CSE 881: Data Mining (Spring 2020) Homework 2

1) Consider the following set of one-dimensional data points: {0.1, 0.2, 0.8, 0.9, 1.0, 1.3, 1.8, 1.9}.

(a) Suppose we apply kmeans clustering to obtain three clusters, A, B, and C. If the initial centroids are located at {0, 0.3, 2.0}, respectively, show the cluster assignments and locations of the centroids after the first three iterations by filling out the following table.

	Cluster assignment of data points								Centroid Locations		
Iter	0.10	0.20	0.80	0.90	1.00	1.30	1.80	1.90	A	B	C
0	-	-	-	-	-	-	-	-	0.00	0.30	2.00
1											
2											
3											

(b) Compute the SSE of the k-means solution (after 3 iterations).

(c) Apply bisecting k-means (with k=3) on the data. First, apply k-means to bisect the data into 2 clusters using the initial centroids located at 0 and 2, respectively.

	Cluster assignment of data points								Centroid	
Iter	0.10	0.20	0.80	0.90	1.00	1.30	1.80	1.90	A	B
0	-	-	-	-	-	-	-	-	0.00	2.00
1										
2										

Next, compute the SSE for each cluster (make sure you indicate the SSE values in your answer). Choose the cluster with larger SSE value and split it further into 2 sub-clusters. You can choose the two data points with the smallest and largest values as your initial centroids. For example, if the cluster to be split contains data points (1.00, 1.30, 1.80, 1.90), then the centroids should be initialized to 1.00 and 1.90. Show the clustering solution produced after applying bisecting k-means.

(d) Compare the results of k-means clustering against bisecting k-means. Which clustering method is more effective for the given data set?

2) Consider a data set $\mathcal{D} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$ that contains N points, where each data point $\mathbf{x}_i$ is a p-dimensional vector of continuous-valued attributes. Suppose the N data points are grouped into two clusters, C_1 and C_2 using k-means clustering. Show that the SSE is non-increasing when the data is split (from 1 cluster containing all N points) into 2 clusters.

3)

Download the S&P-500 stock market time series data from the class web page. There are 3 files provided:

- *prices.txt*, which contains the normalized closing prices of the stocks from January 1, 2007 until December 31, 2012.
- *sp500.class*, which contains the category ID of each stock. There are 10 distinct categories.
- *classes.txt*, which contains the mapping from category ID to category name.

In this exercise, you will investigate the feasibility of applying k-means clustering algorithm to the data.

(a) Load the prices.txt data into Matlab.

(b) Which proximity measure do you think is more appropriate to cluster the data—Euclidean distance or correlation? Explain why.

(c) Run k-means clustering with $k = 10$ using Euclidean as proximity measure. Type `help kmeans` to determine how to set the appropriate measure. To ensure repeatability of your results, use 1 as the seed for your random number generator. Use the k-means setting of replicates=500 to repeat the experiment 500 times with different initial centroids.

```
matlab> rng(1);
matlab> [clusters, centroids] = kmeans( ... );
```

(d) Compute the 10×10 confusion matrix (using the stock categories as ground truth). Type `help confusionmat` to determine how to create the confusion matrix.

(e) Repeat the k-means clustering using correlation as proximity measure (with 500 replicates). Make sure you set the seed to 1 before applying k-means.

```
matlab> rng(1);
matlab> [clusters, centroids] = kmeans( ... );
```

(f) Compute the 10×10 confusion matrix (using the stock categories as ground truth). Compare the results against Euclidean distance. Which proximity measure is better for the data set?

4) Use the distance matrix shown in the table below to perform single and complete link hierarchical clustering. Show your results by drawing a dendrogram. The dendrogram should clearly show the order in which the points are merged and the y-axis show the distance between pairs of clusters being merged at each iteration.

	p1	p2	p3	p4	p5
p1	0	0.5840	0.1955	0.3815	0.1127
p2	0.5840	0	0.6132	0.4956	0.5733
p3	0.1955	0.6132	0	0.2390	0.3067
p4	0.3815	0.4956	0.2390	0	0.4694
p5	0.1127	0.5733	0.3067	0.4694	0

5) Consider the data set shown in Figure 1. Suppose we apply DBScan algorithm with Eps = 0.15 (in Euclidean distance) and MinPts = 3.

