Michael Swenson

HW 4

1 Hierarchical Clustering (35 points)

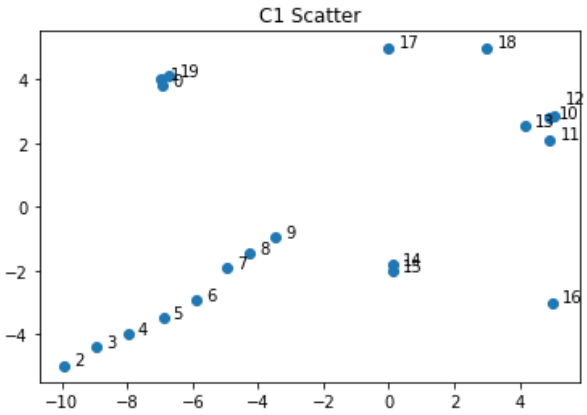
There are many variants of hierarchical clustering; here we explore 3. The key difference is how you measure the distance $d(S_1, S_2)$ between two clusters S_1 and S_2.

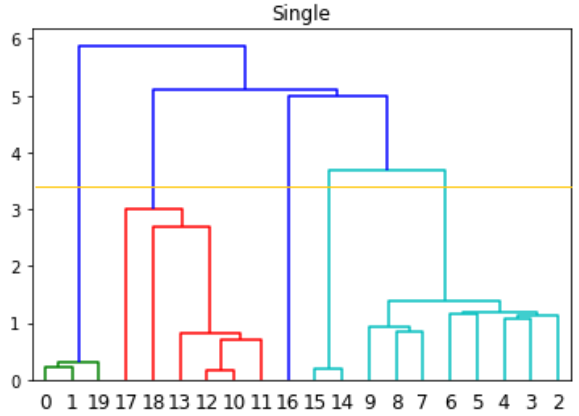
Single-Link: measures the shortest link $d(S_1, S_2) = \min_{(s_1,s_2)\in S_1\times S_2} \|s_1 - s_2\|_2$.

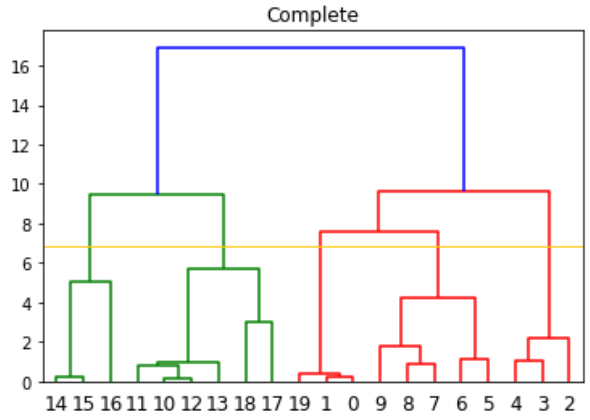
Complete-Link: measures the longest link $d(S_1, S_2) = \max_{(s_1,s_2)\in S_1\times S_2} \|s_1 - s_2\|_2$.

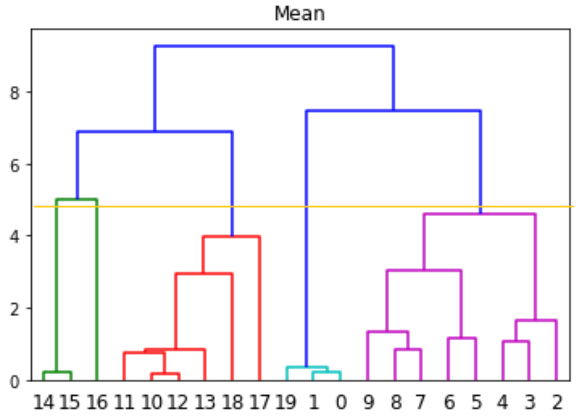
Mean-Link: measures the distances to the means. First compute $a_1 = \frac{1}{|S_1|}\sum_{s\in S_1} s$ and $a_2 = \frac{1}{|S_2|}\sum_{s\in S_2} s$ then $d(S_1, S_2) = \|a_1 - a_2\|_2$.

