CS 6140: Data Mining Assignment 4

Clustering

Yash Gangrade (u1143811), MS in Computing 20th February 2019

Contents

List of Figures

1 Hierarchical Clustering

1.1 Run all hierarchical clustering variants on data set C1.txt and identify the best one

Ans: We are using C1.txt as our dataset to perform hierarchical clustering using three variants until we reach $k = 4$ clusters. To view the clusters, dendograms and linkages in python were used. The table below summarizes all the clusters information (points corresponding to a particular cluster) from all the three variants.

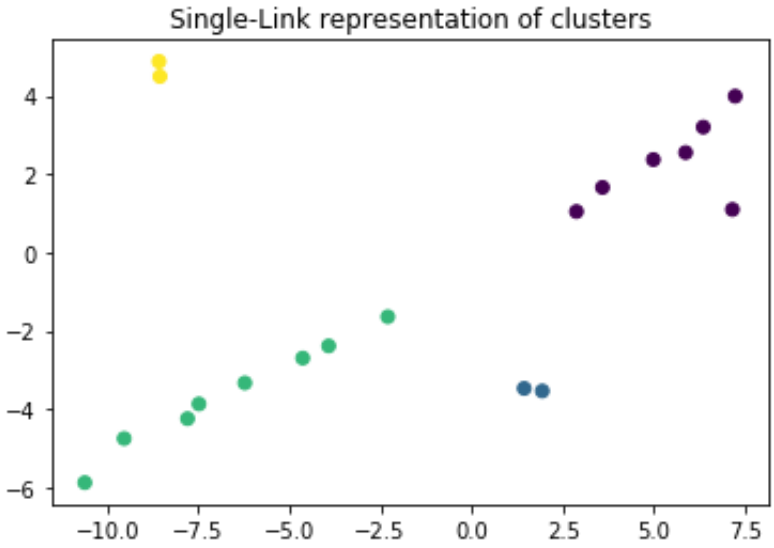
	Single-Link	**Complete-Link**	**Mean-Link**
Cluster 1	[1,3,8,10,12,16,19]	[6,13,14,15,18]	[2,5,9,11,13,15,17,18]
Cluster 2	[6,14]	[1,3,8,10,12,16,19]	[1,3,8,10,12,16,19]
Cluster 3	[2,5,9,11,13,15,17,18]	[2,5,9,11,17]	[4,7]
Cluster 4	[4,7]	[4,7]	[6,14]

The scatter plots are attached in the subsequent pages for each of the three variants. Dendograms were also created but not attached here (available if needed).

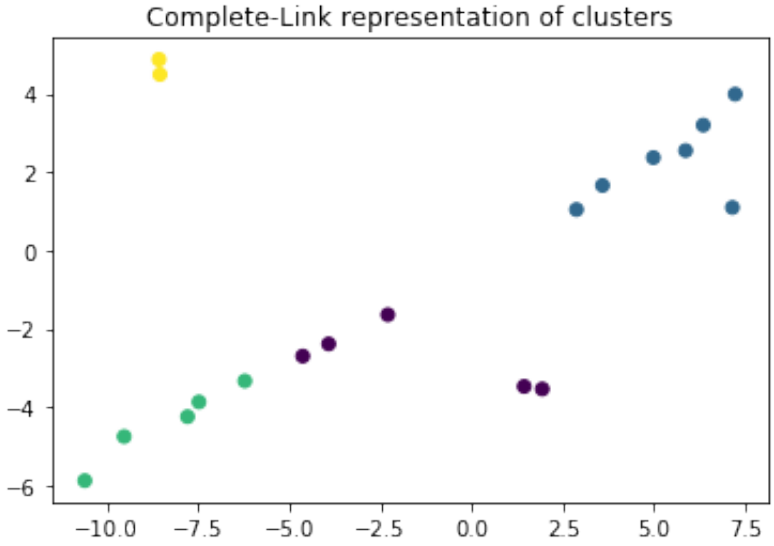
Best Variant: We can compare the time complexity of all the three variants below:

- **Single-Link:** This clustering has a time complexity of $O(n^2)$ because distances between each pair of points is being computed and at the same time, we are also keeping track of the smallest distance for each point and pushing it back to the next best cluster. Finally, we merge the two closest clusters and update the distance matrix according to it.
- **Complete-Link:** This clustering variant has a worst time complexity of $O(n^2 \log n)$. Here, we computer the n^2 distances and sort them for each point.
- **Mean-Link:** This clustering variant has a worst time complexity of $O(n^2 \log n)$. Here, we compute n^2 similarities for the clusters and then sort them for each of the cluster. We iteratively merge the pair of clusters which are more similar.

