CS 596 Machine Learning

**Homework Assignment 7**

Due: 11:59pm, May 3, 2018

The purpose of this homework assignment is to practice programming skills for unsupervised learning method, i.e. K-means.

**Dataset**

In this mini-project, we will analyze the quality of life in U.S. cities using the above methods. The dataset we use include ratings for 9 different indicators of the quality of life in 329 U.S. Cities. These are climate, housing, health, crime, transportation, education, arts, recreation and economics. For each indicator, a higher rating is better. For example, a higher rating for crime means a lower crime rate.

We provide this dataset in the file ‘cities\_life\_ratings.csv’, which includes the 9 ratings of 329 cities.

**Review of K-means**

Our purpose is to group these 329 cities into 10 clusters such that cities in the same group have similar developing indicators. To do so, we simply consider these 9 ratings as 9 features and apply the K-means algorithm.

The K-mean algorithms comprises of 6 major steps.

Step-1: load data and set environment variables. These variables include the number of clusters, K=10, the maximal iterations: MAXITER=50, etc. You don’t need to change these variables.

Step-2: Initialize centroids. You can either randomly generate these centroids or randomly select K samples. Please evaluate how your choices affect the number of iterations to converge (according to the same stopping condition as introduced later)

Step 3: Assign every sample to its closed cluster. To do so, you will need to calculate the distance between every sample to every centroid. Use Euclidean distance.

Step 4: Re-calculate the centroid of every cluster.

Step 5: Check stop conditions. You will specify the condition using one of the following observations: i) number of samples that switch their memberships since last iteration: ii) the distance of the two centroid vectors of the sample cluster over two consecutive iterations; and iii) average within-cluster distance. You are encouraged to use the mixture of these conditions or design new ones.

Step 6: Calculate the sum of squared errors (SSE) as a quantitative metric of your clustering result.

**Starter Script**

We provide a starter code in ‘main\_clustering.py’, and you will need to complete all PLACEHOLDERs to complete this script.

The script includes the following functions.

* *run\_kmeans\_clustering*(k\_value=2, showPlots=False, data=None): main function for K-mean algorithm
* *initialize\_dict*(k\_value): create a data structure for storing K clusters.
* *assign\_points\_to\_groups*(k\_value, xy\_samples, centroid\_points): assign data points to groups
* *calc\_euclidean\_dist\_vector*(vector1, vector2): calculate the Euclidean distance between two vectors.
* *check\_minimum\_changes\_met*(k\_value, current\_iteration, old\_grouped\_samples, new\_grouped\_samples): check to see if points within groups changed or not; return -1 if the number of changes is LARGE than MIN\_SAMPLE\_CHANGE\_; otherwise return the number changes.
* *get\_centroids*(dimensions, xy\_groups): calculate centroid for each group
* *get\_sum\_of\_squares*(dimensions, center, samples): calculate the sum of square errors for the current clustering result.

Please complete the placeholders in the functions: *calc\_euclidean\_dist\_vector*(), *get\_point\_mean*(), and *get\_sum\_of\_squares*().

In your report, please include the Sum of Square error while using different Ks, i.e., number of clusters. Use at least three different K values.

**How to submit**

* Submit your source codes and report through the SDSU Blackboard. The codes should be self-contained, and run without any error. Otherwise, severe penalty will be applied.
* Two re-submissions are allowed.  ***No hard copy required.***