Introduction

Clustering is a Machine Learning technique that involves the grouping of data points. Given a set of data points, we can use a clustering algorithm to classify each data point into a specific group. In theory, data points that are in the same group should have similar properties and/or features, while data points in different groups should have highly dissimilar properties and/or features. Clustering is a method of unsupervised learning and is a common technique for statistical data analysis used in many fields.

The k-Means algorithm is the most used clustering algorithm in machine learning. It is a type of unsupervised learning, which is used when you have unlabeled data with unsupervised learning, Given an unlabelled data we want to group our data into coherent subsets of clusters.

Imagine you have a set of numerical data of cancer tumors in 4 different stages from 1 to 4, and you need to study all the tumors in each stage. However, you have no idea how to identify the tumors that are at the same stage because nobody had the time to label the entire set of features (most data in the world are unlabeled). In this case, you need k-Means algorithm because it works on unlabeled numerical data and it will automatically and quickly group them together into 4 clusters.

For this example, we choose k=4 because, we already know, we have 4 tumor's stages, but if we want to cluster them based on their structure, growth speed, or growth type, then maybe k will be different than 4.

If you don't know how many groups you want, it's problematical, because K-means needs a specific number k of clusters in order to use it. So, whenever, you have to optimize and solve a problem, you should know your data and on what basis you want to group them. Then, you will be able to determine the number of clusters you need.

What is k-Means?

k-Means is a method of determining groups of related things in a list. A more apt way of saying this is that k-Means is used in determining clusters.

'k-Means' should not be confused with another technique called *'kNN'* which stands for *'k Nearest Neighbours'*, as kNN performs a different function when compared to k-Means.

How does k-Means Work?

It works by computing a set of 'k' values (this values are called means) that would be used to group each items to its respective cluster. So, 'k' can be any whole number and it represents the number of groups or clusters that we are looking for in the list of items.

k-Means Clustering Algorithm:

- Choose a value of k, number of clusters to be formed
- Randomly select k data points from the data set as the initial cluster centeroids/centers
- For each datapoint:
  a. Compute the distance between the datapoint and the cluster centroid
  b. Assign the datapoint to the closest centroid

- For each cluster calculate the new mean based on the data points in the cluster
- Repeat 3 & 4 steps until mean of the clusters stops changing or maximum number of iterations reached

k-Means flow chart

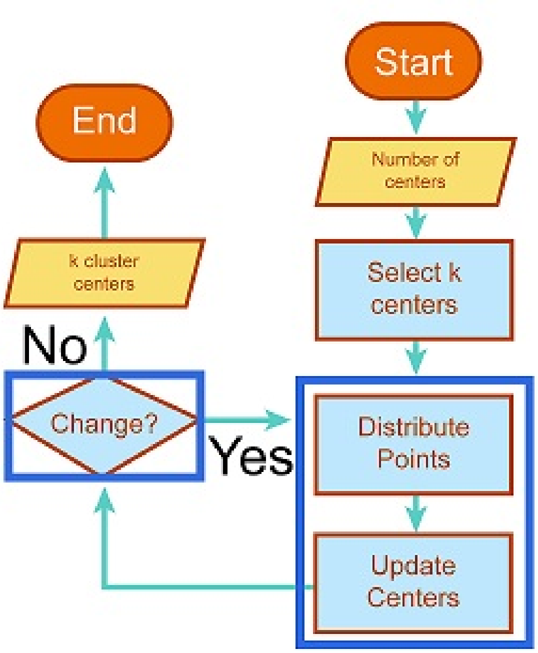

Figure 1

How many clusters?

Selecting a proper value of 'k' is very difficult until we have a good knowledge about our data set.

Therefore we need some method to determine and validate weather we are using the right number of clusters. The fundamental aim of partitioning a data set is minimizing the intra-cluster variation or SSE(Sum of Squared Errors). SSE can be calculated by first taking the difference between each data point with its centroid and then add up all the squares of the differences calculated.

So to find optimal number of clusters, run k-Means for different values of 'k'. For example, k varying from 1 to 10 and for each value of k compute SSE.
Let us plot a line chart k values on x axis and its corresponding values of SSE on y axis as shown below.

