[Approach] K-Means, Kernel K-Keans, K-Medians, K-Medoids Algorithms

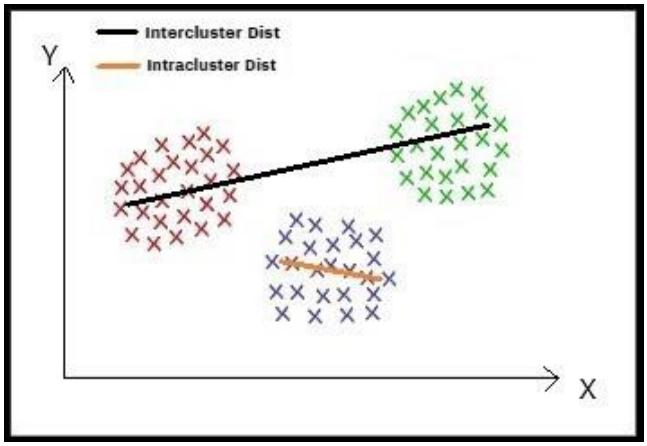
Chen Xi, xi12@illinois.edu

Overview

K-means, Kernel K-Means, K-Medians and K-Medoids are various clustering techniques. Clustering is an unsupervised machine learning algorithm, in which the available input data does not have a labeled response. The goal of clustering is to cluster/group data points, such that the data points in the same cluster are more similar to each other and dissimilar to the data points in other clusters, i.e.

1. Data points in the same cluster are closer to each other.
2. Data points in the different clusters are far apart.

Clustering is evaluated based on the intercluster (distance between two points in different clusters) and intracluster distance (distance between two points in the same cluster). The method is considered best when it achieves maximum intercluster distance and minimum intracluster distance. For example, the following dataset forms 3 clusters after clustering:



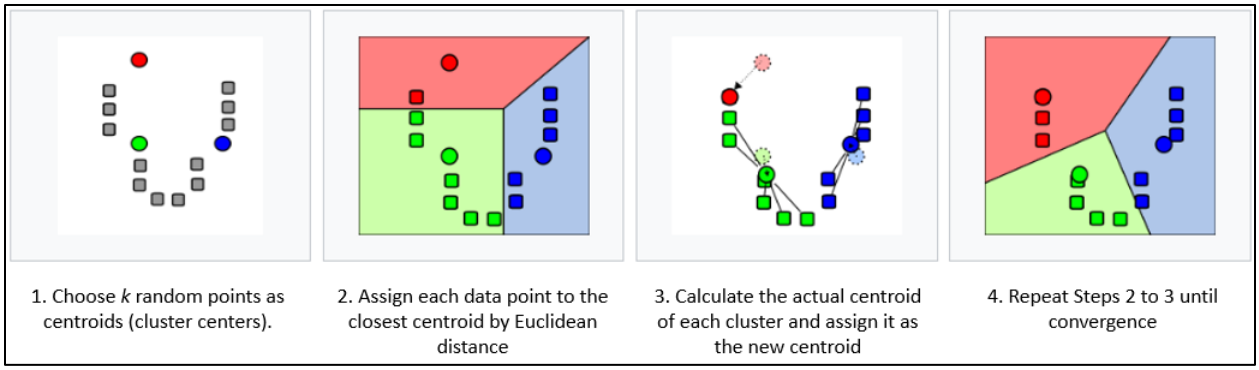
Distance measure determines the similarity between two elements and influences the shape of clusters. There are various kinds of distance measures, such as:

1. Euclidean distance measure
2. Manhattan distance measure
3. Squared Euclidean distance measure
4. Cosine distance measure

K-Means

K-Means is a centroid-based clustering techniques, which means it separates data points based on multiple centroids/centers in the data clusters the dataset into *k* clusters, with each cluster defined by a centroid (cluster center). A centroid is the average of all points in the cluster. It begins with k random points, then iteratively assigning data points with the closest Euclidean distance to the centroids into each cluster and recalculating the centroids, until a convergence is met.