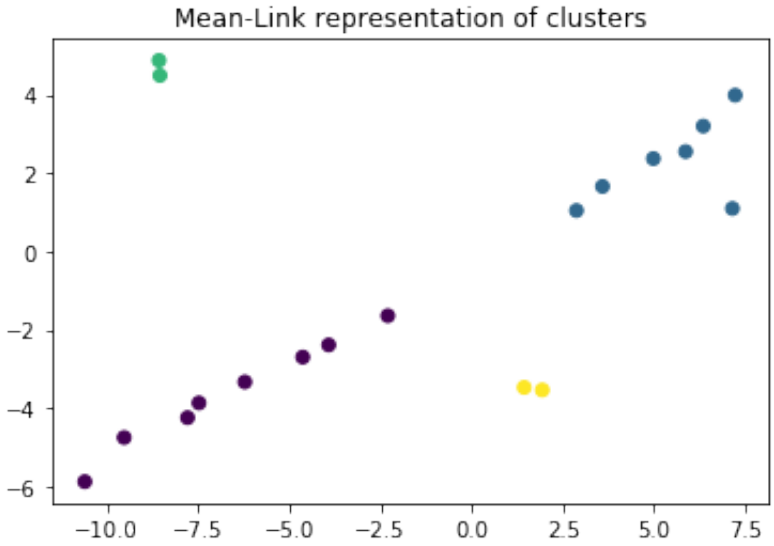
The time complexity of Single-Link clustering is the lowest among the three so it would be easiest to compute for huge datasets. In terms of performance, I would say Mean-Link and Single-Link both performs the best clustering because they performed intuitively correct clustering. Also, the mean cost comparison between the three variants also shows lowest cost for the mean-link clustering. Therefore mean-link has advantage in terms of min cost.



(a) Scatter Plot with Single-Link Clustering



(b) Scatter Plot with Complete-Link Clustering



(c) Scatter Plot with Mean-Link Clustering

Figure 1: Scatter Plots for each type of Clustering

2 Assignment-Based Clustering

2.1 Gonzalez

Following are the results from Gonzalez Algorithm. The center and the subset of the 3 clusters are given in below table while to better visualize the subset refer to the plot below.

Centers: [[-4.4357274, -5.6060004], [0.0, -40.0], [5.1914717, 6.9736254]]
3-center cost maximum: 7.7721092399481115
3-center mean maximum: 3.713989008072261

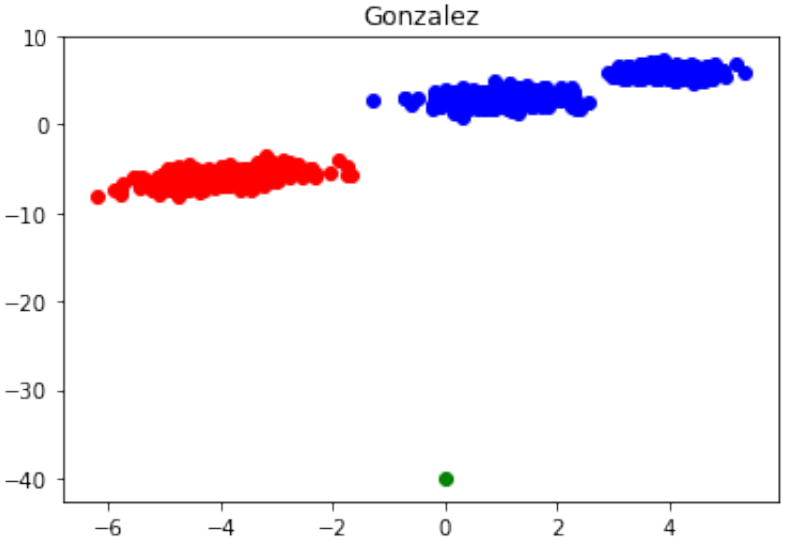


Figure 2: Gonzalez Clustering Algorithm

2.2 K-means++

Following are the results from k-means++ Algorithm. As recommended in the assignment, we ran several trials (20) of the algorithm to get the cumulative density function plot of the 3-means cost. All the times, the subsets doesn't match with the subsets from Gonzalez algorithm (0-fraction trials). The plot is shown below:

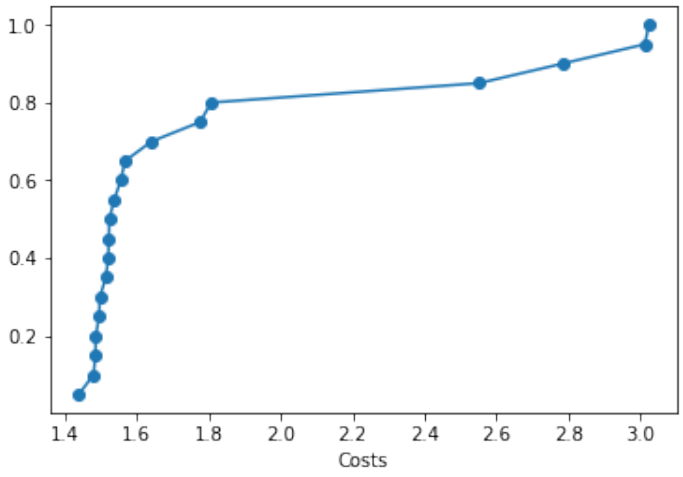


Figure 3: CDF of K-means++

2.3 Lloyd's algorithm for k-means clustering

2.3.1 Run Lloyds Algorithm with C initially with points indexed 1,2,3

The final cluster centers are: [[-3.5916, -5.5615], [-4.5567 -6.8961], [2.49627, 4.4708]].
The 3-means cost maximum is 2.2164314004616155.

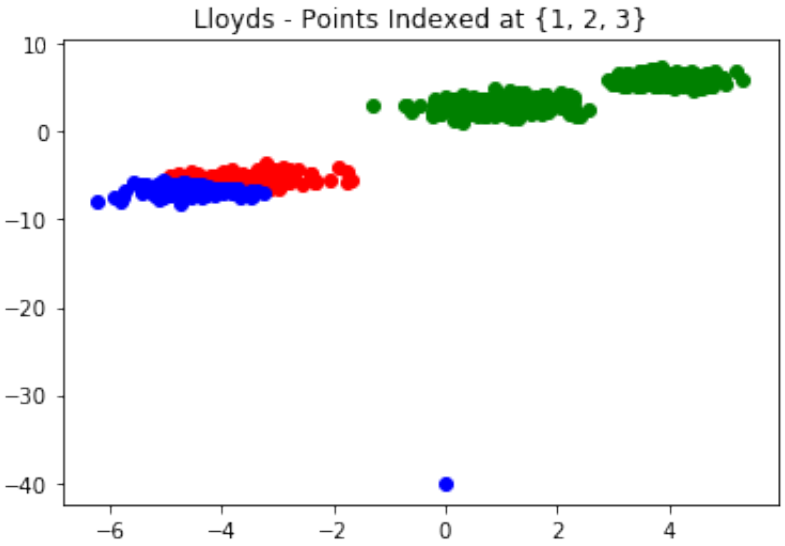


Figure 4: Clustering through Lloyds - Points indexed at 1,2,3

2.3.2 Run Lloyds Algorithm with C initially with points indexed at Gonzalez

The final cluster centers are: [[-4.0193, -6.02335], [2.49627, 4.4708], [0.0, -40.0]].
The 3-means cost maximum is 1.9964154472988538.

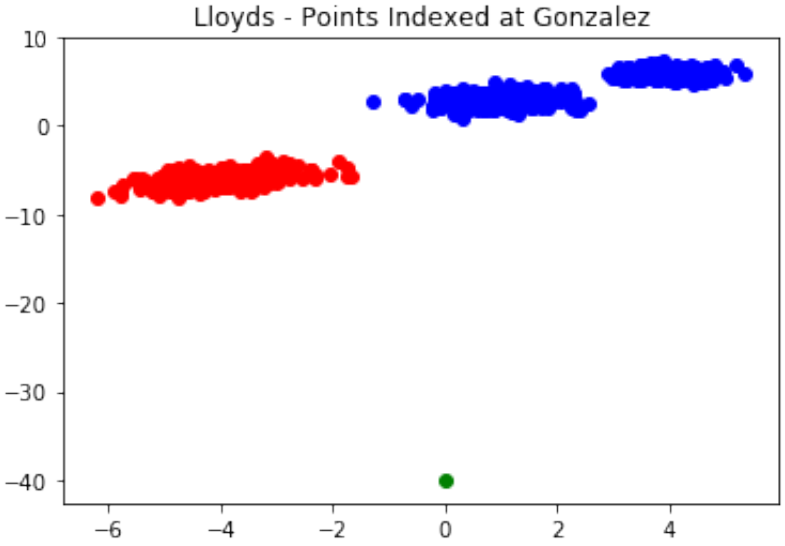
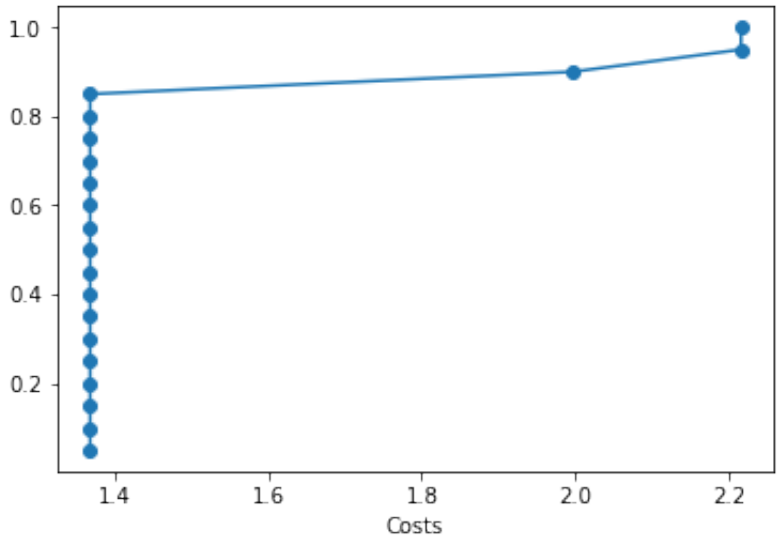


