## K-means clustering with SciPy

The K-Means clustering is one of the partitioning approaches and each cluster will be represented with a calculated centroid. All the data points in the cluster will have a minimum distance from the computed centroid.

**Scipy** is an open-source library that can be used for complex computations. It is mostly used with NumPy arrays. It can be installed by running the command given below.

*pip install scipy*

It has dedicated packages for the process of clustering. There are two modules that can offer clustering methods.

1. cluster.vq
2. cluster.hierarchy

**cluster.vq**

This module gives the feature of **vector quantization**to use with the K-Means clustering method. The quantization of vectors plays a major role in reducing the distortion and improving the accuracy. Mostly the distortion here is calculated using the Euclidean distance between the centroid and each vector. Based on this the vector od data points are assigned to a cluster.

**cluster.hierarchy**

This module provides methods for general hierarchical clustering and its types such as agglomerative clustering. It has various routines that can be used for applying statistical methods on the hierarchies, visualizing the clusters, plotting the clusters, checking linkages in the clusters, and also checking whether two different hierarchies are equivalent.

In this article, cluster.vq module will be used to carry out the K-Means clustering.

**K-Means clustering with Scipy library**

The K-means clustering can be done on given data by executing the following steps.

1. Normalize the data points.
2. Compute the centroids (referred to as code and the 2D array of centroids is referred to as code book).
3. Form clusters and assign the data points (referred to as mapping from code book).

**cluster.vq.whiten()**

This method is used to normalize the data points. Normalization is very important when the attributes considered are of different units. For example, if the length is given in meters and breadth is given in inches, it may produce an unequal variance for the vectors. It is always preferred to have unit variance while performing K-Means clustering to get accurate clusters. Thus, the data array has to pass to whiten() method before any other steps.

***cluster.vq.whiten(input\_array, check\_finite)***

*Parameters:*

1. *input\_array : The array of data points to be normalized.*
2. *check\_finite : If set to true, checks whether the input matrix contains only finite numbers. If set to false, ignores checking.*

**cluster.vq.kmeans()**

This vq module has two methods namely kmeans() and kmeans2().

The **kmeans()** method uses a threshold value which on becoming less than or equal to the change in distortion in the last iteration, the algorithm terminates. This method returns the centroids calculated and the mean value of the Euclidean distances between the observations and the centroids.

***cluster.vq.kmeans(input\_array, k, iterations, threshold, check\_finite)***

*Parameters:*

1. *input\_array : The array of data points to be normalized.*
2. *k : No.of.clusters (centroids)*
3. *iterations : No.of.iterations to perform kmeans so that distortion is minimized. If k is specified it is ignored.*
4. *threshold : An integer value which if becomes less than or equal to change in distortion in last iteration, the algorithm terminates.*
5. *check\_finite : If set to true, checks whether the input matrix contains only finite numbers. If set to false, ignores checking.*

The **kmeans2()**method does not use the threshold value to check for convergence. It has more parameters that decide the method of initialization of centroids, a method to handle empty clusters, and validating whether the input matrices contain only finite numbers. This method returns centroids and the clusters to which the vector belongs.

***cluster.vq.kmeans2(input\_array, k, iterations, threshold, minit, missing, check\_finite)***

*Parameters:*

1. *input\_array : The array of data points to be normalized.*
2. *k : No.of.clusters (centroids)*
3. *iterations : No.of.iterations to perform kmeans so that distortion is minimized. If k is specified it is ignored.*
4. *threshold : An integer value which if becomes less than or equal to change in distortion in last iteration, the algorithm terminates.*
5. *minit : A string which denotes the initialization method of the centroids. Possible values are****‘random’, ‘points’, ‘++’, ‘matrix’****.*
6. *missing : A string which denotes action upon empty clusters. Possible values are****‘warn’, ‘raise’****.*
7. *check\_finite : If set to true, checks whether the input matrix contains only finite numbers. If set to false, ignores checking.*

**cluster.vq.vq()**

This method maps the observations to appropriate centroids which are calculated by the kmeans() method. It requires the input matrices to be normalized. It takes the normalized inputs and generated code-book as input. It returns the index in the code-book to which the observation corresponds to and the distance between the observation and its code (centroid).