Dr. Eick

COSC 3337 *“Data Science I”*

ProblemSet3 Fall 2022

Centering on Clustering

3rd Draft

Individual Task

Deadlines: Task 4 is due Tuesday, December 6, 11:59p

Available Points: 55

Last updated: November 16, noon

Task 4:

**Learning Objectives**:

1. Learn to use popular clustering algorithms, namely K-means, and DBSCAN
2. Learn how to summarize and interpret clustering results
3. Learn to write analysis and evaluation functions which operate on the top of clustering algorithms and clustering results
4. Learning how to interpret unsupervised data analysis results

**Datasets**: In the project we will use the <http://www2.cs.uh.edu/~ceick/UDM/DataSets/complex9_gn8.txt> and the 9D Statlog (Shuttle) Data Set. We also use the original Complex9 dataset <http://www2.cs.uh.edu/~ceick/UDM/DataSets/Complex9.txt> for some postanalysis; Complex9\_gn8 has been created by exposing the orignal Complex9 dataset to Gaussian noise, adding 8% new examples.). We use a variation of the Shuttle dataset for Task4 which can be found at: <http://www2.cs.uh.edu/~ceick/UDM/DataSets/Shuttle22.csv>

The original Shuttle dataset can be found at: <https://archive.ics.uci.edu/ml/datasets/Statlog+%28Shuttle%29> ; The shuttle dataset contains 9 attributes all of which are numerical, with the first one being time. This data set was generated originally to extract comprehensible rules for determining the conditions under which an autolanding would be preferable to manual control of a spacecraft. The purpose is to decide what type of control of the vessel should be employed, which is described by the Class variable in the last column. The explanation for each Class value is as follows:

1 Rad Flow; 2 Fpv Close; 3 Fpv Open; 4 High; 5 Bypass; 6 Bpv Close; 7 Bpv Open;

Example data from the Shuttle dataset:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***1*** | ***2*** | ***3*** | ***4*** | ***5*** | ***6*** | ***7*** | ***8*** | ***9*** | ***Class*** |
| 50 | 21 | 77 | 0 | 28 | 0 | 27 | 48 | 22 | 2 |
| 55 | 0 | 92 | 0 | 0 | 26 | 36 | 92 | 56 | 4 |
| 53 | 0 | 82 | 0 | 52 | -5 | 29 | 30 | 2 | 1 |

We will z-score the dataset before clustering the dataset; we call this dataset *ZSHUT[[1]](#footnote-1)* in the following. The last attribute of each dataset denotes a class variable which should be ignored when clustering the data sets—however, the class variable will be used in the post analysis of the clusters generated by running K-means and DBSCAN.

**Task 4 Subtasks:**

1. Write an function[[2]](#footnote-2) purity(a,b,outliers=FALSE) that computes the purity of a clustering result based on an apriori given set of class lables, where *a* gives the assignment of objects in O to clusters, and *b* is the “ground truth”. Purity is defined as follows: Let

O be a dataset

X={C1,…,Ck} be a clustering of O with Ci ⊆O (for i=1,…,k), C1∪…∪Ck ⊆O and Ci∩Cj=∅ (for i≠ j)

PUR(X)= (number\_of\_majority\_class\_examples(X)/(total\_number\_examples\_in\_clusters(X))

If the used clustering algorithm supports outliers, outliers should be ignored in purity computations; if you use R-clustering algorithms, you can assume that cluster 0 contains all the outliers, and clusters 1,2,…,k represent “true” clusters. If the parameter outliers is set to FALSE, the function just returns a floting point number of the observed purity, if parameter outliers is set to T the function returns a vector: (<purity>,<percentage\_of\_outliers); e.g. if the function returns (0.98, 0.2) this would indicate that the purity is 98%, but 20% of the objects in dataset O have been classified as outliers. **2 points**

1. Run[[3]](#footnote-3) K-means for k=9 and k=13 for the Complex9\_gn8 dataset. Visualize the obtained two clusterings. Also compute their purity using the function you developed in Task a. Assess if the K-means clustering was able to “rediscover” the natural clusters of the original Complex 9 dataset, captured by the 3rd attribute of the dataset \*\* **4 points**
2. Run2 K-means for k=3 for the *ZSHUT* dataset. Compute the purity of the obtained clustering result; also create box plots for all 9 attributes of the obtained 3 clusters and report their centroids. We also recommend to create boxplots for the whole dataset to have a reference point to interpret the boxplots of the three clusters. Finally, summarize based on the obtained boxplots and centroids what kind of objects each of the 3 clusters contains. **15 points**
3. Try to obtain a DBSCAN clustering for the *ZSHUT* dataset, having between 2 and 15 clusters with less than 20% outliers. Report its purity! Don’t worry if the reported clustering has a low purity.**4 points**
4. Develop a search procedure that looks for the “best” clustering by exploring different settings for the (MinPoints, epsilon) parameters of DBSCAN for the complex9-gn8 dataset. The procedure maximizes purity of the obtained clustering, subject to the following constraints:
   * 1. There should be between 2 and 15 clusters
     2. The number of outliers should be 10% or less.

The procedure returns the “best” DBSCAN clustering found and the accomplished purity as its result[[4]](#footnote-4); please limit the number of tested (MinPoints, epsilon)-pairs to 3000 in your implementation! You are allowed to run the search procedure 3 times from different starting positions. Explain how your automated parameter selection method works and demonstrate your automated procedure using an example!

Apply the procedure you developed to the Complex9\_gn8 dataset and report the best clustering you found. Are you happy with the obtained solution? Finally, run DBSCAN for the best parameter setting you found for the orignal Complex9 dataset and summarize the results. Finally, assess to which extend DBSCAN was able to identify the clusters in the original dataset and if it was able to idenfity noise points that were added to Complex9 when generating Complex9\_gn8.

If you did not succeed in writing the function that seeks of the optimal DBSCAN clustering, you can manually seek for the best clustering for the Complex9\_gn8 dataset and report it; also report how you searched manually for it; you will lose 4 points for not having a search procedure. Will will also evaluate the quality of the best clustering you found; that is, better clusterings will receive more points. **15 points** total

Extra credit: Apply your search procedure also to the Shuttle Dataset and report the clusters of the best result and what purity you accomplished. **5 points**

**Deliverables for Task 4:**

1. A Report[[5]](#footnote-5) which contains all deliverables for the 5 subtasks of Task 4.
2. The report should have an Appendix which describes how to run the procedure that you developed for Task e, if you developed such a procedure.
3. Another Appendix which contains the software/code you developed as part of Task 4.

1. ZSHUT has 9 attributes and a class variable. [↑](#footnote-ref-1)
2. This function could be an R-function, a Python function or any other function. You might find some implementation of this function online; it is okay to use those implementations, as long as you acknowledge in your report what you use, and not all software you find on the internet is running properly. [↑](#footnote-ref-2)
3. Run k-means 20 times and use the clustering with the lowest SSE! [↑](#footnote-ref-3)
4. It should report the number of clusters obtained and the percentage of outliers as well. [↑](#footnote-ref-4)
5. Single-spaced; please use an 11-point or 12-point font! [↑](#footnote-ref-5)