Kerem Anar

ENGR 421: Intro. To Machine Learning

HW\_08 Report

Part 1)

I read the data from the given file. The visualization of the given data can be seen in Figure 1.

Chart, scatter chart

Description automatically generated

Figure 1: Given X data set.

Part 2)

To calculate the Euclidian distance between the pairs of data points, I defined the Euclidian\_Distance(x1, x2) function for 2 dimensional x1 and x2 vectors. As it is given in the PDF, I set the threshold parameter to 1.25. Then I constructed the connectivity matrix B by simply 2 for loops as the following:

Text

Description automatically generated

Figure 2: Constructing Connectivity Matrix B.

Visualization of connectivity matrix B and the data set X can be seen in Figure 3:

Chart, diagram, scatter chart

Description automatically generated

Figure 3: Visualization of Connectivity Matrix B.

Part 3)

In this part I calculated matrix D, , L and Lsym according to Lecture 23 notes.

Text

Description automatically generatedText

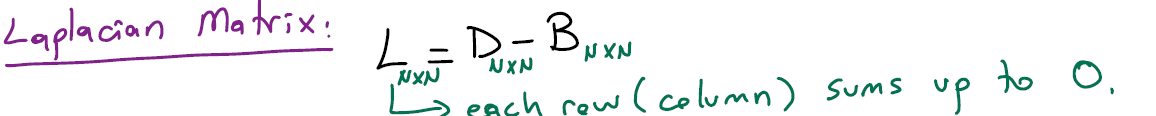
Description automatically generated

🡪

Figure : Constructing D matrix

Text

Description automatically generatedNormally we should check whether any diagonal of D is zero or not. If there is at least 1 zero diagonal, then D is not invertible which means that there is a point which doesn't have any neighbor. In our case, all diagonals of D is nonzero.

Text, letter

Description automatically generated

🡪

Figure : Constructing Laplacian Matrix and its normalized version.

Finding the square root of a matrix is a little bit complicated process. I though using Cholesky Factorization to find square root of D. However, D is diagonal so the inverse and square root of D is simply the power of (-1/2) of diagonals. As we can see in Figure 5, I calculated Laplacian matrix and its normalized version by using Lsymetric formula from the lecture notes.

Part 4)

In this part, I found the eigenvalues and corresponding eigenvectors by simply using the eig function from numpy.linalg. Then, I found the np.argsort(E) function which sorts the matrix E and returns the index numbers in sorted way. I took first R+1 index because the lowest eigenvalue is zero. The first R nonzero lowest eigenvalues indexes assigned to index array. Then, I constructed Z matrix by using the corresponding eigenvectors of these the lowest R eigenvalues.

Part 5)

In this part, I simply applied k-means clustering algorithm on Z matrix. I used same codes from the lab11 except the initialization step of the centroids. I assigned the asked rows in the PDF as initial centroids. When we work on Z matrix our centroids is 5x5 because Z is kind of data which has 5 (R) features and we want to divide it into 5 (K) clusters. In a while loop, I updated and found best memberships and centroids for Z matrix. The memberships will be same for the X data set but we need to find centroids for the X data set specificely because the centroids of X data set should be in the form 5x2 (K x #of feature of X). Therefore, I updated centroids one more time by using final memberships and X data set.

A picture containing text

Description automatically generated

Figure 6: Running K-means clustering algorithm on Z matrix.

Part 6)

Finally, I plotted the clustering results as we did in LAB 11.

Chart, scatter chart

Description automatically generated