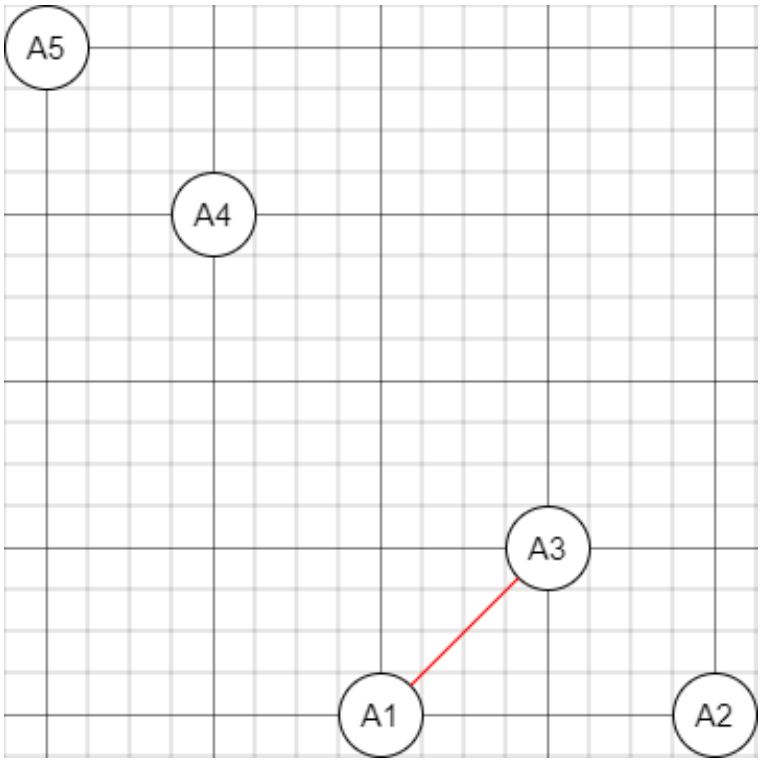
Hierarchical Clustering

- Considers clusters as a hierarchy, i.e., smaller clusters inside larger clusters
- Hierarchy can be created in two ways:
  - Bottom up (Agglomerative): Merge smaller clusters into a larger encompassing cluster
  - Top down (Divisive): Separate larger clusters into smaller subset clusters
- We can stop merging or splitting clusters wherever we like, so do not have to specify K explicitly
- Gives us information about relationships between clusters

Hierarchical Clustering

- Start with each instance as its own cluster. Repeat:
- Calculate distances between clusters
- Merge closest clusters
  - Many ways of calculating distance between clusters, which change properties of the resulting clusters
  - Min (Single Link), Max (Complete Link), Average, Centroid, Ward

