Clustering Algorithms: Analysis and Implementation

Siddhant Mittal
University of Florida
s.mittal@ufl.edu

April 25, 2020

Abstract

One of the most interesting field of machine learning is clustering. Clustering are used to put objects that are similar into one category or class. This project report explore three major clustering algorithms K-means, Hierarchical clustering and DBSCAN to understand the field of clustering. I primarily focus on K-means algorithm and use the other algorithms to draw comparison between different approaches and summaries which algorithm to use in which situation at the end of the report. I have also performed various experiments to answer following questions: what is clustering? How it works? How does it perform on different dataset? How to evaluate how good our clustering was?.

1 Introduction

Machine learning algorithms can be classified as supervised learning and unsupervised learning. Supervised learning is when we have set of input which is labeled and our job is to learn patterns in the input data and be able to assign labels to the data which shares same pattern. Such type of problems can further be classified as regression and classification for continuous and discrete dataset respectively. In unsupervised learning the given input data doesn't have any assigned target or labels, so our goal is to find hidden patterns into the data to group them into clusters. Data points in a same clusters share the same pattern or properties. Such type of unsupervised learning problems are referred to as clustering problems.

Clustering problems can be found in many different application such as data mining. knowledge discovery, data compression and pattern recognition. The notion of which constitute good clustering depends on the application and there are many methods for performing clustering. Among clustering algorithms that are based on minimizing a formal objective function, the most widely used and studied is k-means clustering. Formally stating the problem statement for clustering algorithm, given n points in d dimensional feature space we want

to cluster them into K subgroups based on some similarity score like Cosine similarity, Euclidean distance, Manhattan distance etc

The motivation behind choosing this field is that it is very closely related to my research work which I was doing at Google last fall. I used DBSCAN to perform clustering on virtual machine networks in GCP platform. Also I found that this topic is somewhat covered in class so exploring it thoroughly and implementing few such algorithms and see their performance on few datasets will make me more effluent in this field.

In this paper I explain the K-means algorithm, the science behind it, a simple implementation from scratch in python, then I show show methods to evaluate our clustering results, then compare K-means results with other clustering algorithms like Hierarchical clustering and DBSACN. I conducted many experiments with k-means and other clustering algorithms to get meaningful insight into the clustering algorithms. My implementation of algorithms and experiments are done using Google Colab.

2 Data sets

Before describing the algorithms and their implementation lets talk about the datasets we have considered in our experiment. I want to highlight this before as I have shown various figures in each section below and it will give better understanding if you understand a little bit about the dataset. I have selected various dataset specially for purpose of visualization, show effectiveness of algorithm and run time analysis. All dataset are standard scale to remove any feature being overpowered.

2.1 Blob dataset

It is a synthetic data set created using make blob function in sklearn in python. It generates isotropic Gaussian blobs based on our input of number samples, number of dimensions and number of clusters. Synthetic data is used to understand clustering better as we will be knowing before hand the ground truth about the number of clusters in the dataset. The Blob dataset I created have 4 features(dimensions), 200 samples and 5 clusters.

2.2 Shopping list dataset

It is a real world data which I have downloaded from SuperDataScience website. It is 2 dimensional dataset and have 200 samples i.e each data point(customer ID) is represented by annual income and sending score. This dataset is used to help in visualization of our clustering algorithms as we will be able to make 2D plots and show clusters and their centeroids. For higher dimensional dataset it is hard to visualise the clustering, we have to use special tools or draw pairwise 2D graphs for different features but that too is very complicated to visualize.