A (30 points): Run all hierarchical clustering variants on data set `C1.txt` until there are $k = 5$ clusters, and report the results as sets. It may be useful to do this pictorially.









B (5 points): Which variant did the best job, and which was the easiest to compute (think if the data was much larger)? Explain your answers.

I think Mean did the best job, based visually on the scatter plot. It handled the outlier and grouped the other appropriately. The least expensive to compute would be single, according to scipy documentation it can be implemented with a minimum spanning tree reducing the O(n^3) to O(n^2).

2 Assignment-Based Clustering (65 points)

Assignment-based clustering works by assigning every point $x \in X$ to the closest cluster centers C. Let $\phi_C : X \to C$ be this assignment map so that $\phi_C(x) = \arg\min_{c \in C} \mathbf{d}(x, c)$. All points that map to the same cluster center are in the same cluster.

Two good heuristics for this type of clustering are the Gonzalez (Algorithm 8.2.1 in M4D book) and k-Means++ (Algorithm 8.3.2) algorithms.

A: (15 points) Run Gonzalez and k-Means++ on data set `C2.txt` for $k = 4$. To avoid too much variation in the results, choose the point in the first line as c_1.

Report the centers and the subsets (as pictures) for Gonzalez. Report:

- the 4-center cost $\max_{x \in X} \mathbf{d}(x, \phi_C(x))$ and
- the 4-means cost $\sqrt{\frac{1}{|X|} \sum_{x \in X} (\mathbf{d}(x, \phi_C(x)))^2}$
  (Note this has been normalized so easy to compare to 4-center cost)

Cluster Centroids:

[13.51372985, 45.03355641]

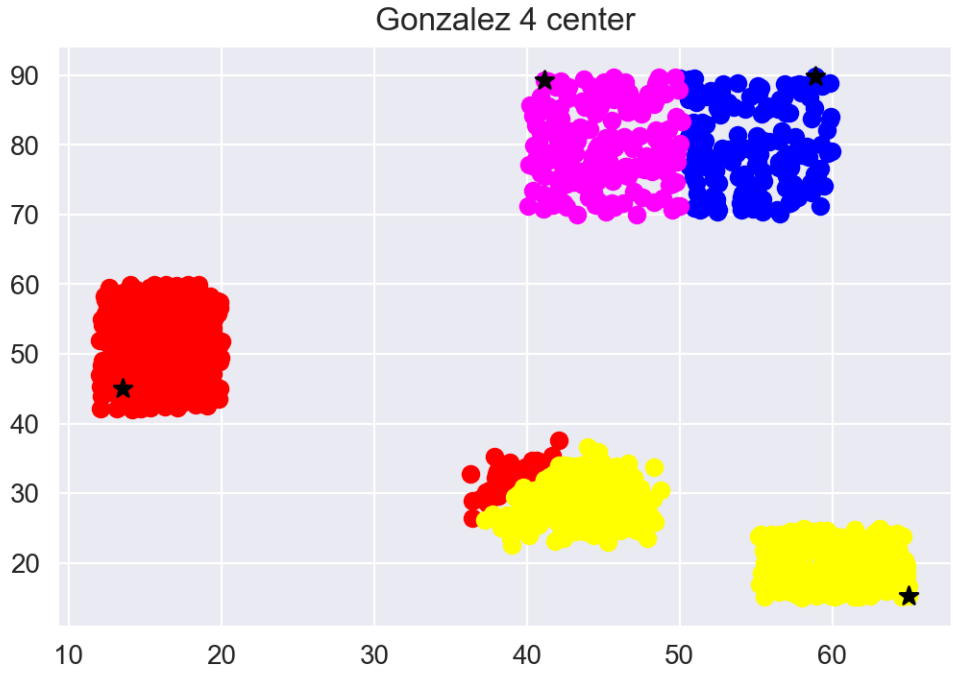
[58.90178825, 89.90968034]

[64.97202851, 15.22366739]

[41.14955242, 89.2626484]

4-max cost: 29.998682717866146

4-means cost: 15.144353586376345



From 1 trial

Starting Centers

[13.51372985 , 45.03355641]

