HOMEWORK #1

Software Project (0368-2161)

Due Date: 07/12/2021

1 Introduction

The K-means algorithm is a popular clustering method for finding a partition of N unlabeled observations into K distinct clusters, where K is a parameter of the method. In this assignment you will implement this algorithm in both Python and C. The goals of the assignment are:

- Practice the material taught in class both for C and Python.
- Transform a known algorithm into working executable code.
- Read input and process it.
- Work with files.
- Create an interface for programs.
- Experience the difference in the programming effort and the running time of both languages.

1.1 K-means

Given a set of N datapoints $x_1, x_2, \ldots, x_N \in R^d$, the goal is to group the data into $K \in N$ clusters, each datapoint is assigned to exactly one cluster and the number of clusters K is such that $1 < K < N$. We will denote the group of clusters by $S_1, S_2, \ldots, S_K$, each cluster S_j is represented by it's centroid which is the mean $\mu_j \in R^d$ of the cluster's members.

Algorithm 1 k-means clustering algorithm

1: Initialize centroids $\mu_1, \mu_2, \ldots, \mu_K$ as first k datapoints $x_1, x_2, \ldots, x_K$
2: **repeat**
3: for $x_i, 0 < i < N$:
4: Assign x_i to the closest cluster S_j: $\underset{S_j}{\text{argmin}}\ (x_i - \mu_j)^2,\ \forall j\ 1 \leq j \leq K$
5: for $\mu_k, 0 < k < K$:
6: Update the centroids: $\mu_k = \frac{\sum_{x_l \in S_k} x_l}{|S_k|}$
7:
8: **until** convergence: [1]$(\|\Delta\mu\|_2 < \epsilon)\ OR\ (iteration\_number = max\_iter)$

[1] $\|\Delta\mu\|_2$: Euclidean Norm for each one of the centroids doesn't change more than ϵ.
More Info: Euclidean Norm

2 Assignment Description

Implement the k-means algorithm as detailed in 1.1 both in C and Python.
The behavior of the program is as follows:

1. The program receives the input: K, filename and optional(max_iter):
   (a) K – the number of clusters required.
   (b) input_filename - *.txt file that contains datapoints separated by commas.
   (c) output_filename - *.txt file that contains the centroids separated by commas.
   (d) max_iter – the maximum number of iterations of the K-means algorithm, if not provided the default value is 200.
2. Apply K-means on the input, and return the final centroids within the output file.
3. use $\epsilon = 0.001$

2.1 Compile and Running

2.1.1 C

The program must compile cleanly (no errors, no warnings) when running the following command:

```
$gcc -ansi -Wall -Wextra -Werror -pedantic-errors kmeans.c -lm -o hw1
```

After successfully running the above command, an executable file called hw1 will be created. Now you can run your program by executing the following line on Nova(assuming K = 3, max_iter = 100, filename = "input.txt"):

```
$./hw1 3 100 input.txt output.txt
```

See example below:

```
$#providing max_iter=100
$./hw1 3 100 input.txt output.txt

$#not providing max_iter
$./hw1 3 input.txt output.txt
```

[1]

2.1.2 Python

Your program must be executed by (no errors, no warnings) the following line on Nova, with: K = 3, max_iter = 100:

```
$python3 kmeans.py 3 100  input.txt  output.txt
```

See examples below:

```
$#providing max_iter=100
$python3 kmeans.py 3 100 input.txt  output.txt
```

```
$#not providing max_iter
$python3 kmeans.py 3  input.txt  output.txt
```

2.2 Assumptions and requirements:

Note that the following list applies to both programs in this assignment:

1. You may assume that the input file is in the correct format and that it is provided.
2. Validate that the command line arguments are in correct format.
3. Outputs must be formatted to 4 decimal places (use: '%.4f') in both languages, for example:
   - 8.88885 ⇒ 8.8888
   - 5.92237098749999997906 ⇒ 5.9224
   - 2.231 ⇒ 2.2310
4. Learn about and consider using the following: `input()`, `split()` in your Python implementation.
5. Learn about and consider using the following: `fopen()`, `fclose()`, `fscanf()`, `fprintf()`, `fgetc()`, `feof()` in your C implementation.
6. 3 input files and their corresponding output files examples are provided within the assignment in Moodle. (**YES**, the files have an extra empty row and this is the expected behaviour)
7. Handle errors as following:
   (a) In case of invalid input, print "Invalid Input!" and terminate the program.
   (b) Else, print "An Error Has Occurred" and terminate.
8. Do not forget to free any memory you allocated.
9. For successful running, the C program must return 0 otherwise 1.
10. You can assume that all given data points are different.
11. You may not import external includes (in C) or modules (in Python) that are not mentioned in this document.
12. Use `double` in C and `float` in Python for all vector's elements.

2.3 Submission

1. Please submit a file named `id1_id2_assignment1.zip` via Moodle, where id1 and id2 are the ids of the partners.
   (a) In case of individual submission, id2 must be 111111111
   (b) Put the following files **ONLY** in a folder called `id1_id2_assignment1`:
      i. `kmeans.c`
      ii. `kmeans.py`
   (c) Zip the folder using the following Linux cmd:

```
$zip -r id1_id2_assignment1.zip id1_id2_assignment1
```

   Do not use other ways to create the zip!

2.4 Remarks

For any question regarding the assignment, please post at the HW_1 discussion forum.