JAVID GULIYEV
SADIG GOJAYEV

Report

Artificial Intelligent project
K-means++

Department of
Computer Science
CS-017

FRENCH-AZERBAIJANI UNIVERSITY

2020

Introduction

Machine Learning allows the systems to make decisions autonomously without any external support. These decisions are made when the machine is able to learn from the data and understand the underlying patterns that are contained within it. Then, through pattern matching and further analysis, they return the outcome which can be a classification or a prediction. There are three important types of Machine Learning Algorithms –

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

Unsupervised Learning

In the case of unsupervised learning algorithm, the data is not explicitly labeled into different classes, that is, there are no labels. The model is able to learn from the data by finding implicit patterns. Unsupervised Learning algorithms identify the data based on their densities, structures, similar segments, and other similar features. Unsupervised Learning Algorithms are based on Hebbian Learning. Cluster analysis is one of the most widely used techniques in supervised learning. Let us look at some of the important algorithms that come under Unsupervised Learning.

What is clustering?

Clustering is the most popular technique in unsupervised learning where data is grouped based on the similarity of the data-points. Clustering has many real-life applications where it can be used in a variety of situations.

The basic principle behind cluster is the assignment of a given set of observations into subgroups or clusters such that observations present in the same cluster possess a degree of similarity. It is the implementation of the human cognitive ability to discern objects based on their nature. For example, when you go out for grocery shopping, you easily distinguish between apples and oranges in a given set containing both of them. You distinguish these two objects based on their color, texture and other sensory information that is processed by your brain. Clustering is an emulation of this process so that machines are able to distinguish between different objects.

It is a method of unsupervised learning since there is no external label attached to the object. The machine has to learn the features and patterns all by itself without any given input-output mapping. The algorithm is able to extract inferences from the nature of data objects and then create distinct classes to group them appropriately.

In clustering machine learning, the algorithm divides the population into different groups such that each data point is similar to the data-points in the same group and dissimilar to the data points in the other groups. On the basis of similarity and dissimilarity, it then assigns appropriate sub-group to the object.

Imagine that you have a group of chocolates and liquorice candies. You are required to separate the two eatables. Intuitively, you are able to separate them based on their appearances. The process of segregating objects into groups based on their respective characteristics is called clustering. In clusters, the features of objects in a group are similar to other objects present in the same group.

Clustering is used in various fields like image recognition, pattern analysis, medical informatics, genomics, data compression etc. It is part of the unsupervised learning algorithm in machine learning. This is because the data-points present are not labelled and there is no explicit mapping of input and outputs. As such, based on the patterns present inside, clustering takes place.

K-means and K-means++

According to the formal definition of K-means clustering – K-means clustering is an iterative algorithm that partitions a group of data containing n values into k subgroups. Each of the n value belongs to the k cluster with the nearest mean.

This means that given a group of objects, we partition that group into several sub-groups. These sub-groups are formed on the basis of their similarity and the distance of each data-point in the sub-group with the mean of their centroid. K-means clustering is the most popular form of an unsupervised learning algorithm. It is easy to understand and implement.

The objective of the K-means clustering is to minimize the Euclidean distance that each point has from the centroid of the cluster. This is known as intra-cluster variance and can be minimized using the following squared error function –

Squared Error FunctionWhere J is the objective function of the centroid of the cluster. K are the number of clusters and n are the number of cases. C is the number of centroids and j is the number of clusters. X is the given data-point from which we have to determine the Euclidean Distance to the centroid. Let us have a look at the algorithm for K-means clustering –

1. First, we randomly initialize and select the k-points. These k-points are the means.
2. We use the Euclidean distance to find data-points that are closest to their centre W and assign to cluster 'W'.
3. Then we calculate the mean of all the points in the cluster which is finding their new centroid.
4. We iteratively repeat step 2 and 3 until all the points are assigned to their respective clusters.

K-Means is a non-hierarchical clustering method. K-means++ is the algorithm which is used to overcome the drawback posed by the k-means algorithm.

This algorithm guarantees a more intelligent introduction of the centroids and improves the nature of the clustering. Leaving the initialization of the mean points the k-means++ algorithm is more or less the same as the conventional k-means algorithm.