[56.5490188 , 22.8070878]

[49.11165447, 76.05052155]

[44.3621677 , 31.999818]

Trial: 0

Max Cost: 60.801053195882346

Mean Cost: 0.9327527051675946

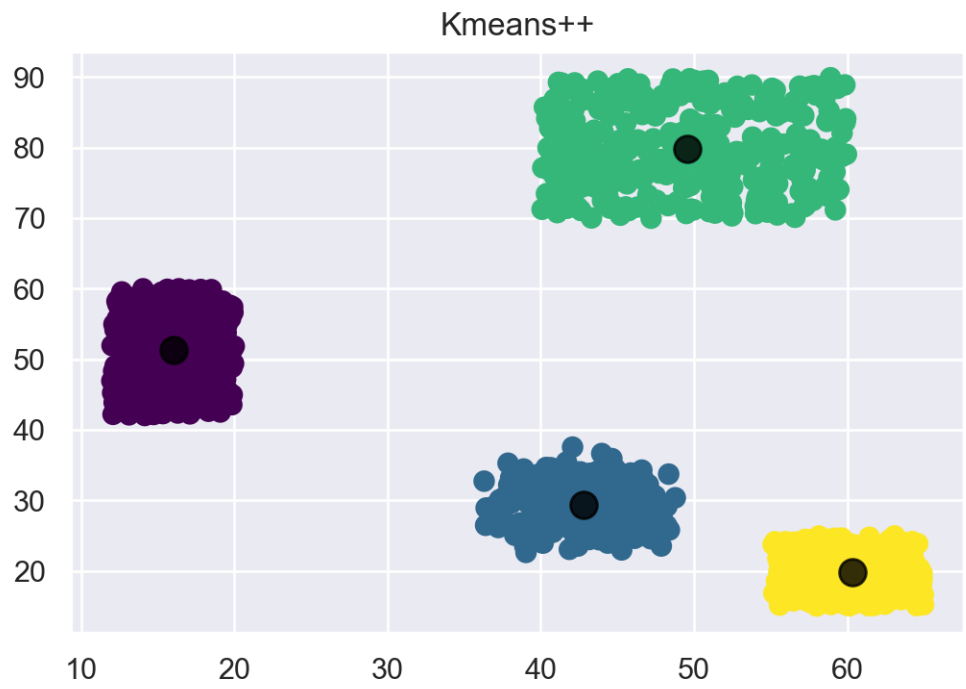
Final Centers:

[16.05120967, 51.32798228],

[60.31142102, 19.83358635],

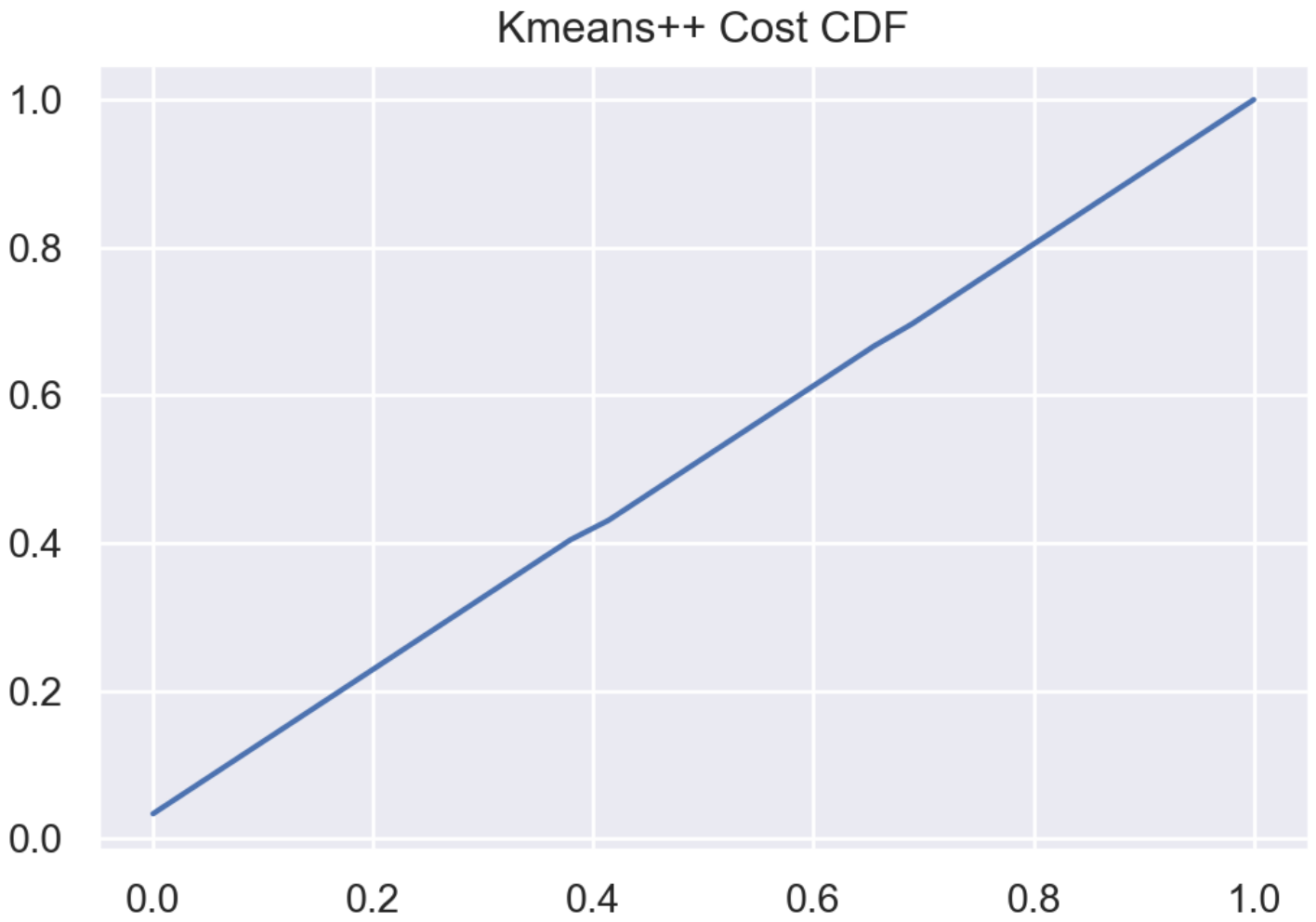
[49.53799554, 79.83826452],

[42.81360295, 29.34661297]



B: (20 points) For k-Means++, the algorithm is randomized, so you will need to report the variation in this algorithm. Run it several trials (at least 20) and plot the *cumulative density function* of the 4-means cost.

