CA Assignment 2

Clustering Algorithms

Assignment Number	2 of (2)
Weighting	15%
Assignment Circulated	15.03.2024
Deadline	30.04.2024
Submission Mode	Electronic Via Canvas
Purpose of assessment	The purpose of this assignment is to demonstrate: (1) the understanding of the KMeans (2) the understanding of KMeans++(3) the understanding of evaluation metrics for clustering.
Learning outcome assessed	A critical awareness of current problems and research issues in data mining. (3) The ability to consistently apply knowledge concerning current data mining research issues in an original manner and produce work which is at the forefront of current developments in the sub-discipline of data mining.

1. Implement k-means clustering algorithm and cluster the dataset provided using it. Vary the value of k from 1 to 9 and compute the Silhouette coefficient for each set of clusters. Plot k in the horizontal axis and the Silhouette coefficient in the vertical axis in the same plot. (20)
2. Generate synthetic data of same size (i.e. same number of data points) as the dataset provided and use this data to cluster K Means. Plot k in the horizontal axis and the Silhouette coefficient in the vertical axis in the same plot. (10)
3. Implement k-means++ clustering algorithm and cluster the dataset provided using it. Vary the value of k from 1 to 9 and compute the Silhouette coefficient for each set of clusters. Plot k in the horizontal axis and the Silhouette coefficient in the vertical axis in the same plot. (20)
4. Implement the Bisecting k-Means algorithm to compute a hierarchy of clusterings that refines the initial single cluster to 9 clusters. For each s from 1 to 9, extract from the hierarchy of clusterings the clustering with s clusters and compute the Silhouette coefficient for this clustering. Plot s in the horizontal axis and the Silhouette coefficient in the vertical axis in the same plot. (20)
5. Compute the confusion matrix, macro-averaged Precision, Recall, and F-score for the clustering shown in Figure 1. (20)
6. For the same clusters as in Excercise 1, compute B-CUBED Precision, Recall, and F-score. (10)

Important Notes

1. No credit will be given for implementing any other type of clustering algorithms or using an existing library for clustering instead of implementing it by yourself. However, you are allowed to use
   - numpy library (any function)
   - random module;
   - matplotlib for plotting; and
   - pandas.read_csv, csv.reader, or similar modules only for reading data from the files.

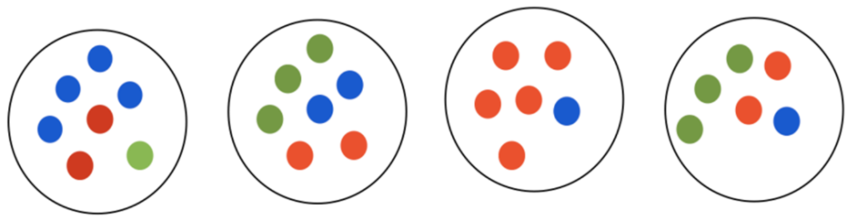


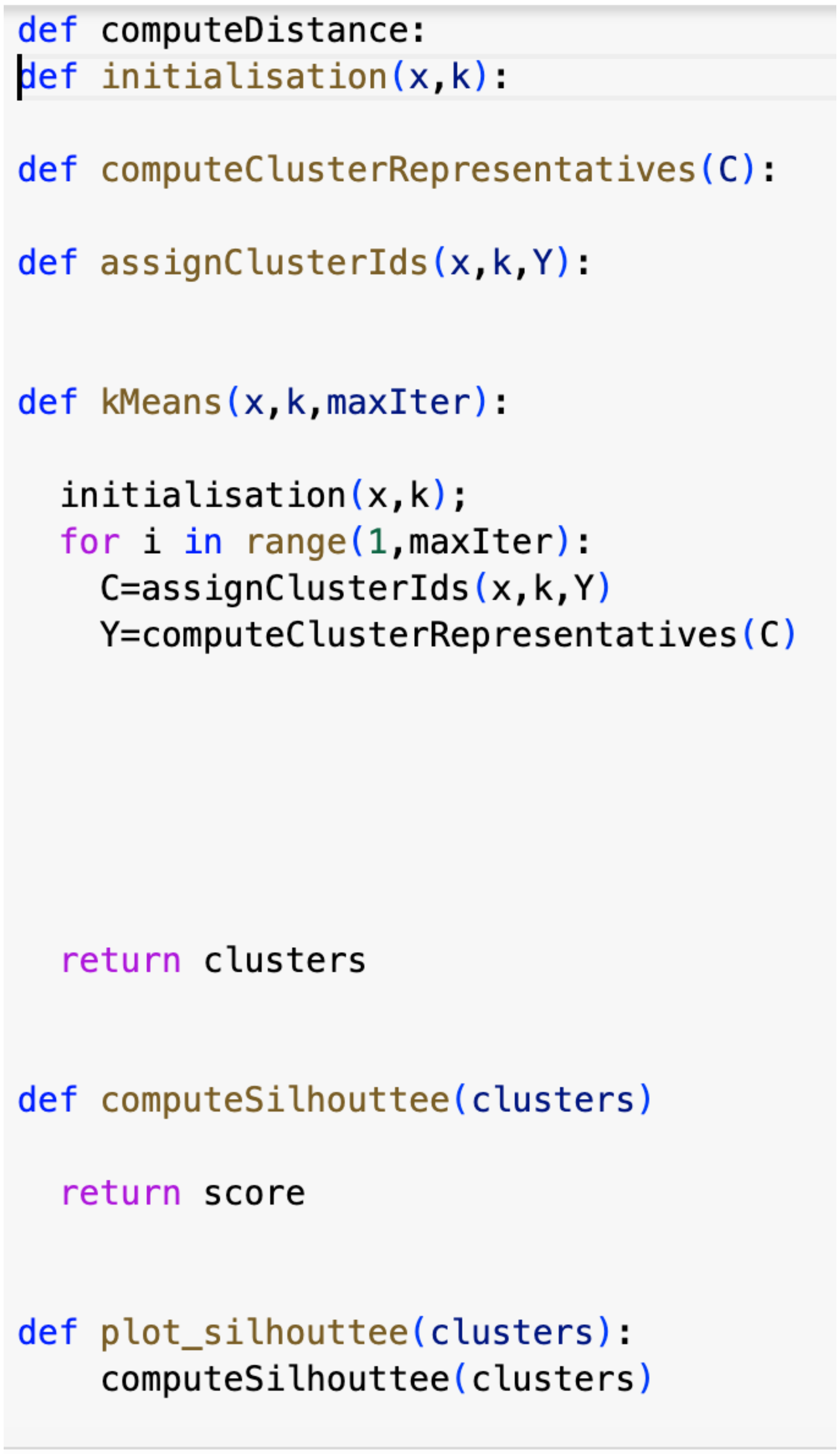
Figure 1: Outcome of a clustring algorithm

