ECE 551 Homework 3 (Spring 2016)

Due date is May 13th, 2016 11:59pm

1. Written[25 pts]

Show the hierarchical bottom-up clustering of the following data points using Euclidean distance.
6, 8, 10, 20, 26, 30, 32, 33

a) use single linkage. Draw the resulting dendrogram.

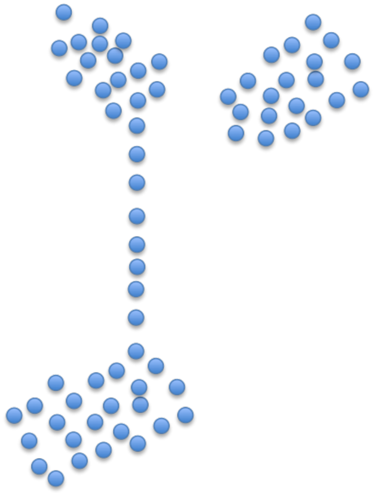
b) use complete linkage. Draw the resulting dendrogram.

c) use average linkage. Draw the resulting dendrogram.

2. Written[25 pts]

a) Cluster the following data points (single dimensional) with Euclidean distance and complete linkage. If you were to cluster these data points into two clusters by eye, would it be the same with the clustering resulting from complete linkage?
1.2, 4, 5.2, 6, 6.9, 7.5.

b) If the data points in the figure below are clustered with Euclidean distance and single linkage, how would the clustering look like if the hierarchical tree is cut so that there are 2 clusters? It'd be sufficient if you can only circle the two clusters. Explain why single linkage results in clusters like this.



3. Programming[50 pts]

This problem is related to K-means clustering algorithm. Implement K-means using any programming language (you can also modify an existing implementation that you find from the internet if you wish). Your code should:

i. Take a tab-delimited file with an N x 2 matrix of N data points with 2 features each. In other words, each row is a data point and each column is a feature value associated with that data point.

ii. Accept K (the number of clusters) as a parameter.

iii. Output the 2-dimensional mean vectors for each of the K clusters, and also an assignment for each clustered data point. iv. You can define the stopping criteria as you wish.

Note: Use generate-clusters.py to make testing input data. You will find that the easiest data to cluster will be spherical (unit variance in both dimensions) and well separated (cluster means farther than 2 standard deviations apart). Run the script with no arguments to display its help information.

i. Generate N = 50 points from 3 Gaussians (150 points total). Generate a data set that is difficult to cluster and requires a nontrivial solution. For example, means should be within 2 standard deviations of each other and one of the Gaussians should be elongated, not spherical. Run both algorithms on this data with K = 2,3 and 4 cluster centers. Discuss what happens when the number of cluster centers is not 3. Include the parameters chosen for data generation.

ii. K-means is a greedy algorithm and is not guaranteed to find the global minimum of this objective. In fact, the quality of the resulting solution often depends significantly on the cluster initializations. For K = 3, repeat K-means 100 times with different random cluster initializations, and report the best mean values (i.e., which minimize the objective). How much variation was there?

[Bonus, 10 pts]

To visually assess the accuracy of your algorithms for the runs above, include at least one clustering plot for each run. The clustering plot should show a scatter of the data points and indicate BOTH the "correct" and "predicted" cluster for each point. For example, each point might be drawn as a *, +, or - according to the correct cluster, and colored red, green, or blue according to your predicted cluster. (Note that the "correct" cluster is provided as a third column in the output of generate-clusters.py.) Optionally, you may want to also plot the final predicted cluster centroids. (You do not need t o include any code for generating plots, just the plots themselves)