2.3 Wine dataset

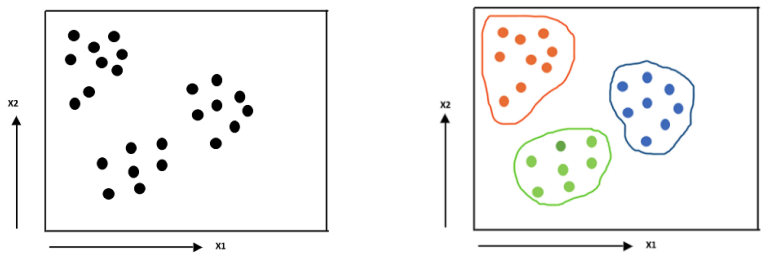
This dataset is downloaded from UCI Machine learning directory here. I have used white wine dataset which have 4898 samples and have 11 features.

2.4 Birch dataset

This is a synthetic 2D data with 100,000 samples and 100 clusters. This data is used both for visualization and to see running time of our clustering algorithms on such a big dataset. This data set can be downloaded here.

3 Clustering Algorithms

Clustering is field of machine learning comes under unsupervised learning. Given a data set without target values we try to find hidden pattern in the data set to group data points into clusters. Let's start with simple example and a plot as shown in figure 1. Each data point in our data set is represented by two features lets call it income and expenditure. If income is considered on the x_1 axis and expenditure is considered x_2 axis then we can plot all the data points on 2D graph as shown in the figure 1. This represents data before clustering and after clustering. These 3 cluster can be thought of groups representing individuals having low income high expenditure, low income low expenditure and high income high expenditure etc. So market strategies can be used to influence individual groups. This type of problems is a basic example of market segmentation and market recommendation problems. There are many clustering algorithm and efficiency of each algorithm depends on dataset and its application. More on this after I introduce different clustering algorithms.



(a) Before clustering (b) After clustering

Figure 1: A simple example to demonstrate clustering

4 K-Means

K-Means is widely used clustering algorithm and is major study of this paper. This algorithm is an iterative algorithm that tries to divide dataset into predefined distinct non-overlapping clusters where each data points belong to different cluster. It tries to group similar data points in cluster to be as close to each other as possible while keeping clusters to as far apart as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the centroids of the clusters i.e the mean of distance of all the data point that belong to that cluster is minimum. Mathematically speaking this can be represented as minimization problem.

$$L = \sum_{i=1}^{n} \sum_{k=1}^{K} \phi_{ik} ||x_i - \theta_k||^2$$

Where

$$\phi_{ik} = 1 \; if \; k = argmin_j ||x_i - \theta_j||^2 \; and \; 0 \; otherwise$$

θ are the K centroids, x are the data points. Jointly optimizing for θ and ϕ is NP-hard

There is no efficient solution known for K-Means problem and some of formulations are NP-hard so we are going to use a heuristic based approach to solve K-Means problem. In which we alternatively optimize just like EM algorithm where there is expectation(E-step) and Maximization(M-step).

The K Means runs in the following steps:

Step 1: Initialize the K centroids.
Step 2: Compute the sum of the squared distance between data points and all centroids.
Step 3: Assign each data point to the closest centroid.
Step 4: Compute the new centroids by taking the average of the all data points that belong to each cluster.
Step 5: Keep iterating from step 2 until there is no change to the centroids or we exceed maximum iteration.

In terms of Expectation-Maximization, Step 2 is the E-step and Step 4 is the M-step. One major drawback which this heuristic approach is that it is too dependent on Step 1 that is initialization of the centroids. Different initialization may lead to different clusters since algorithm may be stuck in a local optimum and may not converge to global optimum. More on drawbacks of K-Means in the later section.

4.1 Initialization

There are several ways to initialize the centroids to start off the K-Means algorithm. I have compared two approaches here one is random selection and the other is Kmeans++. Random selections as the name suggest is selecting centroids randomly from the datsets. The problem with this approach is that if two initial centroids are very near, it would take a lot of iterations for the algorithm to converge. Kmeans++ on other hand chose centroids which are far from one another. This is how it works, the first cluster center is chosen uniformly at random from the dataset, after which each subsequent cluster center is chosen from the remaining data points with probability proportional to its squared distance from the point's closet existing cluster center. Figure 2 shows the comparison between these two approaches using the shopping list data set.

