Middle East Technical University Department of Computer Engineering

CENG 499

Introduction to Machine Learning

Spring 2020-2021

Homework 2 - KNN, K-means, HAC

version 2

Due date: May 18, 2020, Tuesday, 23:59

1 Introduction

In this assignment you have three short tasks. They are on k-nearest neighbor (KNN) algorithm, k-means clustering, and hierarchical agglomerative clustering (HAC). All tasks are independent from each other and have different data for you to work on. All related files can be downloaded from http://user.ceng.metu.edu.tr/~artun/ceng499/ceng499_hw2.zip. Using the lecture materials while doing the homework is strongly recommended. First part is covered in kNN-intro_to_ML.pdf, the second part is covered in k-means.pdf, and the third part is covered in hac.pdf [1].

At each part of this homework, you are going to fill the empty functions given to you. To understand how we expect them to work, you can read their descriptions or inspect the mini testers given with them. After filling those functions, you are expected to use them to complete the tasks. You can write your additional code inside mini testers or in a different file.

2 Part 1: K-Nearest Neighbor

2.1 Dataset

The dataset is created by modifying and introducing some noise to the Iris Dataset [2], so you need to use the version we provided. It resides in the folder **hw2_data/knn**. The dataset is labeled (since the task is supervised learning) and have separate train and test sets. It is in the form of saved numpy array and you can load it by:

```
train_data = np.load('hw2_data/knn/train_data.npy')
train_labels = np.load('hw2_data/knn/train_labels.npy')
test_data = np.load('hw2_data/knn/test_data.npy')
test_labels = np.load('hw2_data/knn/test_labels.npy')
```

2.2 Task

You are expected to fill the empty functions and use k-fold cross-validation on training data to find a suitable k for the KNN algorithm and test your KNN algorithm using that k on test dataset and report your accuracy. You should do:

- Fill the empty functions. You can test your code with the provided mini tester. However, be careful that passing all the tests in the mini tester may not mean your code is correct.
- For k_{KNN}=1, 2, 3, 4, ..., 199 , apply k-fold cross-validation using training data **only** and plot the avarage accuracies from the cross-validation. The plot should have k_{KNN} values for its x-axis and average accuracies for each of them for its y-axis. For this homework you can select k_{CV}=10 for the k-fold cross-validation and you don't need to experiment on the k of cross-validation.
- Comment on why the average accuracy drops in the plot with the very large values of k.
- Calculate accuracy on the test set using the train set and the k you found the best in the cross-validation and report it.

3 Part 2: K-means Clustering

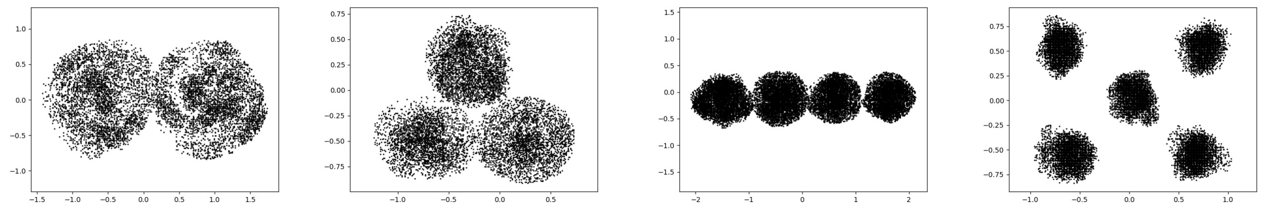

(a) clustering1 (b) clustering2 (c) clustering3 (d) clustering4

Figure 1: Illustration of the clustering data to be used in k-means task. Since there are no labels, the data is all black.

3.1 Dataset

The dataset for this task is synthetically created. It resides in the folder **hw2_data/kmeans**. You can see their visualization in Fig. 1. You don't have the labels since the task is clustering. The data is in the form of saved numpy array and you can load it by:

```
clustering1 = np.load('hw2_data/kmeans/clustering1.npy')
clustering2 = np.load('hw2_data/kmeans/clustering2.npy')
clustering3 = np.load('hw2_data/kmeans/clustering3.npy')
clustering4 = np.load('hw2_data/kmeans/clustering4.npy')
```

3.2 Task

You are expected to fill the empty functions and use elbow method to find a suitable k for the k-means algorithm and apply k-means with the selected k to each clustering data. You should do:

- Fill the empty functions. You can test your code with the provided mini tester. However, be careful that passing all the tests in the mini tester may not mean your code is correct.

  ☞ In real life, you may want to set maximum number iterations for the loop of k-means. However, if you do that, make sure that the value of maximum number of iterations is large enough so that you don't stop the algorithm immaturely. In this homework, I advise you to write a "while True:" loop and see the algorithm until its convergence.

- Write a function to initialize cluster centers. Notice that this function is not given as a function you need to fill. So you need to write a new function. You can just uniformly sample within the boundaries created by the minimum and maximum values of your data.

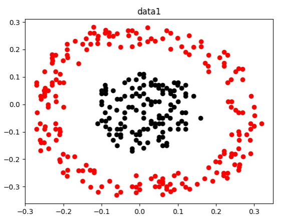
  ☞ There are many ways to intialize the clusters. You can use different strategies; however, be careful that you implemented them correctly. People tend to make mistakes during initialization.

