Asmt 4: Clustering

Turn in through Canvas by 2:45pm:
Wednesday, February 19
100 points

Overview

In this assignment you will explore clustering: hierarchical and point-assignment. You will also experiment with high dimensional data.

You will use three data sets for this assignment:

- `http://www.cs.utah.edu/~jeffp/teaching/cs5140/A4/C1.txt`
- `http://www.cs.utah.edu/~jeffp/teaching/cs5140/A4/C2.txt`
- `http://www.cs.utah.edu/~jeffp/teaching/cs5140/A4/C3.txt`

These data sets all have the following format. Each line is a data point. The lines have either 3 or 6 tab separated items. The first one is an integer describing the index of the points. The next 2 (or 5 for `C3`) are the coordinates of the data point. `C1` and `C2` are in 2 dimensions, and `C3` is in 5 dimensions. `C1` should have n=19 points, `C2` should have n=1040 points, and `C3` should have n=1000 points. We will always measure distance with Euclidean distance.

It is recommended that you use LaTeX for this assignment (or other option that can properly digitally render math). If you do not, you may lose points if your assignment is difficult to read or hard to follow. Find a sample form in this directory: ***`http://www.cs.utah.edu/~jeffp/teaching/latex/`***

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 $k = 4$ clusters, and report the results as sets. It may be useful to do this pictorially.
For single link cluster, the data points are clustered as follow as :

$$Single\ link\ [1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 0, 0, 3, 2, 2]$$

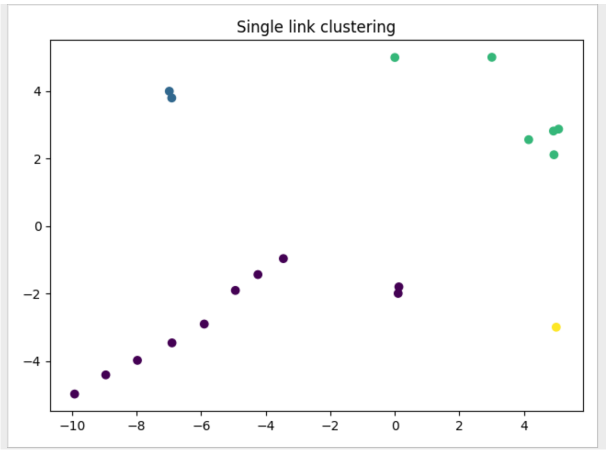

Figure 1: Single Link HAC

For Complete Link, the data points are clustered as follow as,

$$Complete\ link\ [0, 0, 2, 2, 2, 0, 0, 0, 0, 0, 1, 1, 1, 1, 3, 3, 3, 1, 1]$$

For Mean Link, the data points are clustered as follow as,

$$Mean\ link\ [2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 3, 3, 3, 3, 0, 0, 0, 3, 3]$$

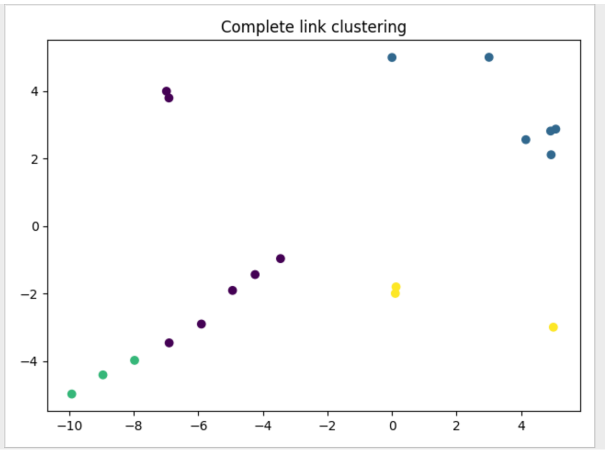

Figure 2: Complete Link HAC

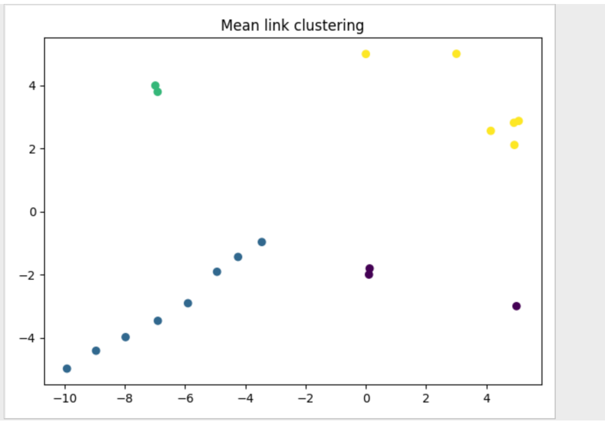

Figure 3: Mean Link HAC

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.