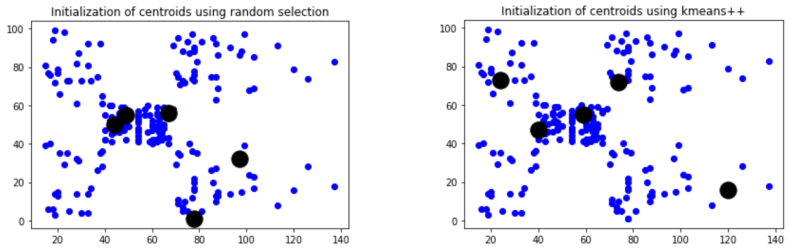


Figure 2: Comparison between random and kmean++ initialization on the shopping list dataset

4.2 Elbow Method

Usually in K-Means algorithm we need to define K in advance but for an unknown dataset how do we decide what should be the value of K to feed to our algorithm. So the question is what is the best value of K? Suppose we consider every data point to be the centroid of a cluster giving us N clusters for N data points. Is it a good model? Obviously no, same is true if we put all the data points in the same cluster giving only one cluster. To find a appropriate K we need to find a clustering in which each data point is nearest to its centroid. In other words sum of squared distances(SSE) of every data point from its corresponding centroid should be as minimum as possible. Following steps are used in Elbow method:

- Perform K means on datset with K ranging from 1 to upper limit.
- We calculate the SSE value of each clustering.
- Plot the value of SSE with the number of clusters K.
- The location of a bend(knee) in the plot is generally considered as an sign of the appropriate number of clusters. The point after which the SSE score don't rapidly decreases.

For the suppose of demonstrating this method lets use the synthetic data created using make blob method from Scikit learn where we know the ground truth that number of clusters are 5. Figure 3 shows the elbow detection on the this data. Figure 4 shows elbow detection on the real world data set.

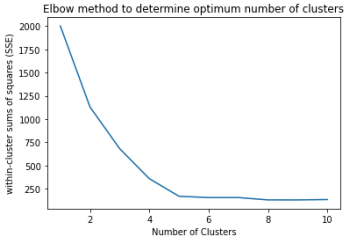


Figure 3: Elbow detection using Blob synthetic data where we know there exist 5 clusters

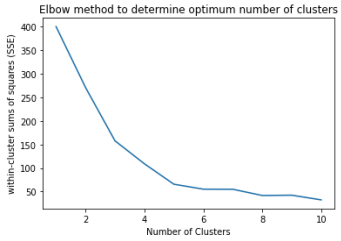


Figure 4: Elbow detection using shopping list data and finding the knee at 5 which is the appropriate number of clusters

5 Other Clustering Algorithm

In this section I will talk about other clustering algorithms briefly which will be used to draw comparison during our experiment with our implemented K-Means algorithm.

6 Hierarchical clustering algorithm

Hierarchical clustering are of two types; agglomerative and divisive. Agglomerative is bottom up approach in which we start with many clusters and merge them together to create bigger cluster based on some similarity score. Divisive on the other hand is top down approach in which we have one big cluster and we keep on dividing it into smaller clusters. We have considered agglomerative approach for our study. Algorithm is simple as follows:

- Consider each data point a cluster.
- Combine two cluster together based on the similarity score.

- Repeat step 2 until we are left with only one cluster.

Now the question is at what time we want to stop our hierarchical clustering. This can be done using Dendrograms which is diagram showing the grouping history. Determine the largest vertical distance that doesn't intersect any of the other clusters. Draw horizontal lines at both end of this vertical lines. The number of optimal clusters is equal to the number of vertical lines going through the horizontal line. This will be more clear from Figure 5. I have used the same dataset shopping list to show optimal number of clusters is 5 as we have seen in Elbow method in K-Means in above section.

