Ira A. Fulton Schools of Engineering
Arizona State University

CSE 572: Data Mining

Project 3: Cluster Validation

Purpose

In this project you will apply the cluster validation technique to data extracted from a provided data set.

Objectives

Students will be able to:

- Develop code that performs clustering.
- Test and analyze the results of the clustering code.
- Assess the accuracy of the clustering using SSE and supervised cluster validity metrics.

Technology Requirements

Python 3.6 to 3.8 (do not use 3.9).

scikit-learn==0.21.2

pandas==0.25.1

Python pickle

Project Description

For this project you will write a program, using Python, that takes a dataset and performs clustering. Using the provided training data set you will perform cluster validation to determine the amount of carbohydrates in each meal.

Directions

There are two main parts to the process:

1. Extract features from Meal data
2. Cluster Meal data based on the amount of carbohydrates in each meal

Data:

Use the Project 1 data files

CGMData.csv

InsulinData.csv

Extracting Ground Truth:

> Derive the max and min value of meal intake amount from the Y column of the Insulin data. Discretize the meal amount in bins of size 20. Consider each row in the meal data matrix that you generated in Project 2. Put them in the respective bins according to their meal amount label.
>
> In total you should have n = (max-min/20) bins.

Performing clustering:

Use the features in your Project 2 to cluster the meal data into n clusters. Use DBSCAN and KMeans.

Report your accuracy of clustering based on SSE, entropy and purity metrics.

Expected Output:

A Result.csv file which contains a 1 X 6 vector. The vector should have the following format

SSE for Kmeans	SSE for DBSCAN	Entropy for KMeans	Entropy for DBSCAN	Purity for KMeans	Purity for DBSCAN

The Result.csv file should not have any headers, just the six values in six columns.

Submission Directions for Project Deliverables

A zip file which has all your code. In the code you should have one main python file which the autograder can run and generate Result.csv file according to specifications. Assume that CGMData.csv and InsulinData.csv are already in the execution folder. You can have as many auxiliary python files as you want but the autograder will only run the main.py and it should generate the Result.csv.

Evaluation

50 points for developing a code in Python that takes the dataset and performs clustering

20 points for developing a code in Python that implements a function to compute SSE, entropy and purity metrics. These two can be written in the same file.

30 points will be evaluated on the supervised cluster validation results obtained by your code. This will be compared against class average to determine the final score.