1. In the starting we have to select a random first centroid point from the given dataset.
2. Now for every instance say 'i' in the dataset calculate the distance say 'x' from 'i' to the closest, previously chosen centroid.
3. Select the following centroid from the dataset with the end goal that the likelihood of picking a point as centroid is corresponding to the distance from the closest, recently picked centroid.
4. Last 2 steps are repeated until you get k mean points.

Data Preparing

Random Initialization

We need data before implementing k-means clustering.So we created 100x2 (x and y) matrix with 100 random points. Created data was between 1000x400 area. Then we create second array with 4 values : x,y coordinates and gx and gy 2 gaussian values for coordinates.

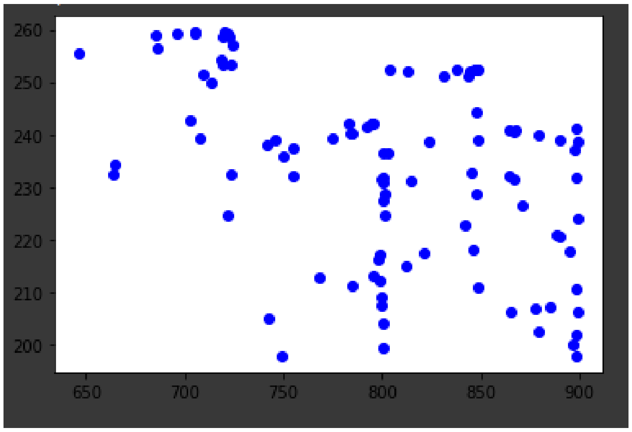
Function : def create_data(points,coordinates,centers) return data matrix and data_c centroid matrix for distribution of gaussian.

Gaussian Distribution

Gaussian distribution (also known as normal distribution) is a bell-shaped curve, and it is assumed that during any measurement values will follow a normal distribution with an equal number of measurements above and below the mean value.

Function : def data_praparing(data,data_c,k) - with the formula below we reform the data

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right)$$



Specifications

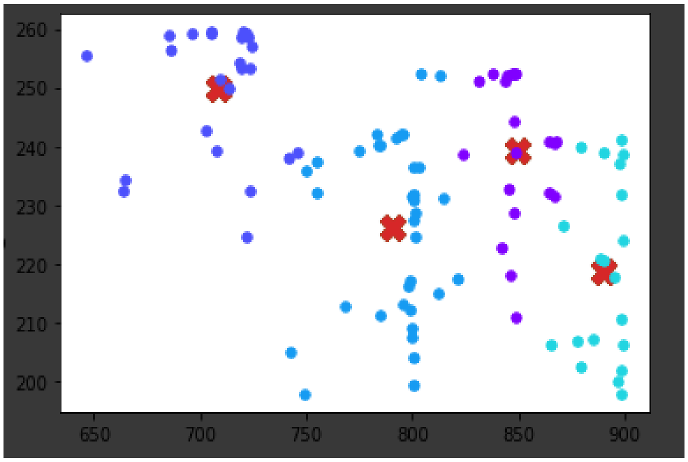
The main idea of this project to analyze/measure the quality of a k-means clustering . There are 2 different versions that have done in this project .

K-means implementation

As introduction was given above, we used core functions - recalculation of clusters and centroids.Except from k-means one, in k-means++ the first initial centres were selected 'precisely' to say. At the end all N clustering inserted to list.

Functions

1. def Centroids_plus(data, k, random_state=42) // initializing centroids with k-means method
2. def recalculate_clusters(D, centroids, k) // data insertion to clusters regarding their distances to centroids
3. def recalculate_centroids(centroids, clusters, k) // recalculating new centroids
4. def k_means(k) // k-means fitting for 1 clustering
5. def n_k_means(k,n) // k-means fitting for n clustering
6. def n_k_means_plot(k,n) // plotting n clustering



Version 1

The first version is all about diversity of n k-means++ clustering. Entropy is used for that.

