Practical 3

**Clustering Review: K-Means & Agglomerative**

# What are we doing?

Among various clustering algorithms, K-means algorithm and Agglomerative algorithm are two popular methods: K-means is a distance-based non-hierarchical clustering method and Agglomerative is a typical hierarchical algorithm.

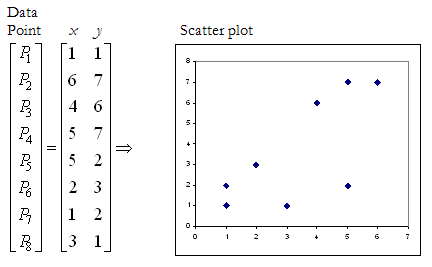
The purpose of this prac is to understand working principle of these two clustering algorithms throughout an example data.

**Submission:**

You are required to submit one document containing the solution of the last task given in this practical.

# Hierarchical Clustering (Agglomerative)

Let’s consider eight data points with two dimensions x and y as candidates for agglomerative clustering. The data points, along with dimensional values and scatter plot, are given in the figure as below:

****

We use the Manhattan distance\*\* function and the single-linkage method to do algglomerative hierarchical clustering of these data.

----------------------------------------------------------------------------------------------------------------

\*\* Manhattan distance (City-block distance):

This distance is simply the addition of all the distances across dimensions, calculated by the following function: Manhattan distance (A, B) =

---------------------------------------------------------------------------------------------------------------

The process is described as a sequence of the following steps:

**Step 1**

At the beginning, each element *Pi* is a cluster on its own, and therefore, the set of clusters are: *C* = {{*P*1}, {*P*2}, {*P*3}, {*P*4}, {*P*5}, {*P*6}, {*P*7}, {*P*8}}

**Step 2** (First iteration)

We seek 7to find the minimum Manhattan distance between the points. We can see that this value is 1 and occurs between the points *P*1, and *P*7 , and also between the points *P*2, and *P*4. Depending on how our minimum function works, we can choose either. Suppose we choose *P*1, and *P*7 arbitrarily. We then combine the two to form the cluster *P*1*P*7, and remove those as individual clusters from the set to get the new cluster set: *C1*= {{*P*2}, {*P*3}, {*P*4}, {*P*5}, {*P*6}, {P1, *P*7}, {*P*8}}

**Step 3** (Second iteration)

In the cluster set C1, the minimum Manhattan distance is 1 between *P*2, and *P*4. By combining *P*2, and *P*4 we get the cluster set: *C2*= {{*P*3}, {P2,*P*4}, {*P*5}, {*P*6}, {P1, *P*7}, {*P*8}}

By applying the single-linkage strategy, the closest point is used as the representative of the cluster and is used for the distance calculation between the cluster and other one.

**Step 4** (Third iteration)

In C2, the minimum distance is 2 (using single-linkage strategy). This is between *P*3, and *P*2*P*4, between *P*6, and *P*1*P*7, and between *P*8, and *P*1*P*7. Breaking the tie arbitrarily and selecting the cluster *P*2*P*3*P*4, we get: *C3*= {{P2,*P*3,*P*4}, {*P*5}, {*P*6}, {P1, *P*7}, {*P*8}}

**Step 5** (Fourth iteration)

In C3, the minimum distance is 2 (using single-linkage strategy) between *P*6, and *P*1*P*7, and between *P*8, and *P*1*P*7. Selecting *P*1*P*6*P*7 arbitrarily, we get:

*C4*= {{P2, *P*3,*P*4}, {*P*5}, {P1, *P*6, *P*7}, {*P*8}}

**Step 6** (Fifth iteration)

In C4, the minimum distance is 2 (using single-linkage strategy) between *P*8, and *P*1*P*6*P*7. Thus putting these into one cluster, we get:

*C5*= {{P2, *P*3,*P*4}, {*P*5}, {P1, *P*6, *P*7, *P*8}}

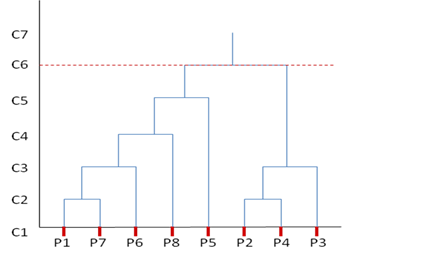
**Step 7** (Sixth iteration)

The minimum distance in C5 is 3 (using single-linkage strategy) between *P*5, and *P*1*P*6*P*7*P*8. Thus putting them in one cluster, we get:

*C6*= {{P2, *P*3,*P*4}, {P1, *P*5, *P*6, *P*7, *P*8}}

**Step 8** (Seventh iteration)

All points are in one cluster. The clustering process is complete.