According to the above experiment, Mean Link clustering did the best job for given dataset as it seems to cluster data points better from visual perspective. Also it has the less mean cost compared to other types. For other two(single and complete) we have to compute the distance between each pair of points present in the corresponding clusters, in order to assign a cluster in each iteration. However, In case of mean link this work is drastically decreased as we only need to compute the distance between the means of the corresponding clusters.

Single link clustering performs the best of all and it is easy to compute. Time complexity is (n^2). To compute all the distances it takes $O(n^2)$ time. In each of the n1 merging steps, we then find the smallest distance in the next-best-merge array. We merge the two identified clusters, and update the distance matrix in O(n). We update the next-best-merge array in $O(n)$ in each step.

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 $\phi_C(x) = \arg\min_{c \in C}(x, c)$. 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 $k = 3$. To avoid too much variation in the results, choose c_1 as the point with index `1`.

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

$$Centers$$

$$C1 : [13.51372985, 45.03355641]$$

$$C2 : [115, 95]$$

$$C3 : [55.09667203, 88.41858189]$$

- the 3-center cost $\max_{x \in X}(x, \phi_C(x))$
  The 3-center cost max is 59.469201975675944

- the 3-means cost

  $\sqrt{\frac{1}{|X|}\sum_{x \in X}((x, \phi_C(x)))^2}$

  The 3-means cost max is 32.62157656815563

  Below is the plt of Gonzalez clustering. Points marked ax 'x' are the centroids of each cluster.

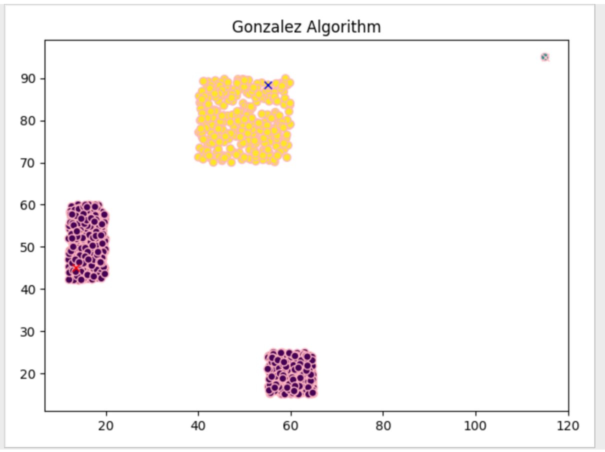

Figure 4: Clustering using Gonzalez

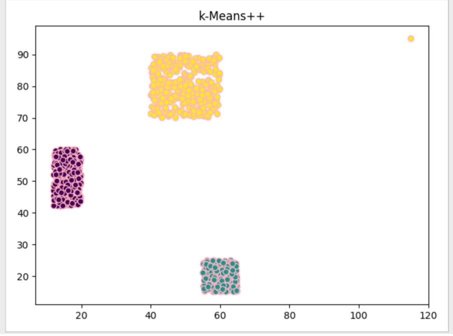

Figure 5: Clustering using k-Means++

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 3-means cost. Also report what fraction of the time the subsets are the same as the result from Gonzalez.

I did the experiment several times with 50 runs each time. Subsets never matched with results from Gonzalez. Hence, the fraction is zero.

Please refer figure 6.

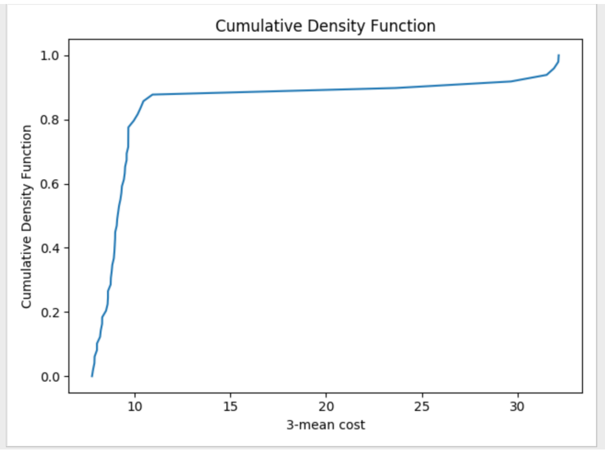

Figure 6: Clustering using k-Means++

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 indexed {1,2,3}. Report the final subset and the 3-means cost.