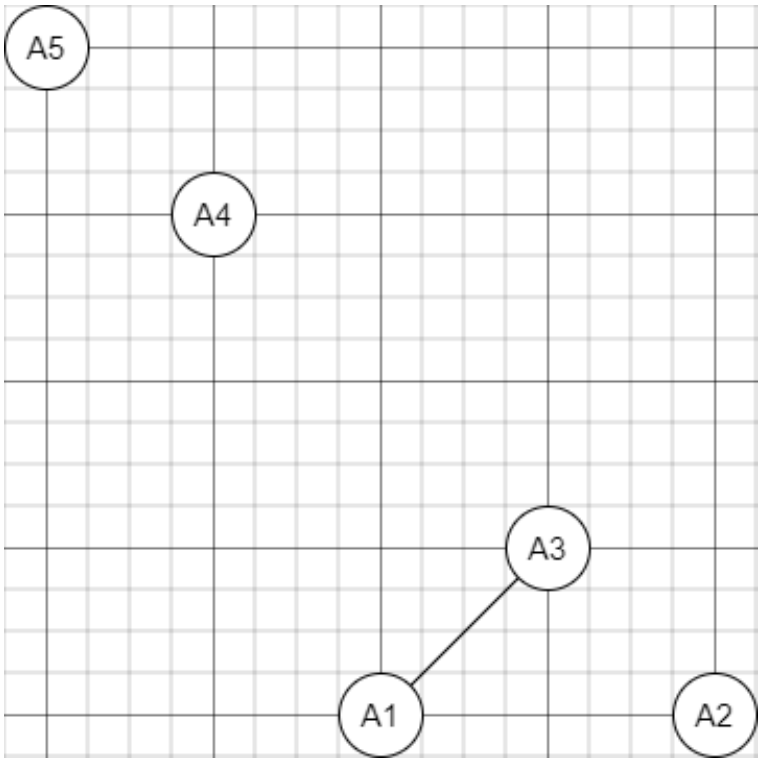
Agglomerative Clustering Example



	A1	A2	A3	A4	A5
A1	0	2	2	4	6
A2	2	0	2	6	8
A3	2	2	0	4	6
A4	4	6	4	0	2
A5	6	8	6	2	0

Table 1: Distances - Note: Using Manhattan distance, Min distance is 2

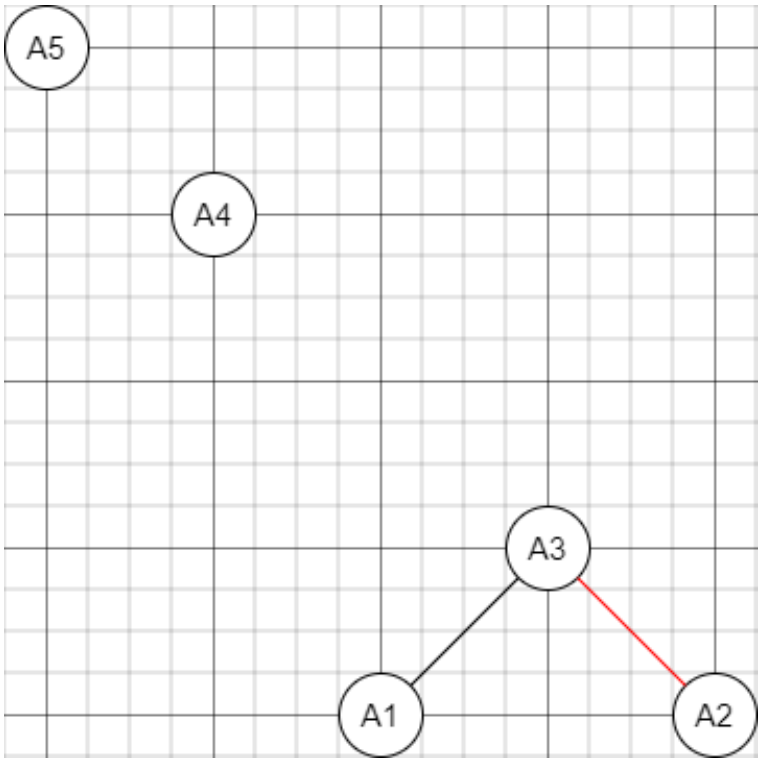
Agglomerative Clustering Example



	C1	A2	C1	A4	A5
C1	0	2	0	4	6
A2	2	0	2	6	8
C1	0	2	0	4	6
A4	4	6	4	0	2
A5	6	8	6	2	0

Table 2: Min (Single-link): Update distances to all nodes in the new cluster to be the min of any node in the cluster

