Kshitij Chhajed

K-Means Clustering

24-Oct-2021

1. Problem statement

Using K-Mean generate clusters and access the quality of the clusters using ASC (Average Silhouette Coefficient) and DI (Dunn's Index) evaluation metrics.

2. Dataset

I have used the Cifar-10 dataset of images. It can be downloaded from the link: https://www.cs.toronto.edu/~kriz/cifar.html. The CIFAR-10 and CIFAR-100 are labeled subsets of the 80 million tiny images dataset. They were collected by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.The dataset is divided into five training batches and one test batch, each with 10000 images. The test batch contains exactly 1000 randomly-selected images from each class. Each example is a 32x32 image, associated with a label from 10 classes. Each image is 32 pixels in height and 32 pixels in width, for a total of 1024 pixels in total. This pixel-value is an integer between 0 and 255. The training and test data sets have 1025 columns including the labels.

3. Languages/tools used

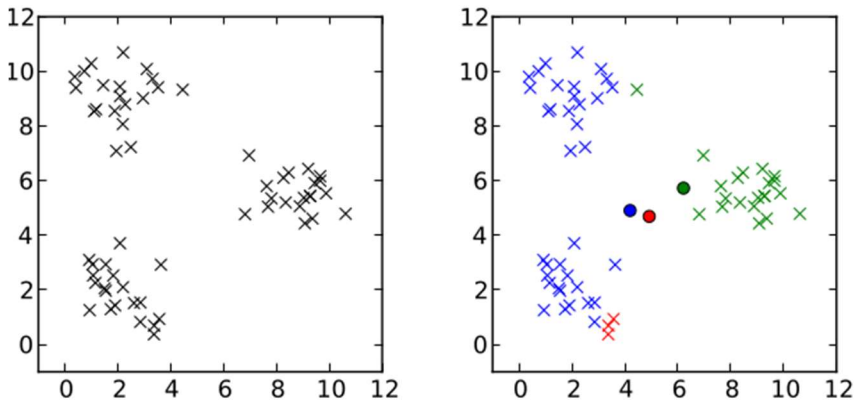
I have used Google Colab for implementation. I have used below libraries:
keras – to load the image dataset
validclust, sklearns.metrics – to calculate ASC (Average Silhouette Coefficient) and Dunn's Index
cv2 – to convert image to grayscale
numpy – to perform numerical operations on matrices
copy – to perform deepcopy of data

4. Data preprocessing

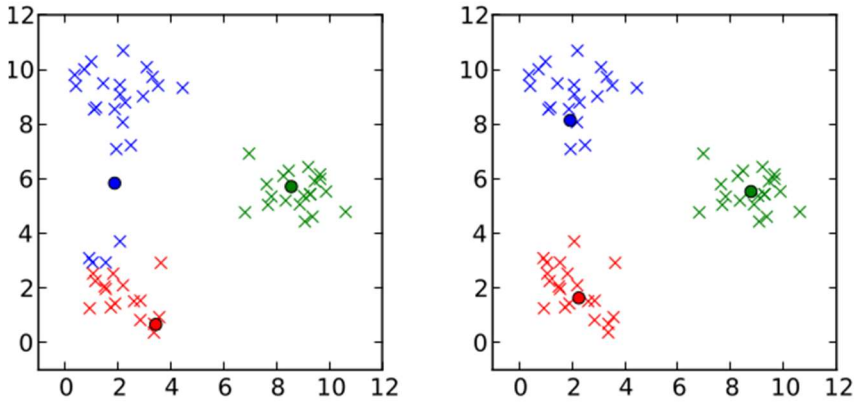
The dataset was loaded into x_train, x_test, y_train, y_test variables using keras function. I have trained K-means algorithm on the data captured in x_test which of 10000 samples. The data is later converted from RGB to grayscale (i.e. 10000*32*32*3 to 10000*32*32*1). The data was then reshaped to (10000*1024) to perform operations for clustering. Here, one image corresponding to one data-point which needs to be grouped in a cluster.

