Clustering Example Problems

Sakib Ashraf Zargar

Department of Mechanical and Aerospace Engineering, North Carolina State University, Raleigh, North Carolina 27606, USA

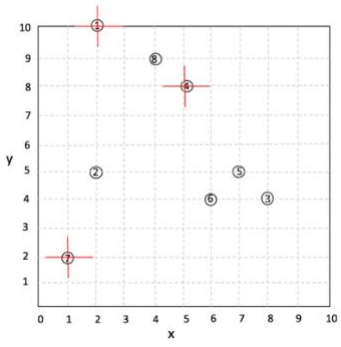
Problem no. (01): k-means clustering

(a) Why is k-means clustering considered sub-optimal?

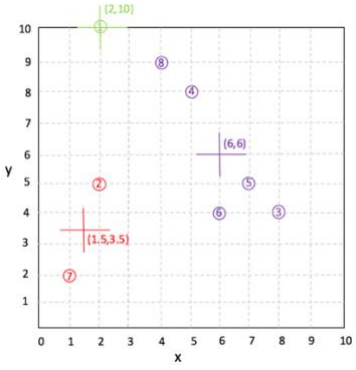
Because of the random initialization, there is a chance that k-means can get stuck in a local optimum i.e., there is no guarantee of it landing in the global optimum always, as such, sub-optimal.

(b) Based on the data points in the table below, cluster the points using k-means algorithm. The distance function is Euclidean distance. Suppose initial centroids are points with IDs 1, 4 and 7. Report: (i) The three cluster centers after the first iteration of execution. (ii) The final three clusters.

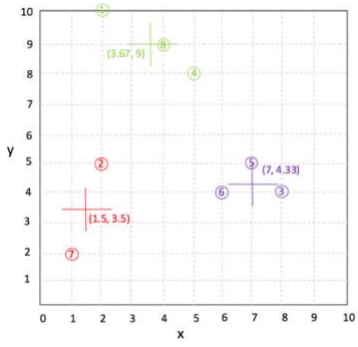
ID	1	2	3	4	5	6	7	8
x	2	2	8	5	7	6	1	4
y	10	5	4	8	5	4	2	9



Starting point



After the 1st iteration



Final clustering results

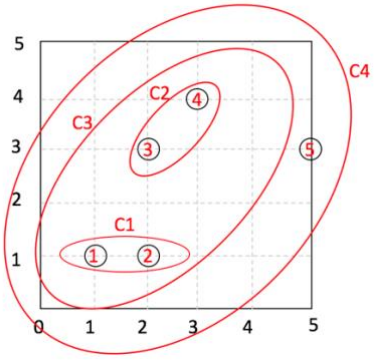
Problem no. (02): Hierarchical clustering

(a) Write down the algorithm for basic agglomerative hierarchical clustering in a way that would enable you to reason about its time complexity and then do so.

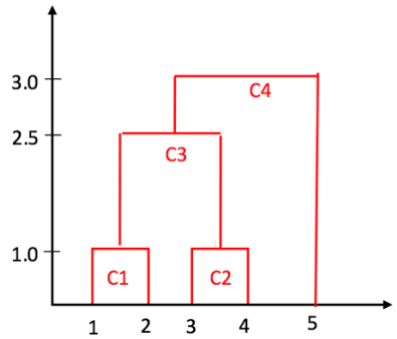
Steps	Time Complexity
1. Compute the proximity matrix	$O(m^2)$
Repeat:	$(m-1)$ iterations
2. Merge the closest two clusters.	$O((m-i+1)^2)$
3. Update the proximity matrix to reflect the proximity between the new cluster and the original clusters.	$O(\ m-i-1\ )$
	$O(m^3)$
Until only one cluster remains	
	$O(m^2 \log(m))$

(b) Perform agglomerative hierarchical clustering on the data in the table below. Use maximum norm distance function and centroid linkage criteria between sets of points. Merge only one pair of clusters in a step and resolve ties by merging sets containing points with smaller IDs first. Draw the final dendrogram *to scale*, be sure to label axes.

Nested Clusters



Dendogram



Distance matrix based on Maximum norm

	1	**2**	**3**	**4**	**5**
1	0	1	2	3	4
2	1	0	2	3	3
3	2	2	0	1	3
4	3	3	1	0	2
5	4	3	3	2	0

1st iteration

	1,2	**3**	**4**	**5**
1,2	0	2	3	3.5
3	2	0	1	3
4	3	1	0	2
5	3.5	3	2	0

2nd iteration

	1,2	**3,4**	**5**
1,2	0	2.5	3.5
3,4	2.5	0	2.5
5	3.5	2.5	0

3rd iteration

	1,2,3,4	**5**
1,2,3,4	0	3
5	3	0

(c) "The dendrogram structure can be cut at different levels to achieve different clustering results, each with a different number of clusters". What advantage does this provide over k-means clustering?

Since we can cut the dendogram at any level to achieve the desired number of clusters, we are the spared the task of choosing a suitable *k* in the beginning of the clustering process as is the case with k-means clustering.