# Clustering

Clustering is one of the most common exploratory data analysis technique used to get an intuition about the structure of the data. It can defined as the task of identifying subgroups in the data such that data points in the same subgroup (cluster) are very similar while data points in different clusters are very different. In other words, we try to find homogeneous subgroups within the data such that data points in each cluster are as similar as possible according to a similarity measure such as Euclidean-based distance or correlation-based distance. The decision of which similarity measure to use is application-specific.

Clustering analysis can done based on features where we try to find subgroups of samples based on features or based on samples where we try to find subgroups of features based on samples. Clustering used in market segmentation; where we try to fined customers that are similar to each other whether in terms of behaviors or attributes, image segmentation/compression; where we try to group similar regions together, document clustering based on topics, etc.

Unlike supervised learning, clustering is considered an unsupervised learning method since we do not have the ground truth to compare the output of the clustering algorithm to the true labels to evaluate its performance. We only want to try to investigate the structure of the data by grouping the data points into distinct subgroups.

# K-Means Algorithm

K-Means clustering is one of the simplest and popular unsupervised machine learning algorithms. K-Means algorithm is an iterative algorithm that tries to partition the dataset into K pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group. It tries to make the inter-cluster data points as similar as possible while also keeping the clusters as different (far) as possible. It assigns data points to a cluster such that the sum of the squared distance between the data points and the cluster’s centroid (arithmetic mean of all the data points that belong to that cluster) is at the minimum. The less variation we have within clusters, the more homogeneous (similar) the data points are within the same cluster.

**The way K-Means algorithm works is as follows:**

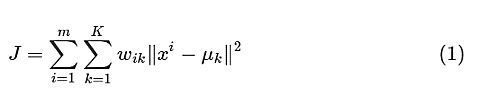
1. Specify number of clusters K.
2. Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement. This involves following steps

* Compute the sum of the squared distance between data points and all centroids.
* Assign each data point to the closest cluster (centroid).

1. Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.
2. Keep iterating step 3 & 4 until there is no change to the centroids. i.e assignment of data points to clusters isn’t changing.

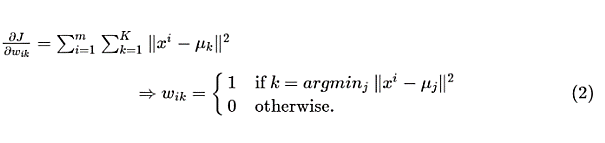
The approach kmeans follows to solve the problem is called Expectation-Maximization. The E-step is assigning the data points to the closest cluster. The M-step is computing the centroid of each cluster. Below is a break down of how we can solve it mathematically (feel free to skip it).

The objective function is:

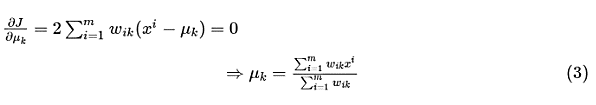


where wik=1 for data point xi if it belongs to cluster k; otherwise, wik=0. Also, μk is the centroid of xi’s cluster.

It’s a minimization problem of two parts. We first minimize J w.r.t. wik and treat μk fixed. Then we minimize J w.r.t. μk and treat wik fixed. Technically speaking, we differentiate J w.r.t. wik first and update cluster assignments. Then we differentiate J w.r.t. μk and recompute the centroids after the cluster assignments from previous step. Therefore,



In other words, assign the data point xi to the closest cluster judged by its sum of squared distance from cluster’s centroid.



Which translates to recomputing the centroid of each cluster to reflect the new assignments.

