ENGR – 421 HW-8

Berke Can Rizai – 69282

First, we import the libraries,

Graphical user interface, text, application

Description automatically generated

Scipy.spatial.distance is used for the distance matrix between each points.

Graphical user interface, text, application, Word

Description automatically generated

We import the data and calculate the distance matrix with cdist.

Graphical user interface, text, application

Description automatically generated

Here, we copy the Distance matrix to B. Than, if the distances are above epsilon, they are zeroed since they are not connected and otherwise, they are 1.

For the connectivity matrix, for each entry in B that is 1, I draw a line between the two corresponding points from the B matrix. I look up x and y coordinates from data\_set.

Chart, scatter chart

Description automatically generated

Then, plot the points with

plt.plot(data\_set[:, 0], data\_set[:, 1], "k.", markersize=10)

than show the plot.

I construct the D matrix that is Dmx with the diagonal entries that are sum of each row in B matrix.



D matrix is as follows.

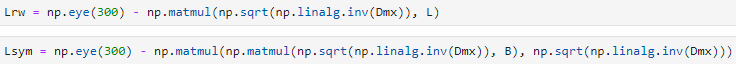
Scatter chart

Description automatically generated

Then create the L,



Initialize the L random walk and L symmetric with, np.eye creates identity matrix and I take the inverse squared root of D, multiply with B and then D again.



Lsym is as follows.

A picture containing text

Description automatically generated

For the Z matrix, I look up the eigen vectors and eigen values with np.linalg.eig.

The 0.th index is array of eigenvalues and 1st is eigenvectors.

I sort the eigenvalues with argsort that returns the indices sorted and with these indices, I look up the eigenvectors and store them in the Z. Only between 1-6 are taken since we need 5 of them and 0th one is 0.



Z for the given dataset is.

Graphical user interface, text

Description automatically generated

I initialize number of clusters and the array X which we will run the k-means algorithm.

Logo

Description automatically generated with medium confidence

I initialize centroids initial values in centr, with stacked indices.



For the k-means algorithm, I copied the algorithm from the lab,

Graphical user interface, text

Description automatically generated

Difference is in update\_centroids, if memberships are uninitialized, centroids are initialized to centr.

In the algorithm, it is same with lab except I removed the if that checks if old\_centroids and centroids are same because it would stop at first iteration.

Text

Description automatically generated

When we run it,

k-means in Z looks like,

Chart, scatter chart

Description automatically generated

And at the end,

I plot the memberships, before that, I update the centroids because they are means for the Z matrix and we need them for normal dataset.

Chart, scatter chart

Description automatically generated

Which concludes the clustering.