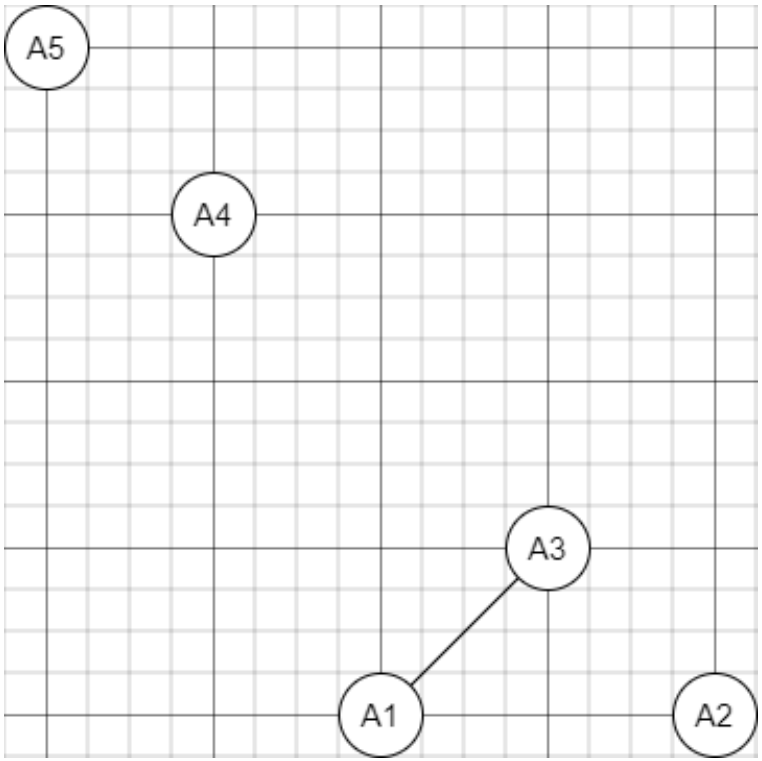
Agglomerative Clustering Example



	C1	A2	C1	A4	A5
C1	0	2	0	4	6
A2	2	0	2	6	8
C1	0	2	0	4	6
A4	4	6	4	0	2
A5	6	8	6	2	0

Table 3: Next min distance is 2

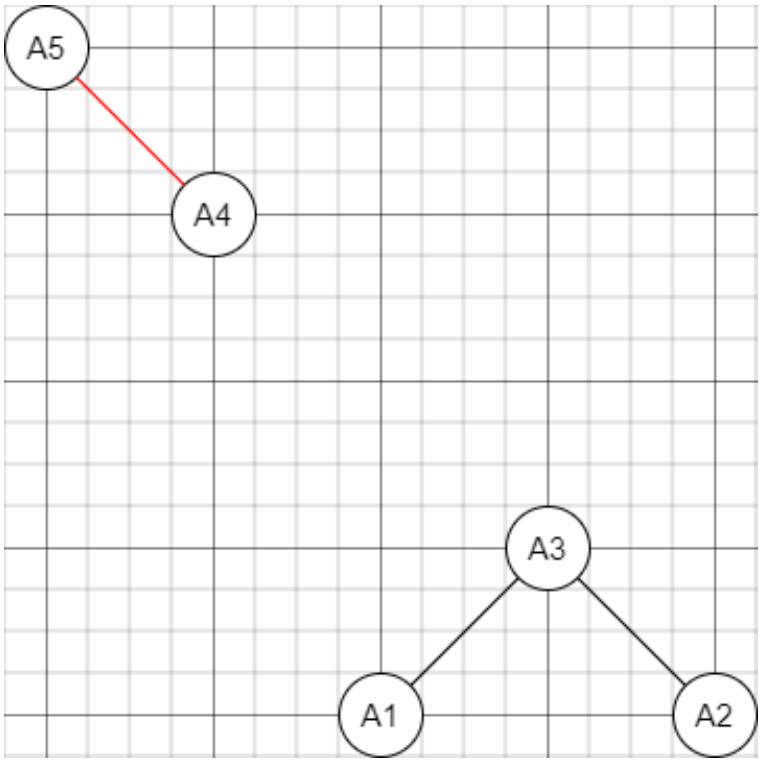
Agglomerative Clustering Example



	C2	C2	C2	A4	A5
C2	0	0	0	4	6
C2	0	0	0	4	6
C2	0	0	0	4	6
A4	4	4	4	0	2
A5	6	6	6	2	0

Table 4: Update Distance

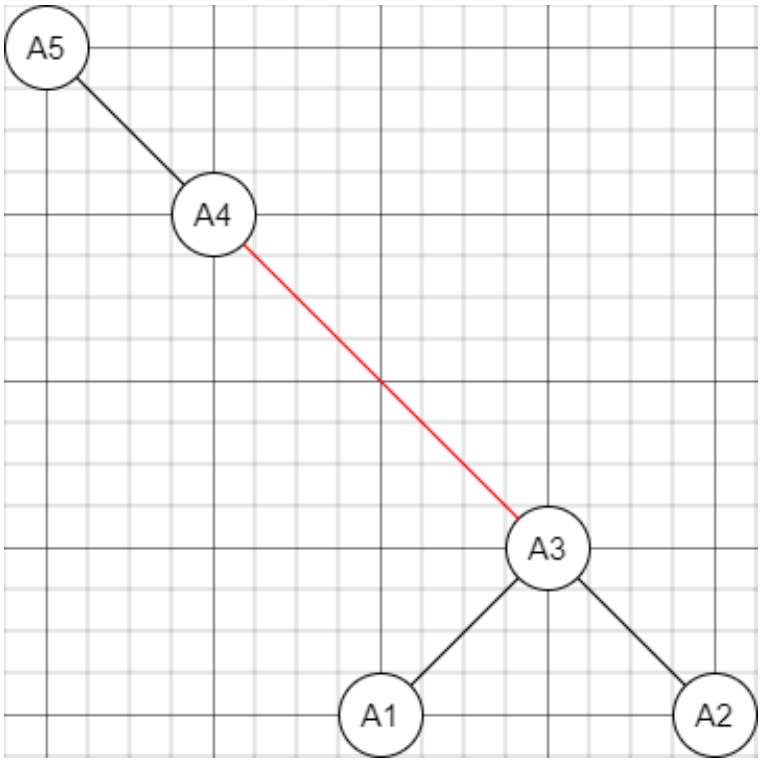
Agglomerative Clustering Example



	C2	C2	C2	C3	C3
C2	0	0	0	4	4
C2	0	0	0	4	4
C2	0	0	0	4	4
C3	4	4	4	0	0
C3	4	4	4	0	0

Table 5: Next min distance is 2 and update

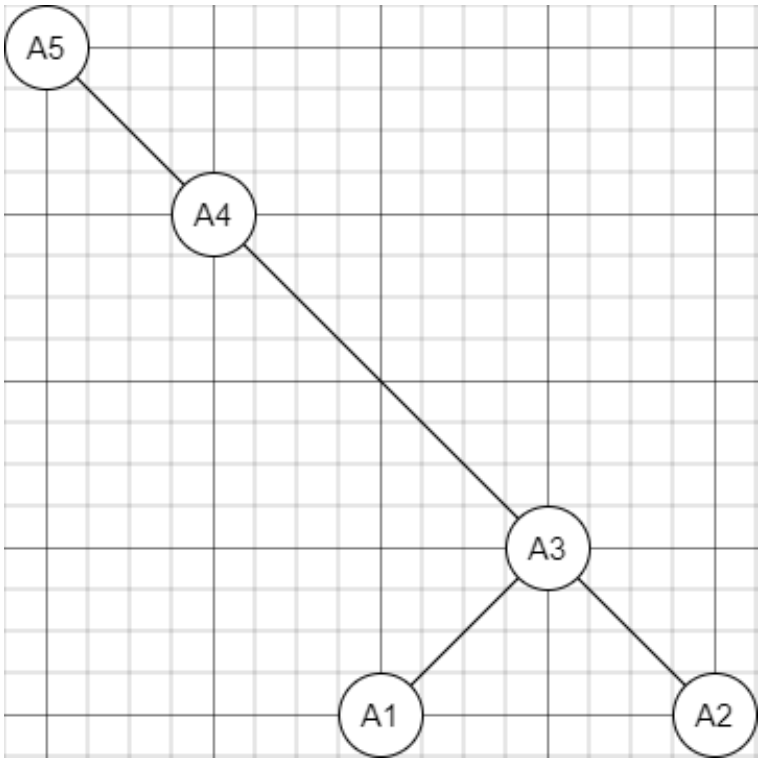
Agglomerative Clustering Example



	C4	C4	C4	C4	C4
C4	0	0	0	0	0
C4	0	0	0	0	0
C4	0	0	0	0	0
C4	0	0	0	0	0
C4	0	0	0	0	0

Table 6: Next min distance is 4 and update

Agglomerative Clustering Dendrogram



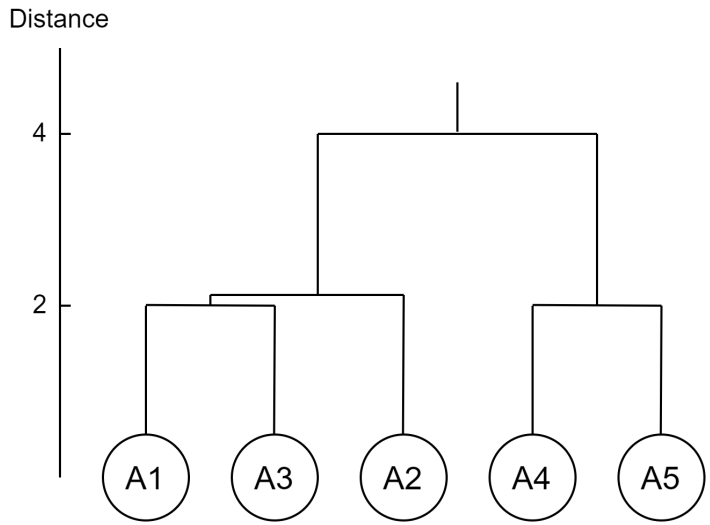


Figure 1: Dendrogram

Agglomerative Clustering Summary

- (+) Starting point is not an issue compared to K-Means
- (-) Tendency to produce large clusters (single-link)
- (-) Sensitive to outliers
- (-) Difficulty handling convex shapes (biased towards globular clusters)
- (-) Difficulty handling clusters of different sizes

Silhouette coefficient