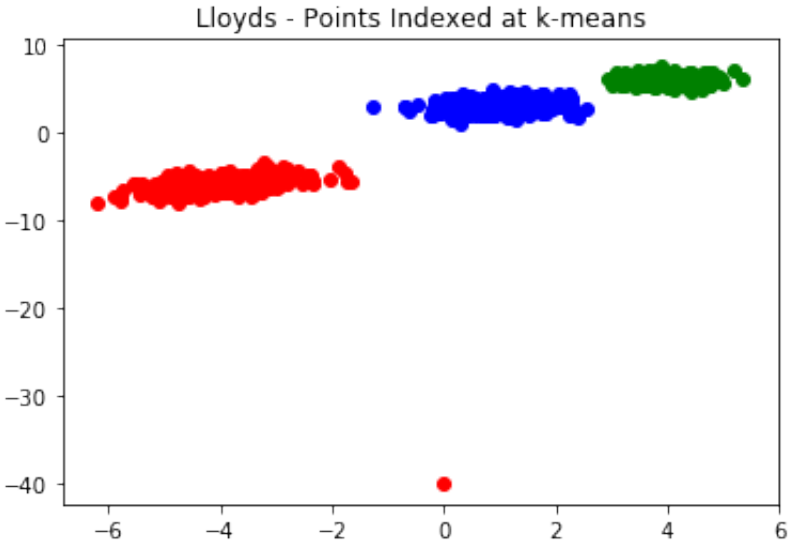
Figure 5: Clustering through Lloyds - Output of Gonzalez

2.3.3 Run Lloyds Algorithm with C initially as output of k-means++

We ran the Lloyds algorithm for 20 times and the figure below represents the final clustering at last trial and also the cdf plot of 3-means cost. The fraction of time that the subsets are the same results as Gonzalez is 1.



(a) CDF Plot of 3-means cost



(b) Scatter Plot with Clustering

Figure 6: CDF Plot and Final K-means Clustering

3 Bonus: k-Median Clustering

The centers of the 4-median are calculated and computed as follows:

Cluster 1: [1.076439, 0.0055612, 0.00450375, -0.0221662, 0.0071928]
Cluster 2: [-0.013918, 0.98616785, 0.0075185, 0.00345015, -0.003192]
Cluster 3: [0.010357, -0.0083566, 0.4989015, 0.4962178, 0.0096139]
Cluster 4: [-0.010534, -0.0004267, 0.0040935, -0.00381, 0.98873]

The final cost for 4-median is 0.4252717383445234.

Method:

- Choose initial k centers using any of the algorithms, Gonzalez Algorithm, Indexed (1,2,3,4), Randomly (in some cases), K-means++. Usually k-means++ is chosen as the initial algorithm.
- Next, to do the assignment step, we iterate over all the data points and then assign each of those points to the cluster in which the Manhattan distance between the data point and the cluster center is minimum. Sometimes, experiments are run multiple times to ensure the correctness and convergence of the clusters.
- We then calculate the new centers of the clusters. In order to do so, we take the the median of the points in that cluster and assign it as the new center.
- Keep repeating steps 2 and 3 till the clusters achieve convergence.