Entropy

The K-means algorithm is implemented and the respective clusters are obtained. These clusters are compared with the true label data set and the values of Purity and Entropy calculated clusters generated by K-means clusters. In our case we used Entropy to calculate the diversity of clusters, although Gini (purity) is more effective regarding to computation time. def entropy(clusters_list) method used to calculate diversity of n clustering with n different initial conditions(centres). clusters_list contains the list of n clustering for each one the entropy was calculated and number of different clustering obtained. Then with log function total entropy was calculated which detects the compatibility of n centres.If entropy is low then 'k' means are good and vice versa. Entropy is 0 in ideal condition where the is 0 different clustering.

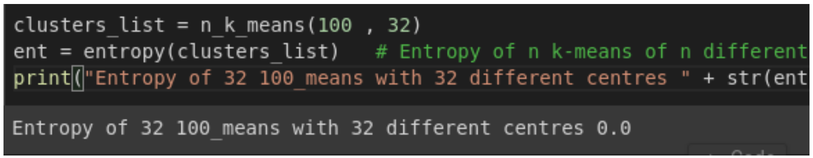
$$H = -\sum p(x) \log p(x)$$

Examples

By assigning entropy(clusters_list) to variable and then print(variable).

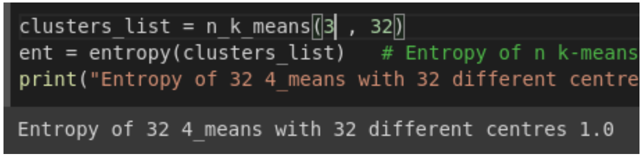
1. entr_k_means = entropy(clusters_list)

2. print(entr_k_means)



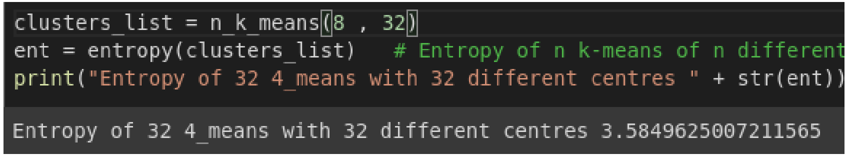
```
clusters_list = n_k_means(100 , 32)
ent = entropy(clusters_list)   # Entropy of n k-means of n different
print("Entropy of 32 100_means with 32 different centres " + str(ent
```

```
Entropy of 32 100_means with 32 different centres 0.0
```



```
clusters_list = n_k_means(3 , 32)
ent = entropy(clusters_list)   # Entropy of n k-means
print("Entropy of 32 4_means with 32 different centre
```

```
Entropy of 32 4_means with 32 different centres 1.0
```



```
clusters_list = n_k_means(8 , 32)
ent = entropy(clusters_list)   # Entropy of n k-means of n different
print("Entropy of 32 4_means with 32 different centres " + str(ent))
```

```
Entropy of 32 4_means with 32 different centres 3.5849625007211565
```

As we can see the entropy of 100 means will be 0 for 100 points as it should be. And for 1 means it will also be 0. So we should deduce the fewest k for clustering by this analyze.

Version 2

The next version is all about measuring quality of cluster with the help of density and scattering variables.

Density

For the calculating density there is need to find rectangle of cluster spread. In the cluster the points should be find with the max and min numbers for 2d cordinates (x and y). However, while finding coordinates of rectangle, its surface is calculated. At the final stage, Number of points in cluster that spread in rectangle divided by surface for to calculate Density value

Scattering

Scattering value is similar to entropy function that described and explained above. Likewise, the value of scattering gives information about how the points grouped in cluster. It means low entropy for cluster witness points grouped with low distance to each other.

Quality

The definition of quality is that how the centeroids (k) placed in data. Low quality means that k-mean is far from clutured points in data. Density and Scattering values helps us scientist to learn k-means quality. Formula below illustrates that for measuring quality, weight of each of the clusters should be multiply to sum of density and scattering weight and values, then sum of those will be calculated.

$$\text{Quality}_k = \sum_{i=1}^{k} w(i)\,(\alpha d(i) + \beta s(i))$$

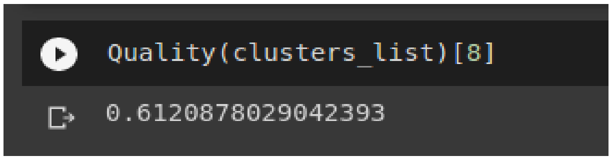
Functions that were implemented

1. density(clusters_list) - calculate denisty of each iteration and returns list of all densities of every cluster of iterations

2. entropy_scatter(clusters_list)-measures and gives list entropies of each iteration

3. Quality(clusters_list) - return list of Quality of iterations

Examples



```
Quality(clusters_list)[8]
0.6120878029042393
```

Quality of 9th clustering.

