INF553 Foundations and Applications of Data Mining

Spring 2019

Assignment 5 Clustering

Deadline: Apr. 10th 11:59 PM PST

1. Overview of the Assignment

In Assignment 5, you will implement the Bradley-Fayyad-Reina (BFR) algorithm. The goal is to let you be familiar with the process clustering in general and various distance measurements. The datasets you are going to use is a synthetic dataset.

2. Assignment Requirements

2.1 Programming Language and Library Requirements

a. You must use Python to implement the algorithm. You can only use the following external Python libraries: numpy and sklearn. **You cannot use packages for calculating Mahalanobis Distance.**

2.2 Programming Environment

We will use **Python 3.6** to test your code. There will be a 20% penalty if we cannot run your code due to the library version inconsistency.

2.3 Write your own code

Do not share your code with other students!!

We will combine all the code we can find from the Web (e.g., GitHub) as well as other students' code from this and other (previous) sections for plagiarism detection. We will report all the detected plagiarism.

2.4 What you need to turn in

Your submission must be a **zip file** with the naming convention: **firstname_lastname_hw5.zip** (all lowercase, e.g., yijun_lin_hw5.zip). You should pack the following required (and optional) files in a folder named **firstname_lastname_hw5** (all lowercase, e.g., yijun_lin_hw5) in the zip file (Figure 1):

a. [REQUIRED] One Python scripts containing the main function, named:

firstname_lastname_bfr.py

b. [OPTIONAL] You can include other scripts to support your programs (e.g., callable functions), but you need to make sure after unzipping, they are all in the same folder "firstname_lastname_hw5".

c. You don't need to include any result. **We will grade your code using our testing dataset.** Our testing dataset will be in the same format as the given dataset.

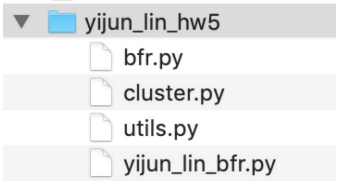

Figure 1: An example of the folder structure after your submission file is unzipped.

3. Dataset

Since the BFR algorithm has a strong assumption that the clusters are **normally distributed with independent dimensions**, we generated synthetic datasets by initializing some random centroids and creating some data points with the centroids and some standard deviations to form the clusters. We also add some other data points as the outliers in the dataset to evaluation the algorithm and **their cluster is represented by -1**. Figure 2 shows an example of a part of the dataset. The first column is the data point index (the first column). The second column is the cluster that the data point belongs to. The rest columns represent the features/dimensions of the data point.

```
2708,0,2.438665943891622,1.027818493861439
2709,4,127.31966592930607,148.31411942629745
2710,4,133.61619393051998,157.97961951108087
2711,3,-91.58594502448611,-119.9550977954772
2712,4,135.69634514856796,177.55732400164032
2713,0,12.108343663244053,1.4969275135468427
2714,-1,467.7774695478297,670.5177459229164
2715,3,-104.62798147039325,-128.2723232272566
2716,1,-52.994607584888456,64.12806862225437
2717,4,144.02609145460397,177.36349742076746
2718,4,151.4896164272161,170.51611070365897
```

Figure 2: An example of the dataset

a. hw5_clustering.txt: the synthetic clustering dataset with 10 dimensions. We created this dataset using 10 centroids. The dataset is in the Google Drive:
https://drive.google.com/open?id=11anZyiXBfhyICo2mkn6PFnegXgiDiUpF

b. We generate the testing dataset using similar method. The testing data will have similar number of data points and number of clusters as hw5_clustering.txt. **Notice that the number of the dimensions could be different from the hw5_clustering.txt.** We do not share the testing dataset.

4. Task

You will implement the Bradley-Fayyad-Reina (BFR) algorithm using **hw5_clustering.txt**.

In BFR, there are three sets of points that you need to keep track of:

Discard set (DS), Compression set (CS), Retained set (RS)

For each cluster in the DS and CS, the cluster is summarized by:

N: The number of points

SUM: the sum of the coordinates of the points

SUMSQ: the sum of squares of coordinates

The conceptual steps of the BFR algorithm:

Please refer to the slide.

Implementation details of the BFR algorithm: (this implementation is just for your reference)

Step 1. Load 20% of the data **randomly**.

Step 2. Run K-Means (e.g., from sklearn) with a large K (e.g., 10 times of the given cluster numbers) on the data in memory using the Euclidean distance as the similarity measurement.

Only for generating RS

many groups in this clusters

Step 3. In the K-Means result from Step 2, move all the clusters with only one point to RS (outliers).

Only for generating DS

Step 4. Run K-Means again to cluster the rest of the data point with K = the number of input clusters.

Step 5. Use the K-Means result from Step 4 to generate the DS clusters (i.e., discard their points and generate statistics).

