HW3 – K-means Clustering

**Name:** Aditya Varma Vetukuri

G0ID : G01213246

**Miner name:** AdityaVetukuri

**Accuracy (PART-2):** 0.79

**Accuracy(PART-1):** 1.00

**Rank:** 2

**Objectives for this assignment are the following:**

* Implement K-means and Bisecting K-means algorithms
* Compare the performance of both algorithms
* Think about best metrics for Evaluating clustering solutions.

**Implementation of my K-means Algorithm and Bisecting K-means Algorithm :**

K-means algorithm is an iterative algorithm that tries to partition the dataset into clusters where each data point belongs to **only one cluster.**

* Firstly, I have implemented using K-means using selecting K different random point as initial Centroids. Later I have implemented Kmeans++ for choosing my centroids.
* Now I have calculated distances between Data points and centers using cdist function in spatial with distance metric as ‘euclidean’.
* Now I appended the data points to the closest centroids and formed clusters.
* Now I calculated the mean of the clusters and adjusted the centroid position to the mean
* This process is repeated till we find convergence or the loop reaches Max\_iterations. Convergence is a situation where the new centroids formed are not same as the old centroids.

**K-means++ :**

* We initialize the centroids list and append a randomly selected data point to the list.
* We calculate points from the nearest centroid and calculate the minimum distances and append to our distance list. As we start with one that will be the nearest centroid.
* Select the data point with maximum distance in our minimum distance list as our next centroid.
* This process is repeated recursively till we get our K centroids.

**Bisecting K means :**

* Bisecting K-means runs the K-means with n\_clusters = 2
* We calculate the length of both the clusters and run bisecting K-means on the bigger cluster and append the smaller cluster.
* This process is repeated till we get our required clusters.

**Part-1 (Iris Clustering)** :

* The data set consists of 150 instances with 4 features i.e sepal length,sepal width, petal length, petal width all in cms.
* I have used Normalizer with the inbuilt parameter norm = ‘max’ to normalize the data and used my K-means classifier to Cluster the data.
* I have achieved an accuracy score of 1.00 for this assignment.

A close up of a device

Description automatically generated

**Part – 2 (Image Clustering) :**

The image dataset consists of 10,740 images with 784 features for each image without any labels.

A picture containing game, strainer, brush

Description automatically generated

Above are the samples images of the dataset.

Following are the preprocess, feature reduction and extraction techniques that I’ve used or tried for this Assignment.

**Pre-Processing :**

* **Binarizing the data :** I have binarized each image into zeroes and ones using Binarizer() from sklearn.preprocessing.
* **Normalization :** After binarization I normalized all values using Normalizer from sklearn.preprocessing.
* **Downsampling** : To Downscale the image I have first reshaped the data to (10740,28,28) and used downscale\_local\_mean() function from skimage.transform. Now I reshaped the image to (10740,196).
* **CV2.blur :** I tried blurring the image and just preserving the edges of the digit so that it will be easy to cluster. But it didn’t work out really well for my model.
* **Caluculate edge features :** From skiimage.filters I’ve calculated the horizontal and vertical features using prewitt\_h and prewitt\_v.
* **Autoencoder** : I have used simple Autoencoder to remove noise from the data using Input and Dense from keras.layers and tried removing excess noise from the image.

**Feature Extraction :**

**Principal Component Analysis :** Firstly I’ve calculated n\_components which wont effect the variance using np.cumsum() and used PCA. PCA wasn’t working that well for images as it stores the larger distances too unlike TSNE.

**TSNE :** It creates a probability distribution using the Gaussian distribution that defines the relationship between the points in high-dimensional space.

**Metrics used for this Assignment :**

* **Silhoutte coefficient :** The Silhoutte coefficient is calculated using the mean intra-cluster distance and the mean nearest-cluster distance for each sample. The Silhoutte coefficient for a sample is (b – a) / max (a , b)
* **Sum of Squares error :** SSE is the sum of the squared differences between each observation and it’s group mean. It can be used as a measure of variation within a cluster. If all cases within a cluster are identical the SSE would be then equal to 0.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| S.NO | K-means | BISECTING K-MEANS | NORMALIZER &Binarize | DOWNSAMPLING | PCA | TSNE | Silhoutte Coefficient | Accuracy |
| 1 | Yes | No | Yes | Yes | No | Yes | 0.1224280 | 0.79 |
| 2 | No | Yes | Yes | Yes | No | Yes | 0.142312 | 0.71 |
| 3 | Yes | No | Yes | Yes | No | No | 0.195612 | 0.48 |
| 4 | No | Yes | Yes | Yes | No | No | 0.203412 | 0.36 |
| 5 | Yes | No | Yes | Yes | Yes | No | 0.162132 | 0.59 |
| 6 | No | Yes | Yes | Yes | Yes | No | 0.175923 | 0.42 |
| 7 | Yes | No | No | No | No | Yes | 0.146213 | 0.71 |
| 8 | Yes | No | Yes | No | No | Yes | 0.132124 | 0.76 |

**Below are the graphs based for SSE values against K values ranging from 2 to 20 :**

**For the Iris DataSet :**

A close up of a map

Description automatically generated

**For the Image dataset:**

**SSE VALUES WITH TSNE For K-means (EUCLIDEAN):**

A close up of a map

Description automatically generated

**SSE VALUES WITH TSNE For K-means (COSINE):**

A screenshot of a map

Description automatically generated

**SSE VALUES WITHOUT TSNE:**

A close up of a map

Description automatically generated

**SSE VALUES FOR BISECTING K MEANS:**

**A close up of a map

Description automatically generated**

**Instructions to execute my code** :

* I have three python scripts named Kmeans\_part2 which is the code with which I got the best accuracy on Miner on Image dataset
* I have Bisecting K-means Implementation on separate python file named Bisecting Kmeans Implementaion.
* I have KmeansonIris.py which is k-means implementation on Iris dataset for part – 1
* The last python is Graphs.py to change my different feature extractions and look for graphs.