CS373- Homework 2

Due date: March 2nd at 11:59pm EST

1 Introduction

In this programming assignment you will use **Python** to implement two clustening algorithms: K-means (`km`) and Agglomerative clustering (`ac`). You are going to apply both of them on the *Yelp* dataset a *Dummy* dataset that we will provide. Some important considerations:

1. You need to implement both algorithms (K-means and Agglomerative clustering) **from scratch** in **Python** (not R). In other words, you **can not** use any already implemented version of these two algorithms like the *scikit-learn* library. Programs that print more than what is required will be penalized.
2. You will need to submit code or written answers in PDF for each question. The required deliverables are going to be specified at the end of each question by writing **CODE** or **PDF** at their end.
3. You will use turnin to submit your deliverables: PDF and Python Script. Further instructions for submission can be found below.
4. **Your code needs to run in the server: data.cs.purdue.edu**. Please take some time to check that it works by excecuting you script from the terminal and verify the output corresponds to what is asked.

2 Submission

Your homework must be typed and must contain your name and Purdue ID. To submit your assignment, log into `data.cs.purdue.edu` (physically go to the lab or use ssh remotely) and follow these steps:

1. To ssh use the command: `ssh your-id@data.cs.purdue.edu`
2. Make a directory named `yourusername-hw2` (all letters in lower case)
3. Copy your PDF and code inside it. To do it remotely use the comand from your computer:
   `scp ./path/to/your-file.pdf your-id@data.cs.purdue.edu:./remote/path/from-home-dir/`
4. Go to the directory containing `yourusername-hw2` (e.g., if the files are in /homes/dan/dan-hw2, go to /homes/dan), and execute the following command:

   ```
   turnin -c s373 -p hw2 yourusername-hw2
   ```

   (e.g. Dan would use: `turnin -c s373 -p hw2 dan-hw2` to submit his work)

   Note that `s373` is the course name for turnin. **It is not a typo**.
5. To overwrite an old submission, simply execute this command again.
6. To verify the contents of your submission, execute the following command:

   ```
   turnin -v -c s373 -p hw2
   ```

3 Code details

3.1 Algorithm Details

You will implement the K-means and Agglemerative Clustering algorithms and apply it to the subset of 4 features specified in the *Dataset Details* section.

Features: Consider 4 continuous attributes in `yelp.csv` for $\mathbf{x} \in \mathbf{X}$.

Score function: Use within-cluster sum of squared errors (where r_k is the centroid of cluster C_k):

$$wc(C) = \sum_{k=1}^{K} \sum_{x(i) \in C_k} (x(i) - r_k)^2$$

Agglomerative clustering: iteratively pick two clusters where their distance is minimal and fuse them. The minumun distance will be decided by the average link cluster distance:

$$cluster\_distance(C_i, C_j) = \frac{1}{|C_i| * |C_j|} \sum_{x_i \in C_i} \sum_{x_j \in C_j} sample\_distance(x_i, x_j)$$

3.2 Dataset Details

Download the file `yelp.csv` from the course page. The document has $24,814$ rows and 19 attributes. You are only going to use 4: `latitude, longitude, reviewCount, checkins`. Also, download the file `dummy.csv`, this dataset will only have the 4 attributes needed.

In order to read the csv and obtain a numpy matrix X with the necessary attributes you can use the following code:

```
import pandas as pd
data = pd.read_csv(file_path, sep=',', quotechar='"', header=0)
data = data[['latitude', 'longitude', 'reviewCount', 'checkins']]
X = data.as_matrix()
```

3.3 Code specification

Your python script should take three (3) arguments as input.

1. *datasetPath*: corresponds to path to the dataset file, in the same format as `yelp.csv`, that your algorithm will use.
2. *K*: the value of k to use when clustering.
3. *model*: model that you want to use. In this case, we will use `km` for K-means and `ac` for Agglomerative clustering.

Your code should read the dataset from the *datasetPath*, extract the required features (`latitude, longitude, reviewCount, checkins`), cluster the dataset using the specified *model* and value of k, and output the within-cluster sum of

squared errors together with the cluster centroids. We will define the centroids in the `ac` case as the average between all points that belong to the same cluster. Each cluster centroid will have 4 dimensions, one for each feature.

Name your file clustering.py. The input and output should look like this:

```
$ python clustering.py ./some/path/file_name.csv K ac
WC-SSE=212.22
Centroid1=[49.00895,8.39655,12,3]
...
CentroidK=[33.33548605,-111.7714182,9,97]
```

Note: This is how we will run your code to grade correctness. You can (and should) use wrapper methods to run your own analysis for the rest of the questions.

4 Assignment

1. Create a python script that can apply K-means and Agglomerative Clustering algorithms in python as specified in the *Code Specification* section. **This is the only piece of code that you need to submit (CODE)**. We will run several tests on your code to assess it's correctness. (30 pts)

2. K-means analysis **You do not need to submit code for this part. Just include the required plots and explanations in the PDF** (50 pts).

   (a) Cluster the full Yelp data with values of $K = [2, 4, 8, 16, 32, 64]$ using a random set of examples as the initial centroids. Then, plot the within-cluster sum of squares as a function of K. Which value of k would you choose? Why? (Plot + answer in the PDF)

   (b) For $K = 4$ build a scatter 2D plot using the samples (one color per cluster) in two ways: (1) `latitude` vs. `longitude` and (2) `reviewCount`, `checkins`. What are the dimensions that are driving (most taken into account by) the clustering model? Why?. (2 Plots + answer in the PDF)

   (c) Do a log transform of `reviewCount`, `checkins`, then repeat the above analysis (a) and (b). Discuss any differences in the results. (1 Plot (a) + 2 Plots (b) + answer in the PDF)

   (d) Repeat the analysis (a) and (b) but first use the function `sklearn.preprocessing.scale()` command in the original dataset (not the one from (c)) to transform the data so that each attribute has $mean = 0$ and $stdev = 1$. Discuss the impact on the empirical results. (1 Plot (a) + 2 Plots (b) + answer in the PDF)

3. K-means vs. Agglomerative clustering. **You do not need to submit code for this part. Just include the required plots and explanations in the PDF** (20 pts).

   (a) Run K-means (using a random set of examples as the initial centroids) and Agglomerative Clustering on the *dummy.csv* dataset with $k = 3$. Make a scatter plot of `latitude` vs. `longitude` after each run where each sample is colored according to its cluster color. Subjectively (looking at the plot), which algorithm performs better? Explain. (2 Plots + answer in the PDF)

(b) Does K-means always yield the same result if it is applied over and over? What about Agglomerative Clustering?

(c) If K-means and Agglomerative clustering are applied on the $Yelp$ dataset, which one is going to take more time? Why? (answer in the PDF)