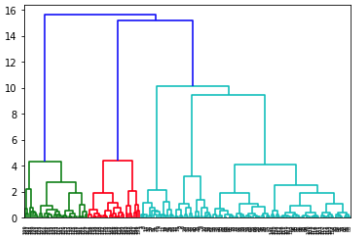


Figure 5: Dendrogram showing grouping history for shopping list dataset showing optimal number of clusters are 5

Let's now talk about similarity score. Similarity score depends on how distance is calculated between clusters, which depends on linkage and distance metric. Distance metric can be Euclidean distance, Manhattan distance etc. Linkage can be: (L is linkage calculated, c_1 and c_2 are two clusters and D is function of our distance metric selected):

- Single Linkage: The distance between two clusters is the shortest distance between two points in each cluster. $L(c_1, c_2) = min(D(x_{ic_1}, x_{jc_2}))$
- Complete Linkage: The distance between two clusters is the longest distance between two points in each cluster. $L(c_1, c_2) = max(D(x_{ic_1}, x_{jc_2}))$
- Average Linkage: The distance between two clusters is the average distance between each points in one cluster to every point in other cluster. $L(c_1, c_2) = \frac{1}{n_{c1} n_{c2}} \sum_{i=1}^{n_{c1}} \sum_{j=1}^{n_{c2}} (D(x_{ic_1}, x_{jc_2}))$
- Ward Linkage: The distance between clusters is the sum of squared differences within all clusters.

In our implementation we use ward linkage and euclidean distance metric.

7 DBSCAN

DBSCAN stands for density based spatial clustering of applications with noise. In DBSCAN we are required to set two parameters min samples and eps. To

understand this algorithm we need to understand these terminologies. Core sample are points that are within a dense region. Min samples are minimum number of points required to for a core sample. Eps specifies how close points should be to each other to be considered a part of a cluster. It means that if the distance between two points is lower or equal to this value, these points are considered neighbors. The parameter estimation is a problem for every data mining task. To choose good parameters we need to understand how they are used and have at least a basic previous knowledge about the dataset that will be used. Some highlights of this algorithm and how it works in shown below:

- Don't have to explicitly tell number of clusters before hand like in K-Means and hierarchical clustering. But we need to estimate and give parameters min samples and eps depending on the dataset
- The eps should be chosen based on the distance of dataset we can use a k-distance graph to find it. But in general small eps values are preferable.
- It is necessary to standard scale the data as in all clustering algorithm to remove dominance by any one feature.
- Min samples can be derived from a number of dimensions in the dataset, as $\geq$ dimensions + 1.
- Core samples that are closer to each other than distance eps are put into the same cluster.
- Points which dont before to a core sample are identified are outliers and given a label '-1'.

8 Effectiveness of clustering

As compared to supervised learning where we have the ground truth to evaluate our model's performance, clustering analysis doesn't have a good evaluation metric that can evaluate the outcome of different clustering algorithms. Hence we need a evaluate metric to find the effectiveness of the clustering. Also since most of the clustering algorithm require a number of cluster before hand there is no right answer in terms of the number of clusters that we should have in any problem. Sometimes domain knowledge and heuristic method like Elbow can be used but still we need a good way to do performance analysis.

Silhouette analysis can be used to determine the degree of separation between each cluster. We can do this in the following way:

- Compute the average distance from all the data points in the same cluster (a_i).
- Compute the average distance from all data points in the closest cluster (b_i).

- Compute the coefficient:

$$\frac{b_i - a_i}{max(a_i, b_i)}$$

- The coefficient can take values in the interval [-1,1].
- If it is 0: the sample is very close to the neighboring cluster.
- If it is 1: the sample is far away from the neighboring cluster which is good.
- If it is -1: the sample is assigned to wrong cluster which is bad.