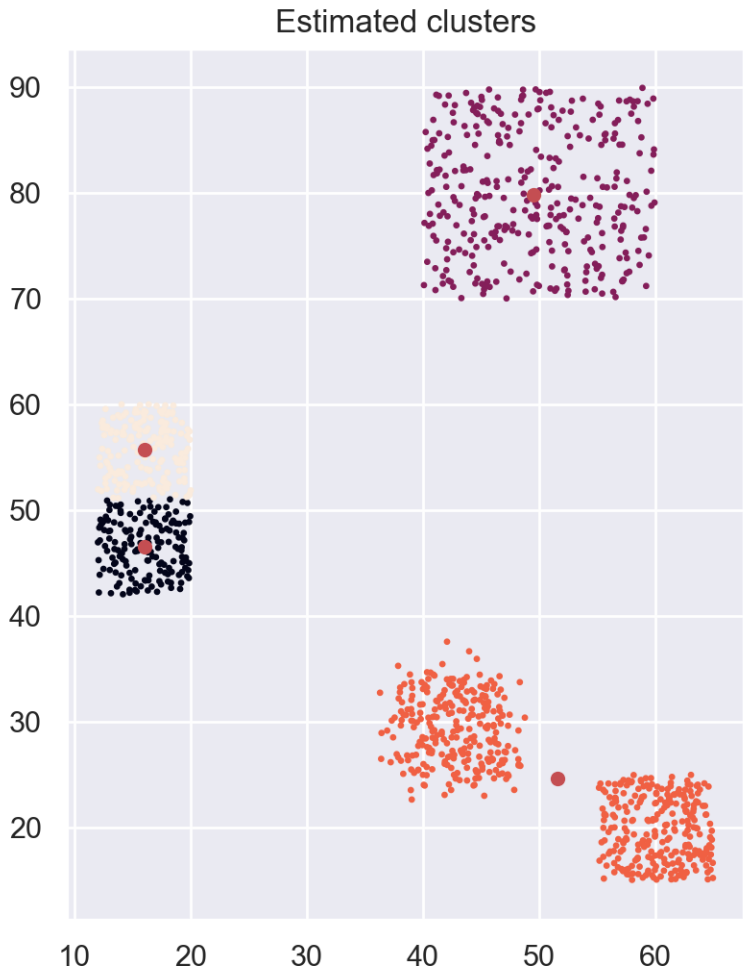
30 trials



C: (30 points) Recall that Lloyd's algorithm for k-means clustering starts with a set of k centers C and runs as described in Algorithm 8.3.1 (in M4D).

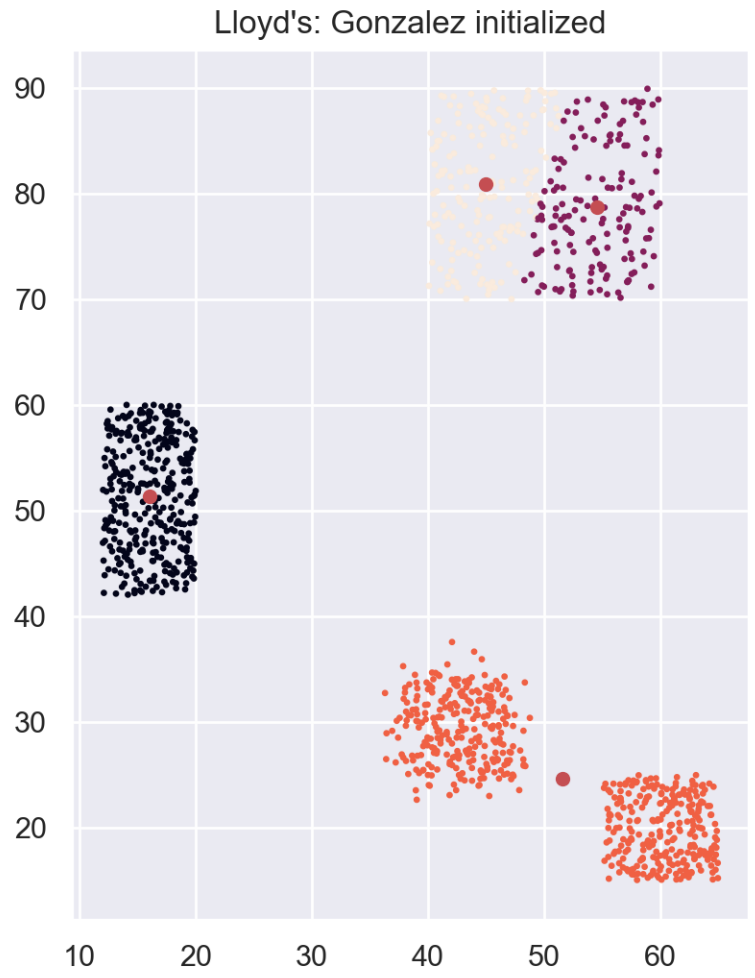
1: Run Lloyds Algorithm with C initially with points in the $1st, 2nd, 3rd, 4th$ lines (the first 4 points). Report the final subset and the 4-means cost.

Inertia: 73.17636670004015



2: Run Lloyds Algorithm with C initially as the output of Gonzalez above. Report the final subset and the 4-means cost.

Inertia: 72.21745623978002



3: Run Lloyds Algorithm with C initially as the output of each run of k-Means++ above. Plot a *cumulative density function* of the 4-means cost. Also report the fraction of the trials that the subsets are the same as the input (where the input is the result of k-Means++).

16/20 were the same : 75%