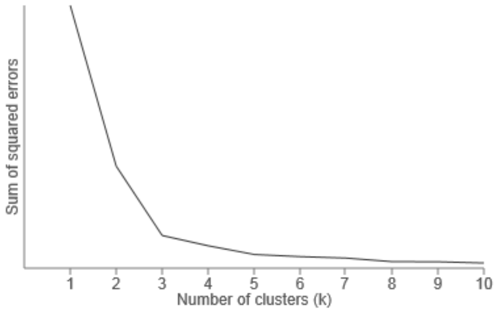

Figure 2

SSE=0 if k=number of clusters, which means that each data point has its own cluster.

As we can see in the graph there is a rapid drop in SSE as we move from k=2 to 3 and it becomes almost constant as the value of k is further increased.

Because of the sudden drop we see an elbow in the graph. So the value to be considered for k is 3. This method is known as elbow method.

There are also many other techniques which are used to determine the value of k.

k-Means is the 'go-to' clustering algorithm because it is fast and easy to understand.

How to Calculate k-Means?

Lets assume we have a set of 10 students, where their scores were [1, 2, 3, 3, 4, 5, 6, 7, 7, 9]. The problem here is how to distribute the students into four categories i.e Distinction, Credit, Pass and Fail.

Lets solve our problem of classifying a list of students grade into either distinction, credit, pass or fail using k-Means:

1. Identify the number of clusters you need - 'k's value

In our case here, we need four clusters (distinction, credit, pass and fail). So, our 'k' is equal to 4.

2. Select 'k' arbitrary points within the range of the items in the list.

The next thing we do is to select 4 arbitrary points from the cluster. From this point on, I'll refer to these points as centers. So for our lists of scores, we select [1, 4, 5, 9] as our 4 centers. Note that the selected points need not be evenly distributed, they can be picked at random.

3. Calculate distances of all items to each 'k' centers.

For each of the centers, we calculate the distance of all points to it.

This would be give a result of:

0, 1, 4, 4, 9, 6, 25, 36, 36, 64 for the first center [1]

9, 4, 1, 1, 0, 1, 4, 9, 9, 25 for the second center [4]

16, 9, 4, 4, 1, 0, 1, 4, 4, 16 for the the third center [5] and

64, 49, 36, 36, 25, 16, 9, 4, 4, 0 for the fourth center [9]

Note that the distance used is the sum of square difference of each points to the centers. This is done by squaring the difference between a point and the center. e.g for the second grade [2] and the first center [1], the distance is $(2-1)^2 = 1^2 = 1$.

4. Classify each items to a center with the shortest distance

So from the results of the distances above we can see that the first Item is closed to center [1], the second item is also closer to center [1], but the third item is closer to center [4].

So based on these distance results, we can classify each grades in the list to the centers as shown below:

- center [1] - [1, 2]
- center [4] - [3, 3, 4]
- center [5] - [5, 6, 7, 7]
- center [9] - [9]

For reference, here are the initial grades: [1, 2, 3, 3, 4, 5, 6, 7, 7, 9].

So there you have it, that is all it takes to calculate the ‘k’ clusters using ‘k’ means.

We know that we initialize cluster centroid randomly but it is possible that k Means can end up converging to different solutions depending on how the clusters are initialized.

To conclude, the main limits of the k-Means algorithm are to be found on the assumptions that it makes: Does all your data have the same variance? Are they sharing a similar and spherical distribution? Of course not. Some signals are left behind and some noise is selected. The more dimensions you have the more severe these limits will manifest themselves in your model.

In addition, your k-Means algorithm results depend on where the centroid starts. As the centroids start in different places, they might also end up in different centers. That means the same k-Means algorithm, re-run, might give a different answer! To solve this issue you should run it several times.

References:

For more details on k-Means Clustering:
https://en.wikipedia.org/wiki/K-means_clustering
https://www.datascience.com/blog/k-means-clustering
https://www.geeksforgeeks.org/k-means-clustering-introduction/