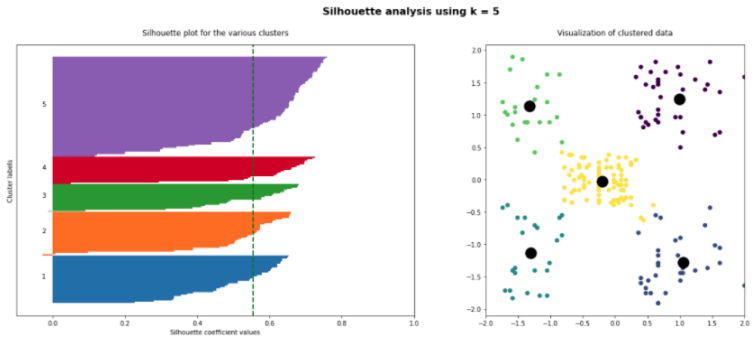


Figure 6: Silhouette analysis done for shopping list data

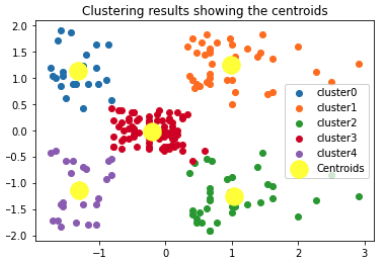
The bottom line for clustering to be good is to have average silhouette score more than 0.5 and having very less samples with -1 values. Above figure shows a good average silhouette score. Moreover the thickness of the silhouette plot gives an indication of how big each cluster is. We will use this analysis to compare the clustering techniques.

9 Results

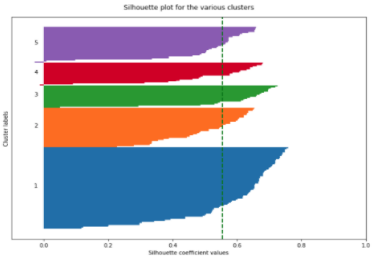
To illustrate the effect of clustering on different datasets and to understand and visualize the concept I have performed following experiments.

9.1 Experiment 1

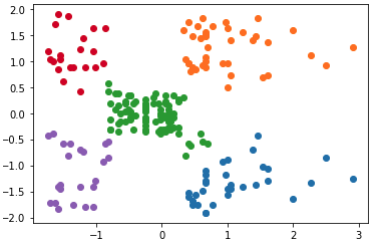
In first experiment we use the shopping data for the purpose of visualizing the clustering. We also do Silhouette Analysis of all the algorithms and compare them with each other.



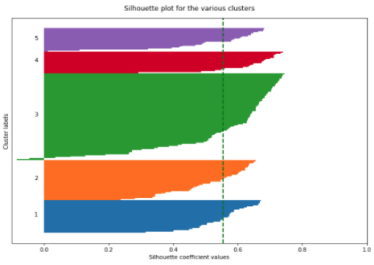
(a) KMeans with initialization KMean++ and K=5



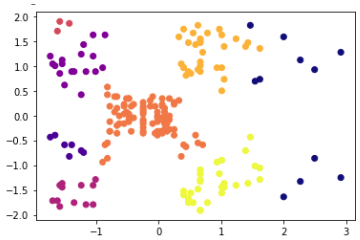
(b) Silhouette score



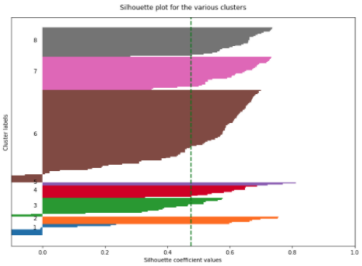
(c) Hierarchical clustering with K=5



(d) Silhouette score



(e) DBSCAN with eps=0.35 and min samples=3



(f) Silhouette score

Figure 7: Results of the experiment 1

Observations

- K-Means and Hierarchical clustering shows the quite same results and perform better than DBSCAN. This is due to the fact that this data is more distance sensitive than density.
- I observed Hierarchical clustering performs better than K-Means because sometimes initialization of centroids in K-Means forces the clustering to be caught in local optimum.