- The result of the k-means algorithm depends on the initial clusters. Therefore, you are expected to do more than one initialization for the same setting and take the best result among them to have more consistent results. For each k=1, 2, 3, ..., 10, restart k-means many times (for example 10 times) and **take the minimum** of the final values of the objective function. Plot the values where x axis denotes the values of k and y axis denotes the best final values of the objective function on that k value.

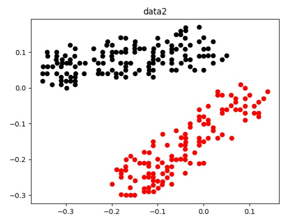
- Use elbow method to decide the suitable k's for each clustering data mentioned above. The k values may be different for different clustering data. Deciding where the elbow is may change from person to person but you can use the visualizations in Fig. 1 to double check your results.

- Apply k-means to each clustering data with their obtained k values and plot the results. The plots should be the colored versions of the ones shown in Fig. 1 with the addition of cluster centers. Draw the cluster centers with different shape than the normal data so that it can easily be understood. All the data provided to you are two-dimensional; therefore, it can be plotted easily.

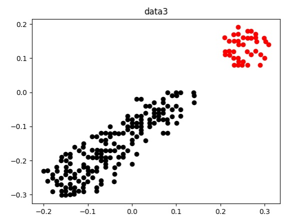
4 Part 3: Hiearchical Agglomerative Clustering



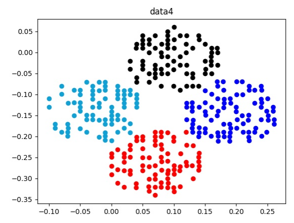
(a) data1



(b) data2



(c) data3



(d) data4

Figure 2: Illustration of the data to be used in HAC task. The data is normally not labeled; however, to remove ambiguity, the plots are colorized to depict what your expectations from the data should be.

4.1 Dataset

The dataset for this task is also synthetically created. It resides in the folder **hw2_data/hac**. You can see their visualization in Fig. 2. You don't have the labels since the task is clustering but to remove ambiguity, it is colorized to depict what your expectations from the data should be. The data is in the form of saved numpy array and you can load it by:

```
data1 = np.load('hw2_data/hac/data1.npy')
data2 = np.load('hw2_data/hac/data2.npy')
data3 = np.load('hw2_data/hac/data3.npy')
data4 = np.load('hw2_data/hac/data4.npy')
```

4.2 Task

You are expected to fill the empty functions and use different criteria on each data while using HAC, and comment on every criterion on every data whether it worked fine or not and a short explanation on why the result is like that. You should do:

- Implement and apply HAC using **Single-Linkage Criterion**, **Complete-Linkage Criterion**, **Average-Linkage Criterion**, and **Centroid Criterion** by filling the empty functions. Implementations do not have to be efficient, it is not a requirement of this homework. You can find their definitions in hac.pdf [1] in Lecture Materials.
- After using HAC, we get only one cluster that consists of every data point since the algorithm runs until it merges all the clusters in the active set. However, to see the effects of the criteria functions, in this assignment you are expected to stop the merging of the last k clusters (stop_length in the arguments specify that) where k is the number of clusters which may change from dataset to dataset. For data1, data2, data3 this number is 2 and for the data4, this number is 4.
- After getting k clusters for each data and for each criterion (k may differ from data to data), plot the data points and colorize them according to their clusters to show the results of the algorithms. All the data provided to you are two-dimensional; therefore, it can be plotted easily.
- Try to describe the behaviour of different criteria on each dataset. A good criterion function may differ from dataset to dataset and may change according to our expectations from the task. For each dataset, give a short reason for each criterion function why it was or it was not suitable for that dataset.

5 Specifications

- Please provide a README file that describes how one can run your code do the experiments. You can also leave your dataset folder there for convenience since it does not take too much space.
- The codes must be in Python3. You are not allowed to use libraries that have an implementation of KNN, K-means, HAC. For example, you cannot use scikit-learn. However, you can and expected to use numpy.
- Falsifying results, changing the composition of training and test data are strictly forbidden and you will receive 0 if this is the case. Your programs will be examined to see if you have actually reached the results and if it is working correctly.
- The late policy given in the syllabus applies here. You have total of 6 late days for **all** your homeworks but you can spend at most 3 for a specific homework. The late submission penalty will be calculated using $5n^2$, that is, 1 day late submission will cost you 5 points, 2 days will cost you 20 points and 3 days will cost you 45 points. No late submission is accepted after reaching a total of 3 late days.
- Using any piece of code that is not your own is strictly forbidden and constitutes as cheating. This includes friends, previous homeworks, or the internet. The violators will be punished according to the department regulations. You can use the codes given in the recitation.
- Follow the course page on ODTUCLASS for any updates and clarifications. Please ask your questions on Homework2 discussion section of ODTUCLASS instead of e-mailing if the question does not contain code or solution.

6 Submission

Submission will be done via ODTUCLASS. If you do not have access to ODTUCLASS send send an email to "artun@ceng.metu.edu.tr" as soon as possible. You will submit a zip file called "hw2.zip" that contains all your source code (including mini testers), hw2_data, README file, and your report in a pdf format compiled from the given latex file.

References

[1] R. P. Adams, “Hierarchical agglomerative clustering.”

[2] R. Fisher, “Iris flower dataset,” 1936.