K-means Clustering

Jennifer Yoon, March 12, 2109, “ mail@JenniferYoon.com “ to contact me.

# Main Concept

Used in non-supervised machine learning. Seeks to understand or interpret the relationship among main clusters of input data. Does not make predictions, such as a y-axis predicted income value or in which group a new data sample will belong to.

* There is an **Initial Stage** where some method is used to create the initial condition.
	+ All input data are randomly assigned to k-groups.
	+ Or, k-numbers of inputs are randomly assigned to be the initial k-centers.
	+ Other options are available.
* The **CENTROID** of each group is re-calculated for each step.
	+ This is the centers of each initial groups.
	+ Conceptually similar to the mean/mode.
	+ Uses 2-D geometry to draw the shortest line between pairs of points.
* **Re-Classification** of all inputs to their closest **CENTROID** containing group.
	+ A given point that was in group 1 before, but is closer to the 2nd groups’ centroid now, is re-classified into the 2nd group. Next, if that same point that is in group 2, is even closer to the 3rd centroid, then it is moved again to the 3rd group.
* The steps of CENTROID calculation and Re-Classification of inputs is repeated.
* Stops repeating when some **Termination** criteria is reached.
	+ No more movement in inputs among groups.
	+ Or some other stopping point, especially for “soft” clusters.

# Pros & Cons

Pros are generally that it is easy to understand and fast to implement. Cons are that the final clusters are sensitive to initially assigned random groups. Or has a high **variance**. Generally the algorithm is run many times, 100 time or more, and the average of all results are used. Also, if k-number is chosen so that there is a bad match between true number of groups in the population and the number of groups used in the algorithm, it is possible to get a bad **bias** as well. That is, the clusters may not look anything like the true clusters in the population.

# Images From Book

The book uses initialization method where all data points are randomly assigned to one of the 3 k clusters. This results in all 3 of the initial **CENTROID** circles being located in the middle of the graph.



Images From Wikipedia

Alternative initialization method is to randomly assign the 3 centroids to any 3 data points. This makes it more likely that the initial centroids will be spread out on the graph.

**Demonstration of the standard algorithm**



 

1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).
2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the [Voronoi diagram](https://en.wikipedia.org/wiki/Voronoi_diagram) generated by the means.

3. The [centroid](https://en.wikipedia.org/wiki/Centroid) of each of the *k* clusters becomes the new mean.

4. Steps 2 and 3 are repeated until convergence has been reached.

**Initialization methods** -- copied from Wikipedia

Commonly used initialization methods are Forgy and Random Partition.[[9]](https://en.wikipedia.org/wiki/K-means_clustering#cite_note-hamerly4-9) The Forgy method randomly chooses *k* observations from the dataset and uses these as the initial means. The Random Partition method first randomly assigns a cluster to each observation and then proceeds to the update step, thus computing the initial mean to be the centroid of the cluster's randomly assigned points. The Forgy method tends to spread the initial means out, while Random Partition places all of them close to the center of the data set. According to Hamerly et al.,[[9]](https://en.wikipedia.org/wiki/K-means_clustering#cite_note-hamerly4-9) the Random Partition method is generally preferable for algorithms such as the *k*-harmonic means and fuzzy *k*-means. For expectation maximization and standard *k*-means algorithms, the Forgy method of initialization is preferable. A comprehensive study by Celebi et al.,[[10]](https://en.wikipedia.org/wiki/K-means_clustering#cite_note-10) however, found that popular initialization methods such as Forgy, Random Partition, and Maximin often perform poorly, whereas Bradley and Fayyad's approach[[11]](https://en.wikipedia.org/wiki/K-means_clustering#cite_note-11) performs "consistently" in "the best group" and [*k*-means++](https://en.wikipedia.org/wiki/K-means%2B%2B) performs "generally well".

From <<https://en.wikipedia.org/wiki/K-means_clustering>>