5. K-Means clustering

K-Means clustering is an unsupervised machine learning method. Cluster means a group of data instances where the properties of the instances are similar. It is a centroid based algorithm where the target is to minimize the Euclidean distance of the datapoints from the centre (centroid) of the cluster to which they belong. In order to achieve this, we choose any n no. of centroids from the dataset if we need to classify the data in n no. of clusters. Later the Euclidean distance of all the datapoints is calculated from the centroids and the datapoints are assigned to clusters whose centroid is nearest to them. The values of datapoints thus obtained in a cluster are averaged to find a new centroid for the cluster. Now, Euclidean distance of the datapoints from these new centroids is calculated and again the datapoints are classified into clusters which can lead to new centroids. This continues till the difference between the new and old values of centroids doesn't become neglible or the number of iterations for this process are not exhausted. Below image demonstrates clustering in 3 clusters step-by-step.



(a) dataset. (b) step 1.



(c) step 2. (d) step 3.

6. Evaluation metrics

K-Means clustering can be evaluated based on different techniques. Extrinsic methods like homogeneity score, completeness score can be used if we have the original labels available with us. Intrinsic methods like silhouette coefficient and Dunn's Index can be used if we don't have the ground truth labels.

A. Silhouette coefficient

$$s(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}}$$

where,
- s(o) is the silhouette coefficient of the data point o
- a(o) is the average distance between o and all the other data points in the cluster to which o belongs
- b(o) is the minimum average distance from o to all clusters to which o does not belong

The value of the silhouette coefficient is between [-1, 1]. A score of 1 denotes the best meaning that the data point **o** is very compact within the cluster to which it belongs and far away from the other clusters**.** The worst value is -1. Values near 0 denote overlapping clusters.

B. Dunn's Index

Dunn index is the ratio of the minimum of Inter-cluster distances and maximum of Intra-cluster distances.

$$\text{Dunn Index} = \frac{\min(\text{Inter cluster distance})}{\max(\text{Intra cluster distance})}$$

The Inter-cluster distance is distance between centroids of two different clusters whereas Intra-cluster distance is the distance of the datapoints within a cluster with respect to its centroid.

Intra cluster distance

Inter cluster distance

7. Algorithm

1. Start.
2. Randomly choose no. of clusters and one centroid for each cluster. Here, I have taken 10 random centroids since I choose the data to be divided in 10 clusters.
3. Repeat steps 3 to 8 till either the no. of iterations (60 in my case) are finished or the termination condition (difference between newly calculated centroids and previous being zero) are met.
4. Calculate Euclidean distance for every datapoint from all the centroids (dist[] array).
5. Find the cluster number of the centroid with minimum value from the datapoint (variable idx)
6. Store the datapoint in an array (2-dimensional) of clusters where every row signifies a cluster and all the columns corresponding to the row denote the datapoints associated with it (cluster[][] array).
7. Also, parallel to step 5, store the cluster number for every datapoint (y_pred[] array). This should be an array of 10000 instances since the dataset being worked upon has 10000 datapoints.
8. Calculate the average of all instances in a cluster and redefine the centroid as this value (a[] array). If there exists not a single instance in any one of the clusters, select a random datapoint and assign it as the new centroid for that cluster (also stored in a[] array).
9. Calculate the difference between the old centroid values (old_a[] array) and the new centroid values (a[] array). If the difference diminishes to a minimum threshold (0.00001), stop the iterations. Otherwise, stop the iterations after a specified value (60).
10. Calculate ASC (score variable) and Dunn's Index (return value of function dunn()). Display both values.
11. Stop.

8. Results

After running K-means clustering algorithm over the dataset, the data converged into different clusters after 55 iterations. ASC value achieved was 0.05585, which was greater than desired value of 0.054 and above. Also, Dunn's Index value of 0.09108 was received which was greater than expected value of 0.089.

```
Iteration:  55 - Diff: 0.0
Silhouette score: 0.05585021207357926
Dunn's index: 0.09108497591578157
```

9. Credits

1. https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/
2. https://towardsdatascience.com/understanding-k-means-clustering-in-machine-learning-6a6e67336aa1
3. https://ludovicarnold.com/teaching/optimization-machine-learning/unsupervised-example-clustering-k-means/
4. https://medium.com/@cmukesh8688/silhouette-analysis-in-k-means-clustering-cefa9a7ad111
5. Numpy library support page
6. https://validclust.readthedocs.io/en/latest/validclust.html