

K-Means Report

Donal Loitam

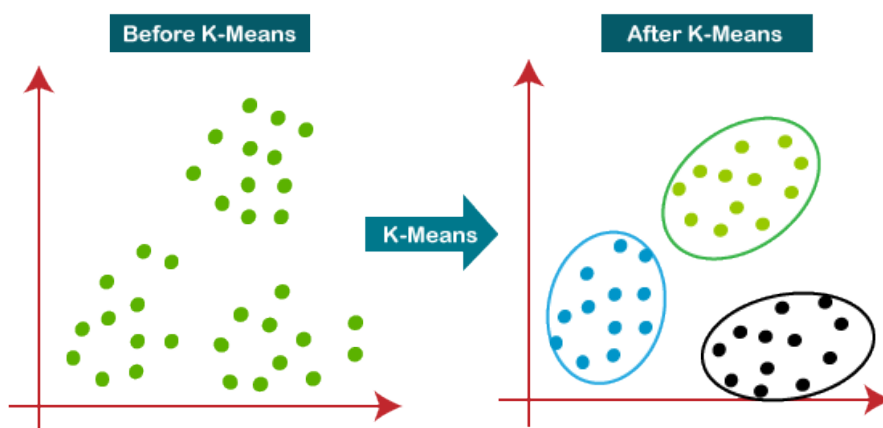
July 19, 2022

Contents

1 Introduction	1
2 Key Points of the Algorithm	2
3 Some Questions	5

1 Introduction

- K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters.
- The number of clusters found from data by the method is denoted by the letter 'K' in K-means.
- The 'means' in the K-means refers to averaging of the data; that is, finding the centroid.
- the K-means algorithm identifies k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible.



2 Key Points of the Algorithm

Let's take a simple example to understand this algorithm. So imagine you have a set of numerical data of cancer tumors in malignant or benign. However, you have no idea how to identify which tumor is what 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 2 clusters.

NOTE : For this example, we chose $k=2$ because, we already know it can either be malignant/benign. But what if we don't actually know what the value of 'k' is ? There is a solution for it

Steps of the algorithm

Step 1: Initialization

The first thing k-means does, is randomly choose K examples (data points) from the dataset or any random K points (the blue and orange points) as initial centroids and that's simply because it does not know yet where the center of each cluster is. (a centroid is the center of a cluster).

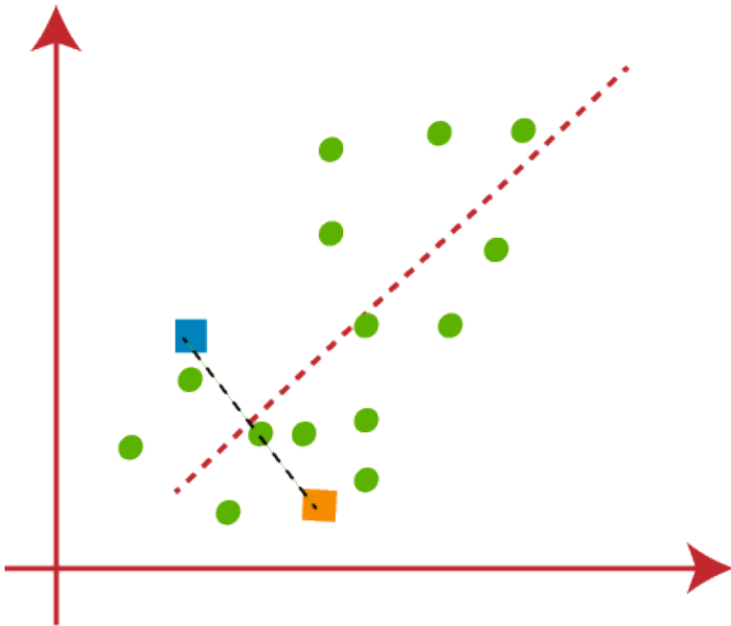


Figure 1: Initialization of centroids

Step 2: Cluster Assignment

Then, all the data points that are the closest (similar) to a centroid will create a cluster. If we're using the Euclidean distance between data points and every centroid, a straight line is drawn between two centroids, then a perpendicular bisector (boundary line) divides this line into two clusters.

From the above image, it is clear that points left side of the line is near to the K1 or blue centroid, and points to the right of the line are close to the yellow centroid. Let's color them as blue and yellow for clear visualization.