Figure 1: Caption

However, it is not a requirement of the assignment to use any of those modules.

2. Your program
   - should run and produce all results for Questions 1, 2, 3 and 4 in one click without requiring any changes to the code;
   - should output only the required data in a clearly structured way; it should NOT output any intermediate steps;
   - should assume that the input file is named ‘dataset’ and is located in the same folder as the program; in particular, it should NOT use absolute paths.
3. Programs that do not run will result in a mark of zero!
4. Your code should be as clear as possible and should contain only the functionality needed to answer the questions. Provide as much comments as needed to make sure that the logic of the code is clear enough to a marker. Marks may be deducted if the code is obscure, implements unnecessary functionality, or is overly complicated.
5. If you use module random to make some random actions, use a fixed seed value so that your program always produces the same output.
6. The answers of Questions 1 to 4 will be in the form of .py files and the answer for Question 5 and 6 should be in a PDF format.
7. The python code of the implementation of the algorithms should be included in the .py file, and not in the report.
8. You are allowed to (re)use any part of the function that computes Silhouette coefficient from the solution to the lab tasks for Week 7.
9. For Question 1, the name of the coding file should be KMeans.py.
10. For Question 2 the name of the coding file should be KMeansSynthetic.py.
11. For Question 3 the name of the coding file should be KMeansplusplus.py.
12. For Question 4 the name of the coding file should be BisectingKMeans.py.
13. For Questions 1 to 4, markers will run python filename.py. This should be able to generate the corresponding plot in the current directory.
14. There will be a load_dataset function for Question 1,3 and 4. This function will be used to process the dataset provided.

15. For questions 1 to 4 there should be following functions defined in your code.
    - a function called plot_silhouttee to write the code for plot number of clusters vs. silhouttee coefficient values.
    - a function called ComputeDistance to computing the distance between two points.
    - a function called initialSelection which will choose initial cluster representatives or clusters.
    - a function called clustername(x,k) where x is the data and k is the value of maxIter.
16. For question 1 to 3, Following functions should be there.
    - a function named assignClusterIds that will assign cluster ids to each data point.
    - a function named computeClusterRepresentatives which will compute the cluster representations.
17. For Question 4, computeSumfSquare function to compute the sum of squared distances within a cluster.
18. You can use the KMeans function implemented for question 1 in Question 2 and 4.
19. Each function should have a comment. Each comment should describe input, output and what the function does.
20. Edge case conditions should be handled (e.g. File not given, File corrupted, only 1 datapoint in the file).
21. Your submission should be your own work. Do not copy or share! Make sure that you clearly understand the severity of penalties for academic misconduct (`https://www.liverpool.ac.uk/media/livacuk/tqsd/code-of-practice-on-assessment/appendix_L_cop_assess.pdf`).
22. Plotting should generate the plot in my current folder
23. You're free to include as many functions in your program as you need. Nevertheless, you should have at least the functions specified earlier.
24. A sample program structure for KMeans is given below just for the illustration purpose. You can follow different program structure with same functions.



```
def computeDistance:
def initialisation(x,k):

def computeClusterRepresentatives(C):

def assignClusterIds(x,k,Y):


def kMeans(x,k,maxIter):

  initialisation(x,k);
  for i in range(1,maxIter):
    C=assignClusterIds(x,k,Y)
    Y=computeClusterRepresentatives(C)







  return clusters


def computeSilhouttee(clusters)

  return score


def plot_silhouttee(clusters):
    computeSilhouttee(clusters)
```

Figure 2: Sample Code Structure