Melinda Vigh

CS 422

Assignment 3 – K-means

1. (2)  Next, look at the code in the file named “bisecting\_K\_means.py”. There is a set of observations named X provided in the file. Make a scatter plot of the observations. (15 pts)

A graph with red and blue dots

Description automatically generated

1. (3)  Decide how many clusters do you think there ‘should be’. Hint: it’s 4. But briefly explain why. It is not a trick question. You will need to answer the question (2) to answer this. (15 pts)

There should be 4 different clusters. The data points seem to be conveniently located in distinct groups near the 4 corners of the graph, appearing at the first glance to have 4 distinct data regions. The data in the X-array from the file seem to have every 4 data points located together in the 4 regions, thus indicating to us that there should be 4 different clusters.

1. (4)  Finish the code in the file (50 pts) so that you can “report” (as in clearly write it down or indicate it in a report) what the sub-cluster with the bigger SSE consists of (20 pts). That is, when the number of cluster (called current\_cluster in the file) equals 4, the while loop is going to terminate. But if you were to continue bisecting, what would be the sub- cluster you would bisect? That sub-cluster consists of only 3 observations. Please write down those three observations in your report. Hint: the variable sse defined in the while loop is a list that stores the SSE of the two clusters you are looking at. You will need to compare their SSE and decide which cluster you need to further divide.

When the initial split happens, the data is split vertically with the datapoints located on the left side having a higher SSE (157.375) than the data on the right side (127.09375). Thus, we would take the cluster with the bigger SSE and focus on it as the original dataset, proceeding to bisect it next to determine other sub-clusters. The program would continue bisecting the cluster with the bigger SSE value until the location of the centroids converge and they do not move anymore.

Observations:

Top left corner 🡪[[1, 10][2,11][1.5, 9][1, 10.5]]

Bottom left corner 🡪[[1,2][2,1][1,1.5][1.5,1]]

Entire right side (initial smaller SSE that was not further split) 🡪[[10, 2][10, 4][10,0][10,1][10.5,9][9,9.5][9.5,9][10,10]]