PROBLEM SET III

MAT 6480 / STT 6705 - Fall Semester 2019

Your solutions must be in a single zip file titled `ps3.zip`. The zip file should include a single PDF titled `ps3.pdf` and MATLAB or Python scripts as specified. Your homework should be submitted via StadiUM before Thursday, Dec. 19, 2019 at 23:59.

Problem 1

1. Compute and compare the time complexity of k-means and PAM-based k-medoids for clustering N data points in $\mathbb{R}^n$ into k clusters based on Euclidean distances. Your complexity can be phrased in terms of N, n, k, and the number of iterations until convergence. Justify your calculations.
2. Prove the "shake & bake" dissimilarity is a distance metric.
3. Prove the attribute-wise mode is indeed the centroid w.r.t the mismatch dissimilarity in terms of minimizing SAE in each cluster.

Problem 2

1. Compute single-link and complete-link clustering, based on Euclidean distances between data points, for the data:

$$\begin{aligned} X = \{&(-31,-12),(-28,-18),(-27,-7),(-9,16),(-5,7),(-11,8),(-16,-8),\\ &(-10,-13),(-15,-19),(7,17),(8,6),(13,12),(18,6),(15,-3),(19,-5),\\ &(3,-10),(8,-17),(7,-8),(-11,-3),(-22,-13)\} \end{aligned}$$

   For each of the two clustering approaches, produce a dendrogram that clearly shows the order in which points/clusters are merged, and attach the two dendrograms to the submitted PDF.
2. To find k clusters, we can use classic k-means, bisecting k-means, and agglomerative clustering using Ward's method, while stopping the dendrogram construction when we have exactly k clusters. These three approaches are all based on the SSE as cluster quality measure. Determine which of them (if any) converges to a local minimum and which (if any) converges to a global minimum of total SSE over the clusters. Explain your answer.
3. Show that using BIRCH's clustering features one can compute the diameter

$$\mathrm{diam}(C) = \sqrt{\frac{\sum_{x,y\in C} \|x-y\|^2}{|C|(|C|-1)}}$$

   of a cluster C (where $|C|$ is the number of points in the cluster), and the inter-cluster distance

$$\mathrm{dist}(C_1, C_2) = \sqrt{\frac{\sum_{x\in C_1}\sum_{x\in C_2} \|x-y\|^2}{|C_1|\,|C_2|}}$$

   between clusters C_1 and C_2 .

Problem 3

Show that given N data points in $\mathbb{R}^n$ the k-dimensional coordinates (for $k < n$) given by PCA and by classical MDS are identical. *Hint:* you can first show this for data centered around its mean (i.e., the corresponding $N \times n$ data matrix X satisfies $\sum_{i=1}^{N} X(i,j) = 0$ for each $j = 1, \ldots, n$) and then consider data that has to be centered.

Problem 4

Notice: for this question you are not required to submit code, but it is recommended to use a script when solving it.
Consider a clustering setting where you are not getting the data points themselves as input, but just a $N \times N$ pairwise distance matrix D. To build an agglomorative hierarchical clustering scheme on this data we apply the following procedure:

Initialize:

- Set threshold $\tau \leftarrow 0$
- Create N clusters at the bottom level of the dendrogram, each containing one point

Repeat the following steps:

- Increase τ until there is at least one new pair x, y such that $D[x, y] \leq \tau$
- Create a new level in the dendrogram by merging together the clusters that are now connected by a threshold graph with edges $E = \{(x, y) \mid D_{[x,y]} \leq \tau\}$ and data points as vertices.

Until there is only one cluster left

1. Apply this approach to cluster data with the following distance matrix, which (for convenience) is also provided in the attached file `problem4_distances.csv`:

$$
\begin{bmatrix}
0 & 90.6697 & 63.8905 & 50.3289 & 66.9104 & 30.0832 & 76.0592 & 22.4722 & 56.1427 & 14.0357 \\
90.6697 & 0 & 41.7253 & 56.4624 & 24.0000 & 61.0737 & 19.7990 & 78.4092 & 37.1618 & 76.6551 \\
63.8905 & 41.7253 & 0 & 56.8595 & 30.4138 & 41.8688 & 21.9317 & 62.0081 & 15.8114 & 51.0784 \\
50.3289 & 56.4624 & 56.8595 & 0 & 35.6090 & 26.5707 & 52.3450 & 29.5296 & 41.1461 & 40.0000 \\
66.9104 & 24.0000 & 30.4138 & 35.6090 & 0 & 37.1214 & 17.2047 & 54.5894 & 18.0278 & 52.9528 \\
30.0832 & 61.0737 & 41.8688 & 26.5707 & 37.1214 & 0 & 48.2701 & 20.2485 & 29.5466 & 16.5529 \\
76.0592 & 19.7990 & 21.9317 & 52.3450 & 17.2047 & 48.2701 & 0 & 67.6757 & 20.0250 & 62.1289 \\
22.4722 & 78.4092 & 62.0081 & 29.5296 & 54.5894 & 20.2485 & 67.6757 & 0 & 49.6488 & 18.1108 \\
56.1427 & 37.1618 & 15.8114 & 41.1461 & 18.0278 & 29.5466 & 20.0250 & 49.6488 & 0 & 42.2966 \\
14.0357 & 76.6551 & 51.0784 & 40.0000 & 52.9528 & 16.5529 & 62.1289 & 18.1108 & 42.2966 & 0
\end{bmatrix}
$$

   sketch the generated dendrogram, where the vertical axis denotes the threshold τ at each level and add it to the PDF.

2. Use MDS to produce a 2D scatter plot of this data, and color it according to the top four level of the dendrogram.
   - These levels do not include the root with only one clusters, so the first level has (at least) two clusters.
   - Each of the levels should be presented in a separate figure, with the same MDS coordinates, but different colors.
   - Add these figures to the submitted PDF.

3. Does this method correspond to any of the discussed linkage models? justify your answer.

4. What if instead of adding connected components, the algorithm would only add cliques as clusters in the dendrogram? would this correspond to one of the learned linkage models? justify your answer.

Problem 5

In this problem, you will implement isomap and test your implementation on several data sets.

1. Your isomap implementation should have the following calling sequence:

$$Y = \text{isomap}(X, \text{epsilon}, d)$$

   where

   - X is the $n \times m$ data matrix corresponding to n points with m attributes,

- *epsilon* is an anonymous function of the pairwise distance matrix used to set the neighborhood parameter. (The neighborhood graph should be formed by removing edges with length greater than *epsilon* from the complete distance weighted graph).
- d is the output dimension parameter, and
- Y is the $n \times d$ isomap embedding matrix.

2. Compare your isomap implementation to classical Multidimensional Scaling (MDS) on the `swiss_roll.mat` data set via the following steps:

   - Load `swiss_roll.mat` and create a three-dimensional scatter plot of the data X colored by the vector c. Include the plot in your PDF.
   - Set the bandwidth function epsilon to be 3rd percentile of the positive entires of the pariwise distance matrix, i.e., in MATLAB,

     ```
     epsilon = @(D) prctile(D(D(:)>0),3);
     ```

     and in Python epsilon can be defined a similar way as lambda function using `numpy.percentile`.
   - Use classical MDS to create a two-dimensional embedding of X, and create a two-dimensional scatter plot of the embedding colored by the vector c. Include the plot in your PDF.
   - Use classical MDS to create a three-dimensional embedding of X, and create a three-dimensional scatter plot of the embedding colored by the vector c. Include the plot in your PDF.
   - Use your isomap implementation to create a two-dimensional embedding of X. Create a two-dimensional scatter plot of your isomap embedding colored by the vector c. Include the plot in your PDF.

3. Test your isomap implementation on the `torodial_helix` data set via the following steps:

   - Load `torodial_helix.mat` and create a three-dimensional scatter plot of the data X colored by the vector c. Include this plot in your PDF.
   - Set the bandwidth function epsilon to be 3rd percentile of the positive entires of the pairwise distance matrix.
   - Use your isomap implementation to create a two-dimensional embedding of the Torodial Helix. Create a two-dimensional scatter plot of this embedding colored by the vector c. Include this plot in your PDF.

4. Test your isomap implementation on the `swiss_roll_hole.mat` data set via the following steps:

   - Load `swiss_roll_hole.mat` and create a three-dimensional scatter plot of the data X colored by the vector c. Include this plot in your PDF.
   - Set the bandwidth function epsilon to be 3rd percentile of the positive entires of the pairwise distance matrix.
   - Use your isomap implementation to create a two-dimensional embedding of the data X. Create a two-dimensional scatter plot of this embedding colored by the vector c. Include this plot in your PDF.
   - Use your isomap implementation to create a three-dimensional embedding of the data X. Create a three-dimensional scatter plot of this embedding colored by the vector c. Include this plot in your PDF.

Implementation note

Complete the exercise without using specialized functions. For Python limit yourself to the following import statements

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.io import loadmat
from scipy.spatial.distance import cdist
```