## HW3: K-means Clustering Report

Date: 10/14/2020

Binhui Xu(G01172196)

Username on miner2: WhitneyXu

LeadBoard Score: 0.91(iris)/0.54(new\_images)

## Approach:

* **Read and preprocessing the test data:** read two datasets iris\_data and new\_image from txt files into numpy array ‘test’, and the array size are 150\*4 and 10000\*784 respectively. For the image\_data, since it contains a range of values from 0 to 255, it has to be normalized, which ensures that each input parameter(pixel) has a similar data distribution[3]. So the image input is divided by 255 and input values are in range of [0,1]. For the iris data normalization, I utilize the normalize from *sklearn.preprocessing*.

*Function: read\_image\_data(); read\_iris\_data()*

* **K-means clustering:**

1. Initialize k centroids by kmean++ algorithm, which could select initial cluster centers for k-mean clustering in a smart way to speed up convergence.
2. Set the maximum iteration of update centroids to 300. Keep iterating until there is no change to the centroids. i.e. assignment of data points to clusters isn’t changing:

* Compute the sum of the Euclidean distance, cosine similarity, correlation between data points and all centroids. On the basis of results, I found that correlation shows better performance for both iris and new\_img datasets.
* Assign each data point to the closet cluster(centroid).
* Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

1. Compute the silhouette score and SSE.
2. For the leaderboard evaluation, repeat 1 to 3 for 10 times, and return the best clustering result with the highest silhouette score and smallest SSE. Output the result into ‘img\_sil.txt’ and ‘img\_sse.txt’.

*Functions: kmeans(test,K); k\_means(test); km\_pluss\_pluss(test,K); SSE(test,clusters,centers,K); silhouette(data,predicted); correlation(v1,v2); Euclidean(v1,v2); consine\_sim(v1,v2)*

## Different distance measures:

I record the 10 times running result of kmeans and get the following tables. First row is the times of the program processed, second row is the silhouette score that run gain, and the third is the sse that run gain.

Correlation: highest silhouette score occurred in 9th time run that is 0.1551 and minimum SSE occurred in 9th run that is 14798.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| runs | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| silhouette | 0.1484 | 0.1413 | 0.1486 | 0.1465 | 0.1496 | 0.1490 | 0.1485 | 0.1490 | 0.1551 | 0.1497 |
| sse | 49645 | 60662 | 14847 | 80590 | 43032 | 44075 | 68852 | 60472 | 14798 | 54009 |

Euclidean distance: highest silhouette score is 0.1504 and minimum SSE is 25696

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| runs | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Silhouette | 0.1378 | 0.1375 | 0.1375 | 0.1283 | 0.1378 | 0.1490 | 0.1331 | 0.1375 | 0.1399 | 0.1504 |
| sse | 52617 | 42346 | 48502 | 42093 | 43115 | 88430 | 61296 | 54474 | 56399 | 25696 |

Consine similarity: highest silhouette score is 0.1508 and minimum SSE is 18165

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| runs | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Silhouette | 0.1508 | 0.1477 | 0.1508 | 0.1476 | 0.1377 | 0.1504 | 0.1488 | 0.1501 | 0.1481 | 0.1478 |
| sse | 18165 | 62906 | 83552 | 80286 | 24809 | 56797 | 43592 | 69011 | 47366 | 69980 |

According to the three tables above, I find that using correlation as the distance measures can get higher silhouette score and lower SSE, this is the reason that I consider correlation a better choice for the dataset of new\_image.

Same with new\_image, I also consider correlation a better choice for the dataset of iris.

## Evaluate K-means clusters by elbow method:

* Elbow method gives us an idea on what a good k number of clusters would be based on the sum of squared distance (SSE) between data points and their assigned clusters’ centroids. We pick k at the spot where SSE starts to flatten out and forming an elbow. [1]. Fig. 1 Shows the SSE curve of the new\_image clustering with the value of K increasing from 2 to 20. And it can be observed that when k=9 the curve might form an elbow and flatten out. In addition, Fig.2 shows the SSE curve of the iris\_data clustering with the value of K increasing from 2 to 12, which can be observed that when k=4 the curve might form an elbow and flatten out.

Chart, line chart

Description automatically generated

Fig. 1

Chart, line chart

Description automatically generated

Fig. 2

## Reference:

1. <https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a>
2. <https://www.kdnuggets.com/2020/06/centroid-initialization-k-means-clustering.html>
3. <https://medium.com/@joel_34096/k-means-clustering-for-image-classification-a648f28bdc47>
4. <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html>
5. <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>
6. <https://arxiv.org/pdf/1405.7471.pdf>
7. <https://hlab.stanford.edu/brian/error_sum_of_squares.html>