The corresponding dendrogram formed from the hierarchy is presented here:

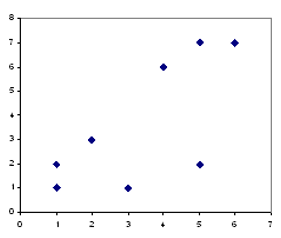
This demonstrates that in hierarchical clustering a nested set of clusters is created and each level in the hierarch has a separate set of clusters. At the root level, all items belong to one cluster, and at the leaf level each item is in its own unique cluster.

# Non-hierarchical Clustering (K-Means)

Let’s consider the same dataset of the previous example (P1~P8) to find two clusters using the k-means algorithm. The computation proceed as follows:

**Step 1**

Arbitrarily, let us choose two cluster centres to be the data points P5 (5,2) and P7 (1,2). Their relative positions can be seen in the figure below. We could have started with any other two points. The initial selection of points does not affect the final result.

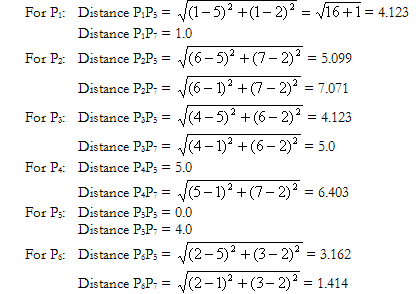


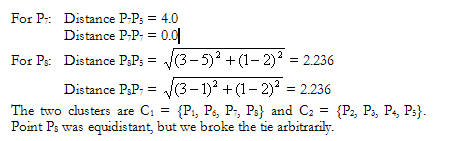
*P*5

*P*7

**Step 2**

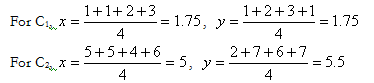
Let us find the Euclidean distances of all the data points from these two cluster centres.

****



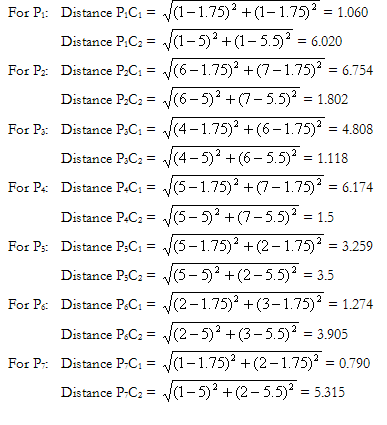
**Step 3**

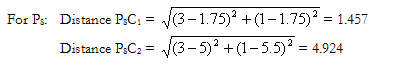
The new cluster centres are:

****

**Step 4**

The distances of all data points from these new cluster centres are:





**Step 5**

By the closest centre criteria P5 should be moved from C2 to C1, and the new clusters are C1={P1, P5, P6, P7, P8} and C2={P2, P3, P4}.

The new cluster centres are:

For C1,

For C2,

**Step 6**

If we repeat the computations of Step 4, we will find that no data point will switch clusters. Therefore, the iteration stops and the final clusters are C1={P1, P5, P6, P7, P8} and C2={P2, P3, P4}.

# Practice on Clustering

Draw the dendrogram for the first six points (P1, P2, P3, P4, P5, P6) of the example data set we used in this practical previously, after applying the hierarchical clustering algorithm (agglomerative).

# Laboratory Questions

Open bank\_data.csv in weka. Run Hierarchical clustering for both single and complete link. Compare the dendograms. Next, run K-means clustering for both ‘Euclidean’ and ‘Manhattan’ distance metric. Compare your results.

Submit your solution to LearnJCU.