9.2 Experiment 2

In this experiment we try to use the wine quality dataset to check the accuracy of K-Means clustering algorithm. This is basically a classification datset as we have quality labels assigned to each sample already. But classification problems can converted in clustering problems by removing the labels and perform clustering on the data points. This will help to check the prediction accuracy of our K-Means algorithm. The classes are ordered and not balanced (e.g. there are many more normal wines than excellent or poor ones) as shown in figure 8.

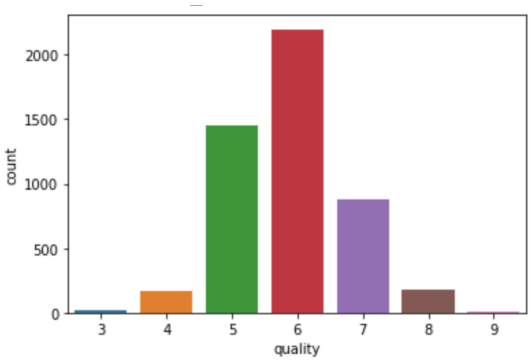


Figure 8: Number of wine samples in each label

To solve this problem we have divided the classes into three major categories good wine2(≥ 7), bad wine0(≤ 4) and otherwise normal1.

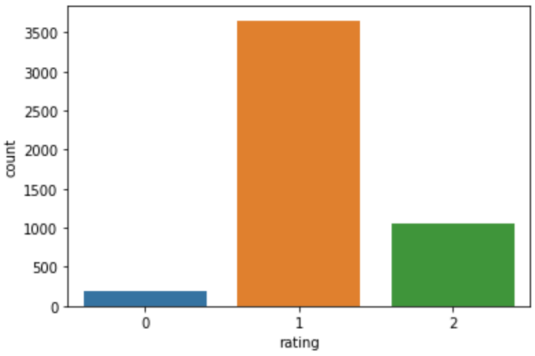


Figure 9: Distribution of good, normal and bad wines

Then we used our implementation of K-Means clustering algorithm to check its accuracy in this transformed three class clustering problem. Average accuracy on 5 trials: 73.79%

9.3 Experiment 3

This experiment aims at doing the run time analysis of our K-Means algorithm on large dataset. We are using the Birch dataset which have 100,000 samples. It will easier to plot it also as it is 2D data. We also check running time of Hierarchical and DBSCAN algorithm on this.

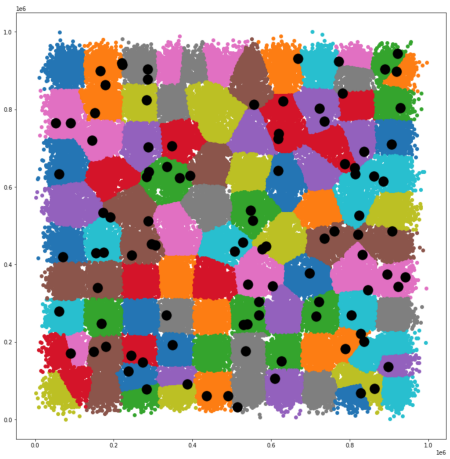


Figure 10: Figure showing the Brich dataset with data points colored according to final clustering. The black dots show initial centroids position using Kmeans++ method

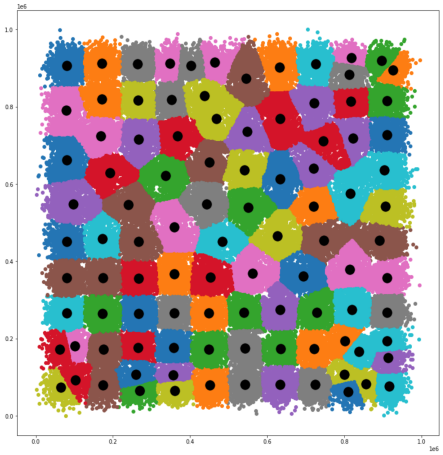


Figure 11: Figure showing the Brich dataset with points colored according to KMeans clustering(K=100). Now the black dots shows final positions of the centroids