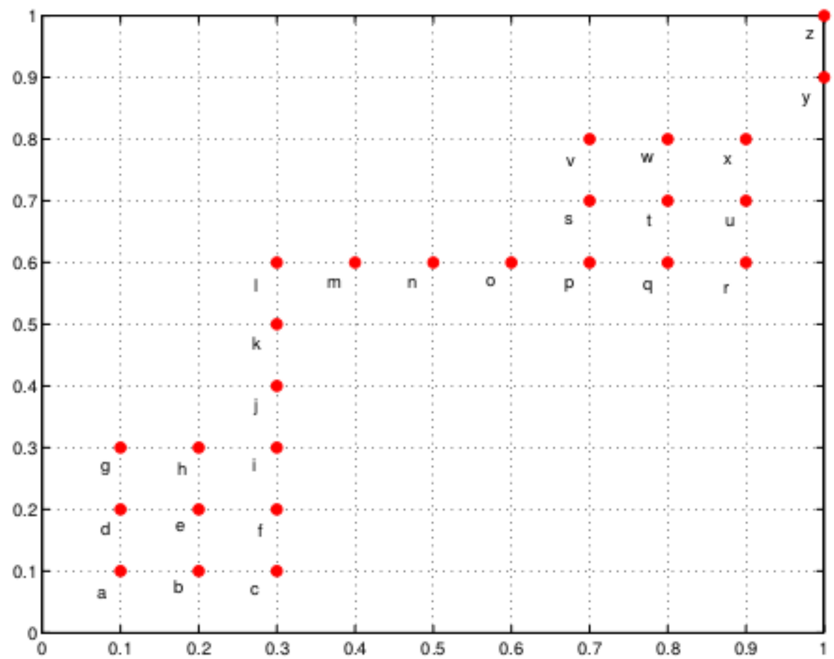

Figure 1: Data set for DBScan clustering.

(a) List all the core points in the diagram (you can use the labels of the data points in the diagram). Note: a point is considered a core point

if there are **more than MinPts** number of points (including the point itself) within a neighborhood of radius Eps.

(b) List all the border points in the diagram.

(c) List all the noise points in the diagram.

(d) Using the DBScan algorithm, how many clusters will be obtained from the data set?

6) Consider the graph data shown in Figure 2. Assume the weights for all the links are equal to 1.

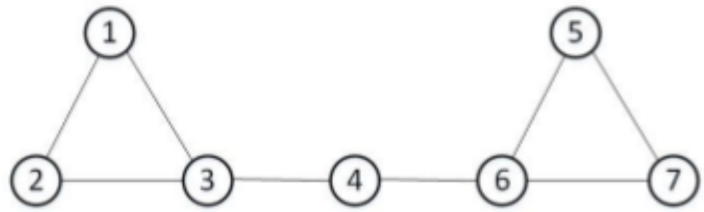

Figure 2: Graph data

(a) Compute the Laplacian matrix for the graph. Use the node indices shown in Figure 2 to order the rows and columns of the matrix.

(b) Compute the first three smallest eigenvalues of the graph Laplacian matrix.

(c) Compute the eigenvectors that correspond to the three smallest eigenvalues given in part (b).

(d) Apply k-means on the eigenvector matrix to generate 3 clusters. List the three clusters found.

(e) Calculate the normalized cut obtained for the 3 clusters found. Let V denote the set of all the nodes in a graph and $W = [w_{ij}]$ denote its adjacency matrix. Suppose V is partitioned into 3 disjoint subsets, V_1, V_2, and V_3, where $V_1 \cup V_2 \cup V_3 = V$. The normalized cut for the partitions can be computed as follows:

$$\text{Ncut}(V_1, V_2, V_3) = \sum_{i=1}^{3} \frac{\text{Cut}(V_i, V - V_i)}{d(V_i)} \tag{1}$$

where

$$\begin{aligned} d(V_i) &= \sum_{k \in V_i, j \in V} w_{ij}, \\ \text{Cut}(A, B) &= \sum_{i \in A, j \in B} w_{ij} \end{aligned} \tag{2}$$

(f) Suppose the 3 clusters found are as follows:

$$(1, 2), \quad (3, 4, 6), \quad (5, 7)$$

Compute the normalized cut of the clusters. Is the normalized cut smaller, larger, or equal to the solution found in part (d)?