Assignment 4: K-Means Clustering

Assignment Overview

In this assignment you will implement the k-means clustering algorithm. You will use it to cluster a data set. For this assignment, we will use a data set from the UC Irvine Machine Learning Repository at: https://archive.ics.uci.edu/ml/index.html.

Write your own code!

For this assignment to be an effective learning experience, you must write your own code! **Do no share code with other students in the class!!**

Here's why:

- The most obvious reason is that it will be a huge temptation to cheat: if you include code written by anyone else in your solution to the assignment, you will be cheating. As mentioned in the syllabus, this is a very serious offense, and may lead to you failing the class.
- However, even if you do not directly include any code you look at in your solution, it surely will influence your coding. Put another way, it will short-circuit the process of you figuring out how to solve the problem, and will thus decrease how much you learn.

So, just don't look on the web for any code relevant to this problem. Don't do it.

Format of data file

The data file that you are clustering is a database related to iris plants. A complete description can be found here: https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.names

You will use the file at https://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data as your input file.

Each line of the csv file looks something like this: 5.1,3.5,1.4,0.2,Iris-setosa

It consists of four floating point values and a text label for the type of iris plant.

The four floating point attributes correspond to:

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm

The string attribute is the iris class, one of the following:

-- Iris Setosa
-- Iris Versicolour
-- Iris Virginica

Submission Details

What you will turn in: `lastname_firstname_clustering.py`

Program Description

Your program should do the following:
1. Read the data from the file. Use only the floating point values for the clustering. Don't discard the class information. While you can't use it for clustering, you will need it later for assigning names to the clusters and for checking the accuracy of the clusters.
2. **Apply the k-means algorithm to find clusters.** (There are 3 natural clusters in the case of the iris data.) (See below for more information on k-means.) **Use Euclidean distance** as your distance measure.
3. **Assign each final cluster a name** by choosing the most frequently occurring class label of the examples in the cluster.
4. **Find the number of data points that were put in clusters in which they didn't belong** (based on having a different class label than the cluster name).

k-means algorithm:
Given k initial points that will act as the centroids of the clusters for the first iteration, you will run the standard k-means clustering algorithm that we discussed in class.

- For each point, place it in the cluster whose current centroid it is nearest
- After all points are assigned, update the locations of centroids of the k clusters
- Repeat for the specified number of iterations.

Output of your program

The program will produce output of the form:

Cluster <clustername1>:
(List of points in that cluster, one per line)

Cluster <clustername2>:
(List of points in that cluster, one per line)

Cluster <clustername3>:
(List of points in that cluster, one per line)

Number of points assigned to wrong cluster:
(number of points)

Running your code

python lastname_firstname_clustering.py dataFileName k iter initialPoints

where:

dataFileName is a string indicating the name of the data file to be clustered
k is an integer representing the number of clusters (three in the case of the iris data set)
iter is the number of iterations for the k-means clustering to run

initialPoints is a string indicating the name of a file that contains a list of data points that are to be used as the starting centroids for each cluster

Testing your code

The sample command to execute is :
python3.4 clustering.py iris.data 3 10 initialPoints

We will provide sample output.