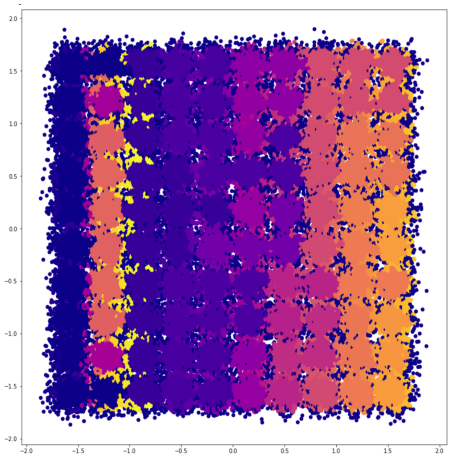


Figure 12: Figure showing the Brich dataset with points colored according to DBSCAN clustering. Clusters formed 97 with eps=0.015 and min samples=3

Clustering	Run time
K-Means	887.11 secs
DBSCAN	1.89 secs
Hierarchical	Crashed after using all available RAM

Observations

- K-Means and DBSCAN works well on large dataset.
- DBSCAN performances the fastest and hierarchical clustering failed due lack of RAM on my system hence shows hierarchical is not memory efficient.
- K-Means gives the best clustering but is time consuming as compared to DBSCAN.

10 Conclusion

Finally I would like to conclude my findings on K-means algorithm along with summarising the clustering algorithms. Advantages of K-means:

- Relatively simple to implement.
- Scales to large data sets.
- Guarantees convergence.
- Generalizes to clusters of different shapes and sizes, such as elliptical clusters.

Disadvantages of K-means:

- Choosing K manually.
- Being dependent on initialization of centroids.
- Clustering data of varying sizes and density.
- Outliers effects the clustering results so have to preprocess to remove outliers.
- Curse of dimensional can really effect K-means. I didn't explore this as dataset used in my experiment were of low dimensions. Application of PCA or spectral clustering can solve this.

Finally the choice of clustering algorithm depends upon dataset and application we are using it for. Every clustering have its advantages and disadvantages whether it is centroid based clustering like K-means or density based clustering like DBSCAN or Hierarchical clustering. Dataset where distance between points, large size of dataset and elliptical or sphere shape clusters are required K-means performance is best. If data is sensitive to density more, DBSCAN is the best choice and also this allows arbitrary shaped distribution. Also other advantage is that this algorithm do not assign outliers to clusters. Hierarchical clustering creates a tree of clusters well suited for hierarchical data such as taxonomies. In addition, another advantage is that any number of clusters can be chosen by cutting the tree at the right level.

There are many more clustering algorithms like MeanShift, spectral clustering, affinity propagation etc which have similar advantages and disadvantages. I haven't showed their working and performed experiments on them as far as this paper is considered. Though I have also implemented a distribution-based clustering known as Gaussian mixture model in my notebook. It shares its backbone with K-means the E-step and M-step we talked about with some modifications. The advantage of GMM over K-means is that it is soft clustering as compared to hard clustering in K-means. Each cluster is a Gaussian model and any point belonging to a cluster depends on probability distribution rather than just the distance metrics. I already explored a lot on this while working on newsgroup dataset using EM algorithm for GMM in my assignments so wont be speaking much about it here.

The implementation of this work can be found as notebook on my Github here.

11 References

- "An Efficient k-Means Clustering Algorithm: Analysis and Implementation" by Tapas Kanungo, David M. Mount, Member, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu.
- towardsdatascience.com.

- machinelearningmastry.com.
- scikit-learn.org.