The initialization of DS has finished, so far, you have K numbers of DS clusters (from Step 5) and some numbers of RS (from Step 3).

Step 6. Run K-Means on the points in the RS with a large K to generate CS (clusters with more than one points) and RS (clusters with only one point).

Step 7. Load another 20% of the data **randomly**.

Step 8. For the new points, compare them to each of the DS using the **Mahalanobis Distance** and assign them to the nearest DS clusters if the distance is < $2\sqrt{d}$ (d is the number of dimensions).

You cannot use sklearn package for calculating the Mahalanobis Distance. Hint: you will need to compute variance using $\frac{SUMSQ}{n} - \frac{SUM^2}{n}$ between the cluster and the point.

Step 9. For the new points that are not assigned to DS clusters, using the **Mahalanobis Distance** and assign the points to the nearest CS clusters if the distance is < $2\sqrt{d}$

Step 10. For the new points that are not assigned to a DS cluster or a CS cluster, assign them to RS.

Step 11. Run K-Means on the RS with a large K to generate CS (clusters with more than one points) and RS (clusters with only one point).

Step 12. Merge CS clusters that have a Mahalanobis Distance < $2\sqrt{d}$.

Repeat Steps 7 – 12: you can refer to the following piece of code.

```
n_sample = len(data)  # the number of data points we have
percentage = 0.2  # percentage of the data points to be load in the memory
init_data = data[:int(n_sample * percentage)]  # generate the data points for initialization

bfr.init(init_data)  # initialize the DS, CS, and RS

start = int(n_sample * percentage)
end = start + int(n_sample * percentage)

while start < n_sample:  # while there are data continuously coming in
    bfr.bfr_main(data[start:end])  # do the BFR algorithm
    start = end
    end = start + int(n_sample * percentage)
    if last_round:
        bfr.finish()  # assign CS to DS
```

If this is the last run (after the last chunk of data), merge CS clusters with DS clusters that have a Mahalanobis Distance < $2\sqrt{d}$.

At each run, including the initialization step, you need to count and output the number of the discard points, the number of the clusters in the CS, the number of the compression points, and the number of the points in the retained set.

Input format: (we will use the following command to execute your code)

```
Python: $ python3 firstname_lastname_bfr.py <input_file_name> <n_cluster> <output_file_name>
```

Param: input_file_name: the name of the input file (e.g., hw5_clustering.txt), including the file path.
Param: n_cluster: the number of the clusters.
Param: output_file_name: the name of the output txt file, including the file path.

Output format:

IMPORTANT: Please strictly follow the output format since your code will be graded automatically. We will not regrade on formatting issues.

The output file is a text file, containing the following information (see Figure 3):

a. The intermediate results (the line is named as "The intermediate results"). Then **each line should be started with "Round {i}:" and i is the count for the round (including the initialization, i.e., initialization would be "Round 1:".** You need to output the numbers **in the order of** "the number of the discard points", "the number of the clusters in the compression set", "the number of the compression points", and "the number of the points in the retained set".

b. The clustering results (the line is named as "The clustering results"), including the data points index and their clustering results after the BFR algorithm. The clustering results should be in **[0, the number of clusters)**. The cluster of outliers should be represented as -1.

```
The intermediate results:
Round 1: 2898,10,20,82
Round 2: 5236,6,147,17
Round 3: 7566,4,234,0
Round 4: 9694,3,306,0
Round 5: 9998,1,2,0
...

The clustering results:
0,1
1,1
2,0
3,0
4,-1
5,1
6,0
7,1
...
```

Figure 3: the output example for text file

Grading:

a. We will compare the percentage of discard points after the last round to our threshold.

b. We will compare the centroids of the clusters you find to those in the ground truth and compute the distance between the two centroids.

c. We will compute the accuracy of your clustering results to the ground truth. We will check the percentage of data points that you issue to the correct cluster **(except the outliers)** using the following formula:

$$Accuracy = \frac{the\ total\ number\ of\ points\ in\ the\ correct\ clusters}{the\ total\ number\ of\ points\ in\ the\ ground\ truth}$$

d. No point if your runtime is more than or equal to 500 seconds.

We provide the threshold for the hw5_clustering.txt here for your reference.

Dataset	hw5_ clustering.txt
Percentage of discard points after last round	95.0%
Accuracy	95.0%

5. Grading Criteria

(% penalty = % penalty of possible points you get)

1. You can use your free 5-day extension separately or together.
2. If we cannot run your programs with the command we specified, there will be an 80% penalty.
3. If your program cannot run with the required Python versions, there will be a 20% penalty.
4. We can regrade on your assignments within seven days once the scores are released. No argue after one week. The regrading will have a 20% penalty if our grading is correct.
5. There will be a 20% penalty for late submissions within a week and no point after a week.