Evaluate and compare the clusters discovered (i) agglomerative clustering task with single-link policy, and (ii) DBSCAN using the internal measurement: Silhouette Coefficient. You may need to determine an appropriate cutting threshold for the agglomerative clustering task with a single-link policy.

	A1	A2	A3	A4	A5	A6	A7	A8
A1	0	45	63	57	41	28	95	6
A2		0	55	49	35	11	5	25
A3			0	11	23	54	47	65
A4				0	2	7	26	5
A5					0	5	21	35
A6						0	13	27
A7							0	53
A8								0

Silhouette coefficient

What does it mean if the silhouette coefficient is low/high at the individual level, cluster level, and overall clustering?

A high silhouette coefficient at the individual level indicates the instance is well-matched to its own cluster and poorly matched to neighboring clusters. At the clustering level, it means clusters are well apart from each other and clearly distinguished. Vice versa for low silhouette coefficient.

Silhouette coefficient

Which cluster algorithm delivers the resulting clustering with the lowest and the highest silhouette coefficient?

The silhouette coefficient for each method is:
k-means: 0.527, DB scan: 0.271, agglomerative: 0.223

Silhouette coefficient

Which cluster algorithm delivers balanced resulting clusters (i.e., each cluster has a similar amount of points)?

The clustering for each method is:
k-means: [1, 0, 0, 0, 0, 0, 0, 1],
DBSCAN: [0, 1, -1, 0, 0, 0, 1, 0],
Agglomerative: [0, 0, 1, 0, 0, 0, 0, 0]
In this case, DBSCAN produces the most balanced clustering.

k-Nearest Neighbour

- Instance-based Learning that use the entire dataset as a model
- Uses k closest points (nearest neighbors) for classification
- For an unknown instance:
  - Compute distance to other training instances according to some distance metric (*e.g.* Euclidean/Manhattan)
  - Identify k nearest neighbors
  - Assign class label based on the label of the k nearest neighbors
- Suffers from curse of dimensionality, calculating the distance between instances can be expensive for high number of attributes

k-NN Hyperparamters

- If k too small, sensitive to noise points
- If k too large, makes it more computationally expensive and more likely to include instances of different classes
- The number of features and classes don't contribute in determining the value of k
- A general rule of thumb for choosing k is $k = \sqrt{n}$, where n is the number of instances in the training set.

k-NN Decision Boundary

- kNN decision boundaries can be any arbitrary shape
- The 1NN decision boundaries can be drawn by creating a Vonoroi diagram.
- Formed by creating a perpendicular bisector to the neighbours.