Final subsets are [[16.05120967, 51.32798228] [49.73053084, 79.88285786] [60.2816089, 19.80587656]] 3-mean cost for k-means with cluster centers as indexes 1,2,3 is 6.476439964596597

2: Run Lloyds Algorithm with C initially as the output of Gonzalez above. Report the final subset and the 3-means cost.

Final subsets are [[38.79827213, 35.11661363], [115, 95],[49.53799554,

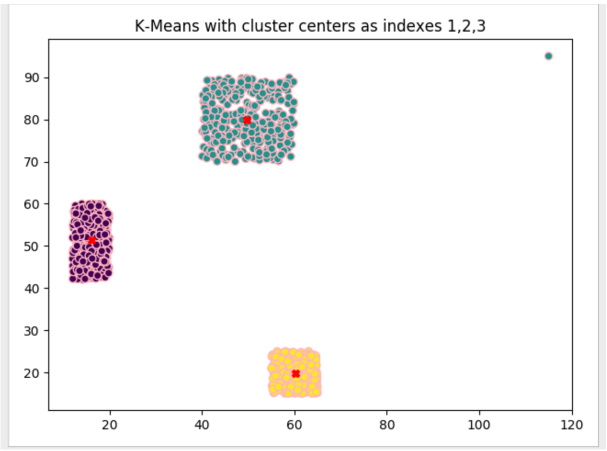

Figure 7: Clustering using Lloyd's

79.83826452]]
3-mean cost for k-means with cluster centers as Gonzalez's output is 23.099778090758615

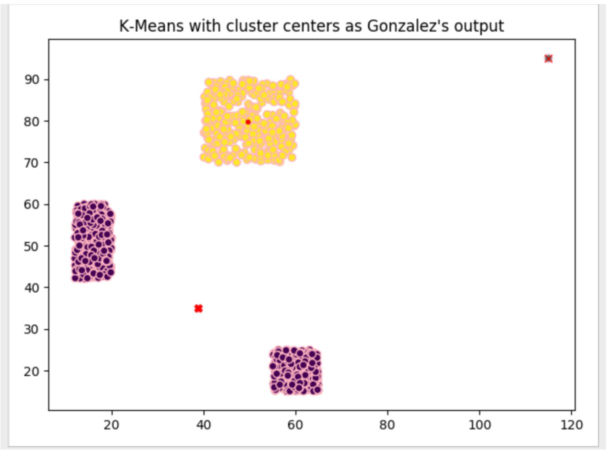

Figure 8: k-Means with cluster centers as Gonzalez's outputt

3: Run Lloyds Algorithm with C initially as the output of each run of k-Means++ above. Plot a *cumulative density function* of the 3-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++).

Fraction of trials that the subset are same as input is 1. Or in other words, all the time subsets were same as inputs.

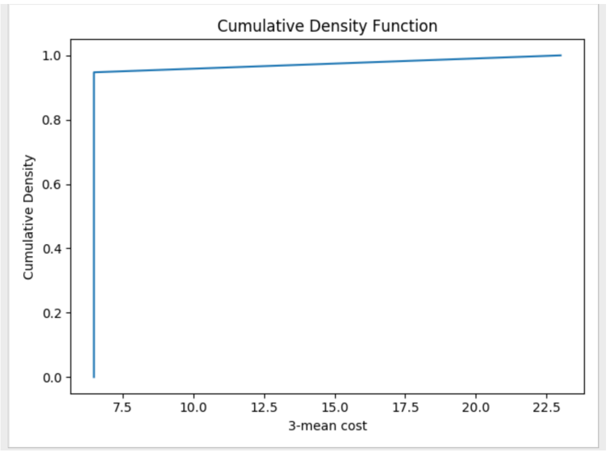

Figure 9: Cumulative Density Function of 3-Means cost