Homework 3: K-Means Clustering

Yohan Jhaveri

February 25th 2020

1 Collaboration Statement

This code was my own work and it was written without consulting any sources outside of those approved by the instructor. I did not collaborate with any student for this homework.

2 Dataset Descriptions

2.1 Iris Dataset:

http://archive.ics.uci.edu/ml/datasets/Iris

- Background: This is perhaps the best known database to be found in the pattern recognition literature. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other two however the latter are not linearly separable from each other.
- Instances: 150 instances
- Attributes: 4 attributes
    1. Sepal Length (cm)
    2. Sepal Width (cm)
    3. Petal Length (cm)
    4. petal width (cm)
- Class Frequencies:
    1. Iris Setosa: 50
    2. Iris Versicolour: 50
    3. Iris Virginica: 50

2.2 Wine Dataset:

http://archive.ics.uci.edu/ml/datasets/Wine

- Background: The data is the result of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines.
- Instances: 178 instances
- Attributes: 13 attributes
  1. Alcohol
  2. Malic acid
  3. Ash
  4. Alcalinity of ash
  5. Magnesium
  6. Total phenols
  7. Flavanoids
  8. Nonflavanoid phenols
  9. Proanthocyanins
  10. Color intensity
  11. Hue
  12. OD280/OD315 of diluted wines
  13. Proline
- Class Frequencies:
  1. Class 1: 59
  2. Class 2: 71
  3. Class 3: 48

3 Implementation

3.1 Pre-processing:

I did not perform any preprocessing on the data other than removing the class label while reading the data in.

3.2 Setup:

3.2.1 Helper Functions:

The helper functions employed in the implementation were:

- **euclidean_distance(x, y)**: calculates the euclidean distance between vectors x and y
- **variance(clusters, centroids)**: calculates the total SSE of all clusters
- **silhouette(clusters, centroids)**: calculates the silhouette coefficient of the clusters
- **classify(centroids)**: assigns data points to clusters based on given cluster centroids
- **select_random_centroids(k)**: initializes random data points as cluster centroids

3.2.2 Steps:

The steps taken in the algorithm were:

1. Read in input file and store data points in a list
2. Initialize cluster centroids using **select_random_centroids(k)**
3. Classify data points according to randomly selected centroids using **classify(centroids)**
4. ***Begin while loop***
5. If any cluster contains 0 data points, re-initialize centroids using **select_random_centroids(k)**
6. Else find the mean of data points in the same cluster and label them as the new cluster centroids
7. Classify data points according these new centroids to generate new clusters using **classify(centroids)**
8. While cluster classification changes from iteration to iteration, ***continue while loop***
9. If the clusters do not change when classified with the newly calculated means, ***exit while loop***
10. Write final cluster labels to output file along with SSE and silhouette coefficient

4 Results

4.1 Iris Dataset:

- k = 2
  - SSE: 16.4717
  - Silhouette: 0.6808
- k = 3
  - SSE: 15.1331
  - Silhouette: 0.5510

- k = 4
  - SSE: 15.09125
  - Silhouette: 0.4951
- k = 5
  - SSE: 14.9055
  - Silhouette: 0.4929
- k = 6
  - SSE: 15.3983
  - Silhouette: 0.3291
- k = 7
  - SSE: 15.2081
  - Silhouette: 0.3622
- k = 8
  - SSE: 16.3628
  - Silhouette: 0.4406
- k = 9
  - SSE: 15.4655
  - Silhouette: 0.2693
- k = 10
  - SSE: 16.2241
  - Silhouette: 0.2699

4.2 Wine Dataset:

- k = 2
  - SSE: 3012.0089
  - Silhouette: 0.6543
- k = 3
  - SSE: 2577.0286
  - Silhouette: 0.5731
- k = 4
  - SSE: 2295.8989

  - Silhouette: 0.5617

- k = 5
  - SSE: 2088.8513
  - Silhouette: 0.5607
- k = 6
  - SSE: 2042.3474
  - Silhouette: 0.5043
- k = 7
  - SSE: 1839.6220
  - Silhouette: 0.5200
- k = 8
  - SSE: 1787.6400
  - Silhouette: 0.5008
- k = 9
  - SSE: 1712.9563
  - Silhouette: 0.4884
- k = 10
  - SSE: 1572.8955
  - Silhouette: 0.5094

5 Experiences

Overall, I really enjoyed implementing K-Means Clustering. The algorithm is very intuitive and it was great to finally learn an unsupervised learning algorithm and its implementation after having learnt only supervised learning algorithms in CS334. I learned a few tricks using excepting errors and realized the power of python list comprehensions (especially line 33 of my code). I also learned about the strategy of silhouette coefficient in determining the "goodness" of a cluster.