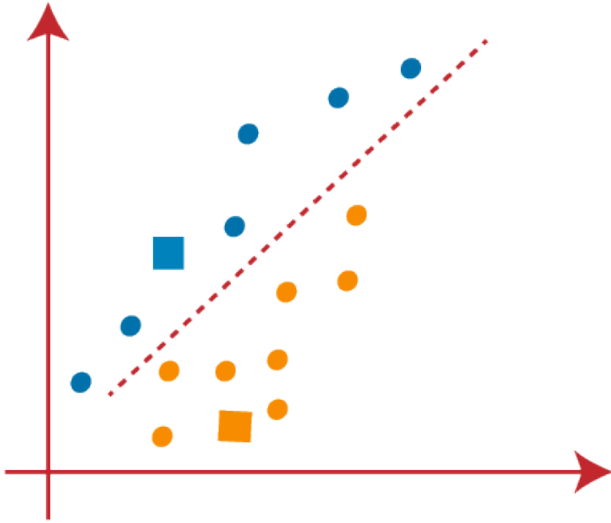


Figure 2: Cluster assignment

Step 3: Move the centroid

Now, we have new clusters, that need centers. A centroid's new value is going to be the mean of all the examples in a cluster. We'll keep repeating step 2 and 3 until the centroids stop moving, in other words, K-means algorithm is converged.

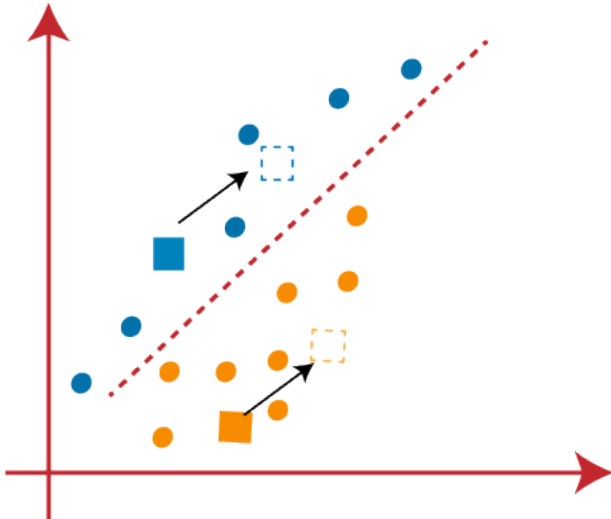


Figure 3: Changing Centroids

Here is the k-means algorithm as a psuedo code :

```

randomly chose k examples as initial centroids
while true:
    create k clusters by assigning each
        example to closest centroid
    compute k new centroids by averaging
        examples in each cluster
    if centroids don't change:
        break

```

Figure 4: Pseudo code

Unlucky Centroids

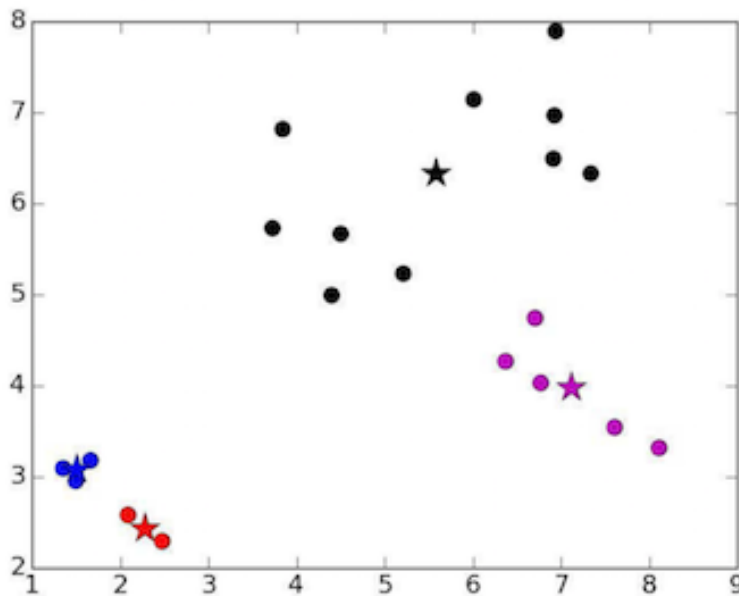


Figure 5: The blue and red stars are unlucky centroids :(

Choosing poorly the random initial centroids will take longer to converge or get stuck on local optima which may result in bad clustering. in the picture above, the blue and red stars are unlucky centroids. There are two solutions:

- Distribute them over the space.
- Try different sets of random centroids, and choose the best set.

3 Some Questions

1. Is Feature Scaling required for the K means Algorithm?

Ans

2. Why is the plot of the within-cluster sum of squares error (inertia) vs K in K means clustering algorithm elbow-shaped? Discuss if there exists any other possibility for the same with proper explanation.

Ans.

3. What are the challenges associated with K means Clustering?

Ans

4. What are the ways to avoid the problem of initialization sensitivity in the K means Algorithm?

Ans.

5. How to decide the optimal number of K in the K means Algorithm?

Ans

6. How does Naïve Bayes treat numerical and categorical values?

Ans