Steps of K-Means Clustering



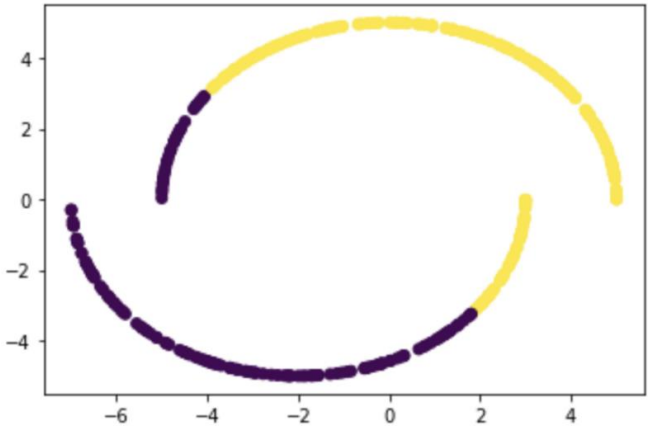
Limitations of K-means Clustering

K-Means may not perform well when (not limited to):

1. Clusters are non-spherical
2. Clusters have different sizes
3. Data has outliers (consider K-Median which is less sensitive to outliers)
4. Clusters are non-linearly separable (handled by Kernel K-means method)
5. Clusters have overlap
6. Cluster centroids have poor initialization

Kernel K-Means

K-Means performs best when clusters are spherical, dense, and linearly separable. Kernel K-means is an extension of K-Means to address K-Means' limitations of clustering non-linearly separable data/clusters, i.e. non-linear boundaries, like the one below:

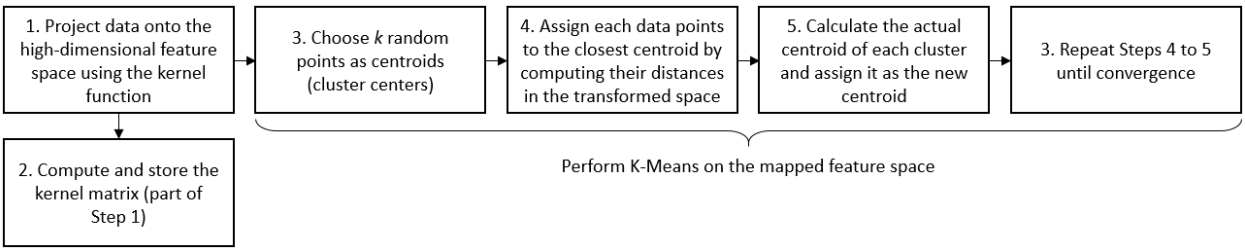


The idea to cluster such data is to embed/project the data points onto a higher dimensional feature space via a nonlinear mapping, to ensure linear separability of the data. However, directly applying K-Means and computing the data coordinates in a high-dimensional space is computationally expensive. As such, a kernel function is applied to solve the K-Means problem more efficiently in the kernel space instead, hence the name "Kernel K-Means".

A kernel function corresponds to an inner product of vectors in a certain space – it can be thought of as a similarity function over pairs of data points in this space. With this "kernel trick", we only need to

compute the inner products between the images all pairs of data in the feature space, without explicitly computing the coordinates of the data in the higher-dimensional space.

Steps of Kernel K-Means



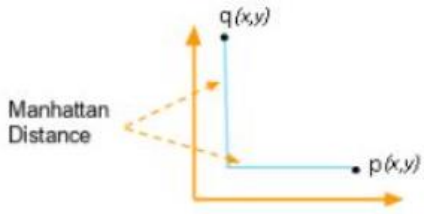
Limitations of Kernel K-Means

1. Can only be used when the projection has an associated kernel function (not all projections have an associated function).
2. The NxN kernel matrix needs to be computed, which can be computationally expensive if N (number of data points) is large.
3. Algorithm is complex in nature and time complexity is large. It can be very difficult if the data is projected to a very high dimension.

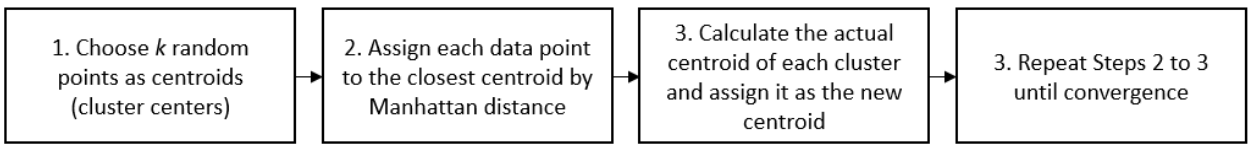
K-Medians

K-Medians is a variation of K-Means, where the median is used to determine the centroid of the cluster, instead of the mean. This has the effect of minimizing error over all clusters with respect to the 1-norm distance metric, as opposed to K-Means' squared 2-norm distance metric. Instead of Euclidean distance, the Manhattan distance measure is used, which is the sum of the horizontal and vertical components or the distance between two points measured along axes at right angles.

$$\mathbf{d} = \sum_{i=1}^{n} |q_x - p_x| + |q_x - p_y|$$



Steps of K-Medians



The K-Medians algorithm shifts the cluster centroid to the position of the vector whose elements are equal to the median value of each dimension of all of the instances assigned to the cluster. The centroids may not be actual points in the dataset. K-Medians alleviates the sensitivity of K-Means to outliers by using the Manhattan distance measure because the outliers are only contributing with their actual distance to the center, instead of the square of the distance.