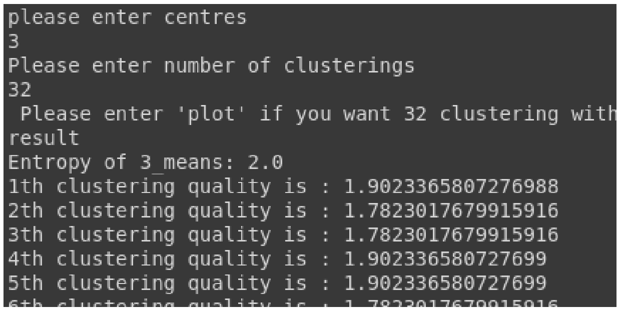
Conclusion

The 2 methods of measurement of k-means clustering were implemented for predicting right value of means.

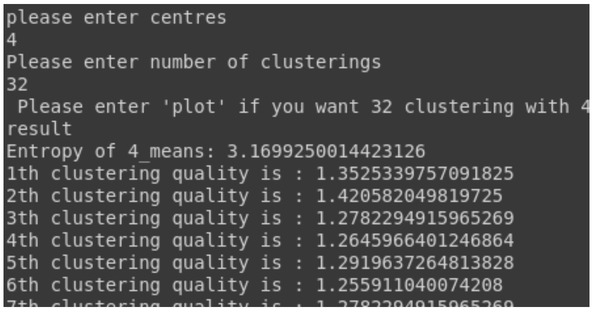
1. Diversity of n k-means

2. Quality of clusters

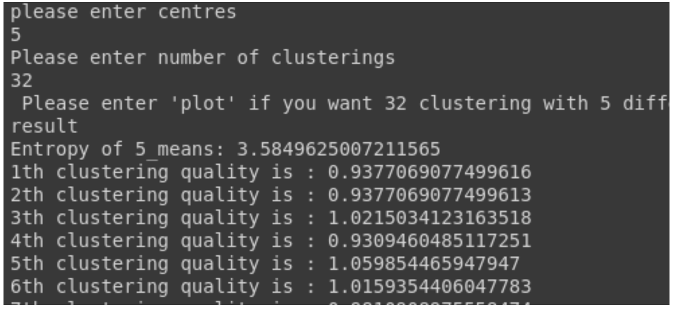
From the implemented methods expected k value have not exactly obtained , the error is +-1 . Data with the 3 or 4 centroids analyzed accurately and the k value predicted precisely same.



```
please enter centres
3
Please enter number of clusterings
32
 Please enter 'plot' if you want 32 clustering with
result
Entropy of 3_means: 2.0
1th clustering quality is : 1.9023365807276988
2th clustering quality is : 1.7823017679915916
3th clustering quality is : 1.7823017679915916
4th clustering quality is : 1.902336580727699
5th clustering quality is : 1.902336580727699
6th clustering quality is : 1.7823017679915916
```



```
please enter centres
4
Please enter number of clusterings
32
 Please enter 'plot' if you want 32 clustering with 4
result
Entropy of 4_means: 3.1699250014423126
1th clustering quality is : 1.3525339757091825
2th clustering quality is : 1.420582049819725
3th clustering quality is : 1.2782294915965269
4th clustering quality is : 1.2645966401246864
5th clustering quality is : 1.2919637264813828
6th clustering quality is : 1.255911040074208
7th clustering quality is : 1.2782294915965269
```



```
please enter centres
5
Please enter number of clusterings
32
 Please enter 'plot' if you want 32 clustering with 5 diff
result
Entropy of 5_means: 3.5849625007211565
1th clustering quality is : 0.9377069077499616
2th clustering quality is : 0.9377069077499613
3th clustering quality is : 1.0215034123163518
4th clustering quality is : 0.9309460485117251
5th clustering quality is : 1.059854465947947
6th clustering quality is : 1.0159354406047783
```

For distributed data with 3 centralized point