Few things to note here:

* Since clustering algorithms including K-Means use distance-based measurements to determine the similarity between data points, it’s recommended to standardize the data to have a mean of zero and a standard deviation of one since almost always the features in any dataset would have different units of measurements such as age vs income.
* Given K-Means iterative nature and the random initialization of centroids at the start of the algorithm, different initializations may lead to different clusters since K-Means algorithm may stuck in a local optimum and may not converge to global optimum. Therefore, it’s recommended to run the algorithm using different initializations of centroids and pick the results of the run that that yielded the lower sum of squared distance.
* Assignment of examples isn’t changing is the same thing as no change in within-cluster variation:



## ****Drawback of standard K-means algorithm:****

One disadvantage of the K-means algorithm is that it is sensitive to the initialization of the centroids or the mean points. Therefore, if a centroid is initialized to be a “far-off” point, it might just end up with no points associated with it and at the same time more than one clusters might end up linked with a single centroid. Similarly, more than one centroids might be initialized into the same cluster resulting in poor clustering.   
A poor initialization of centroids resulted in poor clustering.

|  |  |
| --- | --- |
|  |  |

## K-mean++

To overcome the above-mentioned drawback we use K-means++. This algorithm ensures a smarter initialization of the centroids and improves the quality of the clustering. Apart from initialization, the rest of the algorithm is the same as the standard K-means algorithm. That is K-means++ is the standard K-means algorithm coupled with a smarter initialization of the centroids.

### Steps involved

* Randomly select the first centroid from the data points.
* For each data, point compute its distance from the nearest, previously chosen centroid.
* Select the next centroid from the data points such that the probability of choosing a point as centroid is directly proportional to its distance from the nearest, previously chosen centroid. (i.e. the point having maximum distance from the nearest centroid is most likely to be selected next as a centroid)
* Repeat steps 2 and 3 until k centroids have been sampled

### Intuition

By following the above procedure for initialization, we pick up centroids, which are far away from one another. This increases the chances of initially picking up centroids that lie in different clusters. In addition, since centroids are picked up from the data points, each centroid has some data points associated with it at the end.

## Evaluation Methods

Contrary to supervised learning where we have the ground truth to evaluate the model’s performance, clustering analysis doesn’t have a solid evaluation metric that we can use to evaluate the outcome of different clustering algorithms. Moreover, since kmeans requires k as an input and doesn’t learn it from data, there is no right answer in terms of the number of clusters that we should have in any problem. Sometimes domain knowledge and intuition may help but usually that is not the case. In the cluster-predict methodology, we can evaluate how well the models are performing based on different K clusters since clusters are used in the downstream modeling.

* Elbow method
* Silhouette analysis

### Elbow Method

Elbow method gives us an idea on what a good k number of clusters would be based on the sum of squared distance (SSE) between data points and their assigned clusters’ centroids. We pick k at the spot where SSE starts to flatten out and forming an elbow. We’ll use the geyser dataset and evaluate SSE for different values of k and see where the curve might form an elbow and flatten out.

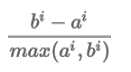
|  |  |
| --- | --- |
|  |  |

The graph above shows that k=2 is not a bad choice. Sometimes it’s still hard to figure out a good number of clusters to use because the curve is monotonically decreasing and may not show any elbow or has an obvious point where the curve starts flattening out.

### Silhouette Analysis

Silhouette analysis can used to determine the degree of separation between clusters. For each sample:

* Compute the average distance from all data points in the same cluster (ai).
* Compute the average distance from all data points in the closest cluster (bi).
* Compute the coefficient:



The coefficient can take values in the interval [-1, 1].

* If it is 0 –> the sample is very close to the neighboring clusters.
* It it is 1 –> the sample is far away from the neighboring clusters.
* It it is -1 –> the sample is assigned to the wrong clusters.

Therefore, we want the coefficients to be as big as possible and close to 1 to have a good clusters. We’ll use here geyser dataset again because it’s cheaper to run the silhouette analysis and it is actually obvious that there is most likely only two groups of data points.

### Takeaways

* Scale/standardize the data when applying K-Means algorithm.
* Elbow method in selecting number of clusters does not usually work because the error function is monotonically decreasing for all k’s.
* K-Means gives more weight to the bigger clusters.
* K-Means assumes spherical shapes of clusters (with radius equal to the distance between the centroid and the furthest data point) and does not work well when clusters are in different shapes such as elliptical clusters.
* If there is overlapping between clusters, K-Means doesn’t have an intrinsic measure for uncertainty for the examples belong to the overlapping region in order to determine for which cluster to assign each data point.
* K-Means may still cluster the data even if it cannot clustered such as data that comes from uniform distributions.