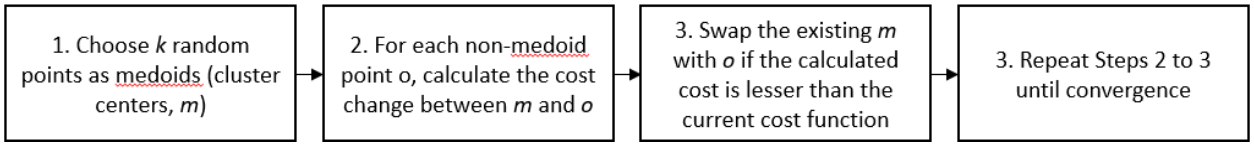
Limitations of K-Medians

In cases where outliers are desirable, K-Medoids may not separate the outliers into a separate cluster, which will impact the analysis. It's important to choose the most suitable technique based on the requirements.

K-Medoids

Medoids are representative objects of a dataset or a cluster within a dataset whose sum of distances to other objects in the cluster is minimal. Medoid has to be an actual instance from the dataset, and this is the key difference between K-Medoids and the above methods. K-Means form clusters based on the distance of observations to each centroid, while K-Medoid forms clusters based on the distance to medoids. K-Medoids has mitigated the sensitivity to outliers by not relying on centroids.

Steps of K-Medoids



Limitations of K-Medoids

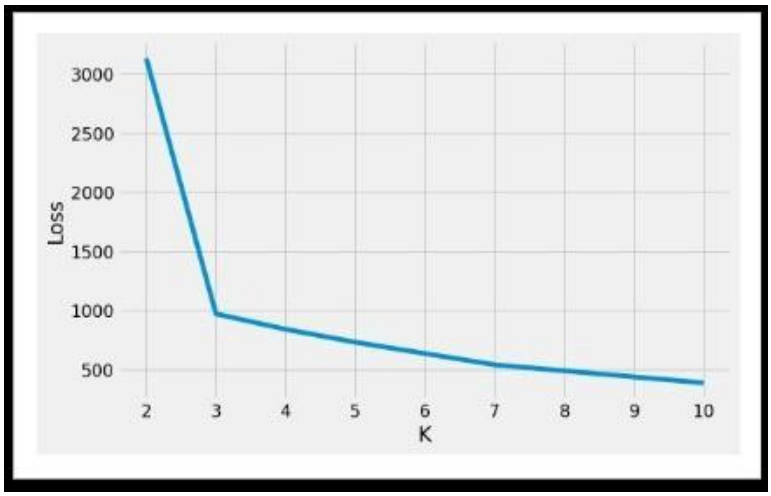
Similar to K-Medians, in cases where outliers are desirable, K-Medoids may not separate the outliers into a separate cluster, which will impact the analysis.

Selecting K

The best value of K can be computed using the Elbow method. The cost functions of the above clustering techniques are to minimize intercluster distance and maximize intracluster distance. This can be achieved by minimizing the loss function:

$$loss = argmin \sum_{i=1}^{k} \sum_{x \epsilon S_i} ||x - C_i||^2$$

To determine the right "k", we can plot loss vs k. For the below plot, it is observed that as "k" increases, loss decreases, but the increment after k=3 is plateauing. We can pick k=3 as our "k" value.



References

https://towardsdatascience.com/understanding-k-means-k-means-and-k-medoids-clustering-algorithms-ad9c9fbf47ca

https://www.simplilearn.com/tutorials/machine-learning-tutorial/k-means-clustering-algorithm#how_does_kmeans_clustering_work

https://medium.com/udemy-engineering/understanding-k-means-clustering-and-kernel-methods-afad4eec3c11

https://machinelearningjourney.com/index.php/2020/02/07/k-means-k-medians/

https://towardsdatascience.com/use-this-clustering-method-if-you-have-many-outliers-5c99b4cd380d

https://medium.com/@ali.soleymani.co/beyond-scikit-learn-is-it-time-to-retire-k-means-and-use-this-method-instead-b8eb9ca9079a

Image Reference: https://towardsdatascience.com/understanding-k-means-k-means-and-k-medoids-clustering-algorithms-ad9c9fbf47ca

Image reference: https://en.wikipedia.org/wiki/K-means_clustering