CSC3022F: Machine Learning

Machine Learning – Assignment 1

K-Means Clustering

Department of Computer Science
University of Cape Town, South Africa

Problem Description

Implement (in *Python*) the *K-means* clustering algorithm [Jin and Han, 2010] with a Euclidean distance metric. See the online tutorial at:

http://www.saedsayad.com/clustering_kmeans.htm

Use the implemented K-means algorithm to cluster the following 8 examples (see table 1) into 3 clusters. When running K-means, set the initial seeds (initial centroid of each cluster) as example numbers: 1, 4 and 7 (table 1).

Table 1: Data (examples have two attributes: X, Y, both in range: [1, 10]).

Example Number	X	Y
1	2	10
2	2	5
3	8	4
4	5	8
5	7	5
6	6	4
7	1	2
8	4	9

Question 1

How many iterations are needed for k-means to converge?

In a text file output the results of each iteration (for each cluster, list the examples that fall into each cluster), and the centroids of each cluster, **e.g.:**

```
 Iteration 1

 Cluster 1: 1, 2, 3
Centroid: (3.0, 9.5)

 Cluster 2: 4, 5, 6
Centroid: (6.5, 5.25)

 Cluster 3: 7, 8
Centroid: (1.5, 3.5)

    ...

 Iteration N

 Cluster 1: 8, 7, 6
Centroid: (1.5, 3.5)

 Cluster 2: 5, 4, 3
Centroid: (6.5, 5.25)

 Cluster 3: 2, 1
Centroid: (3.0, 9.5)
```

In a ZIP file named as your *student number* (e.g. GWRBRA001.ZIP), place the *py* source code and the output text file (answer to question 1), and upload the ZIP file to the assignments tab in *Vula* before 6.00 PM, 21 May, 2021.

* * *

References

[Jin and Han, 2010] Jin, X. and Han, J. (2010). K-means clustering. In Sammut, C. and Webb, G., editors, *Encyclopedia of Machine Learning*, pages 563–564. Springer, Boston, USA.

Please Note the following for ALL ML Assignments:

1. A working Makefile must be submitted. If the tutor cannot build a python virtual environment on nightmare.cs by typing make, you will only receive **50%** of your final mark.

2. You must use version control from the get-go. This means that there must be a .git folder alongside the code in your project folder. A **10%** penalty will apply should you fail to include a local repository in your submission.

   With regards to git usage, please note the following:

   -10% - usage of git is absent. This refers to both the absence of a git repo and undeniable evidence that the student used git as a last minute attempt to avoid being penalized.

   -5% - Commit messages are meaningless or lack descriptive clarity. eg: "First", "Second", "Histogram" and "fixed bug" are examples of bad commit messages. A student who is found to have violated this requirement for numerous commits will receive this penalty.

   -5% - frequency of commits. Git practices advocate for frequent commits that are small in scope. Students should ideally be committing their work after a single feature has been added, removed or modified. Tutors will look at the contents of each commit to determine whether this penalty is applicable. A student who commits seemingly unrelated work in large batches on two or more occasions will receive this penalty.

   Please note that all of the git related penalties are cumulative and are capped at -10% (ie: You may not receive more than -10% for git related penalties). The assignment brief has been updated to reflect this new information.

   We cannot provide a definitive number of commits that determine whether or not your git usage is appropriate. It is entirely solution dependent and needs to be accessed on an individual level. All we are looking for is that a student has actually taken the time to think about what actually constitutes a feature in the context of their solution and applied git best practices accordingly.

3. You must provide a README file explaining what each file submitted does and how it fits into the program as a whole. The README file should **not** explain any theory that you have used. The README is used by the tutors if they encounter any problems.

4. Do **not** hand in any binary files. Do **not** add binaries (.o files and your executable) to your local repository.

5. Please ensure that your tarball works and is not corrupt (you can check this by trying to downloading your submission and extracting the contents of your tarball - make this a habit!). Corrupt or non-working tarballs will not be marked - **no exceptions.**

6. A 10% penalty per day will be incurred for all late submissions. No hand-ins will be accepted if later than 3 days.

7. **DO NOT COPY. All code submitted must be your own.** ***Copying is punishable by 0 and can cause a blotch on your academic record.*** **Scripts will be used to check that code submitted is unique.**

8. You are **NOT** allowed to simply download python packages to solve your coding problems. Unless specifically stated in the Assignment brief, we expect all of your code to be your own.