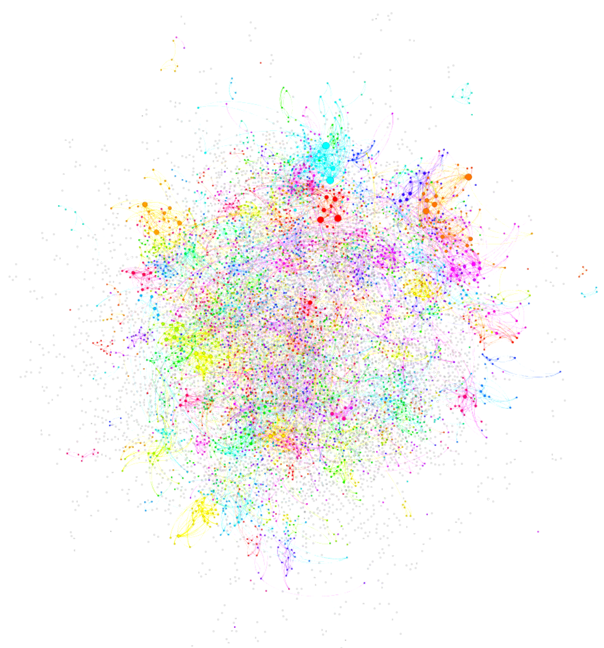
# k-Means Clustering

**K-means clustering** is a traditional, simple machine learning algorithm that is trained on a test data set and then abl
a new data set using a prime, $k$ number of clusters defined a priori.



*Data mining can produce incredible visuals and results. Here, k-means algorithm was used to assign items to 1000 clusters, each represented by a color* [1] .

An objective of machine learning is to derive techniques for unsupervised learning on data. This kind of data analysi
helpful in many applications that require classification of data, such as identifying cancerous cells within a large sam
clustering words with similar definitions for better search engine accuracy, identifying outliers in student's academic
performance for better refinement of habits, or even for detecting landmines in a battlefield [2].

EXAMPLE

Building a Baseball Team Using Classification

Imagine a high school baseball coach wants to use data analysis to predict whether potential new recruits will be spectacular, mediocre, or dismal.

He has data on the players currently on his team: position, years of experience, batting average, on-base percenta
number of stolen bases per games. He also has some data on the players he is considering recruiting. How might h
about selecting a new player that fills any gaps of skill currently on his team?

Contents

K-means algorithm
Technical Analysis
Complexity
Example
When to Use K-Means Clustering
References

## K-means algorithm

This clustering algorithm separates data into the best suited group based on the information the algorithm already h separated in $k$ different clusters, which are usually chosen to be far enough apart from each other spatially, in Eucled Distance, to be able to produce effective data mining results. Each cluster has a center, called the **centroid**, and a da clustered into a certain cluster based on how close the features are to the centroid.

A computer-generated program showing k-means clustering

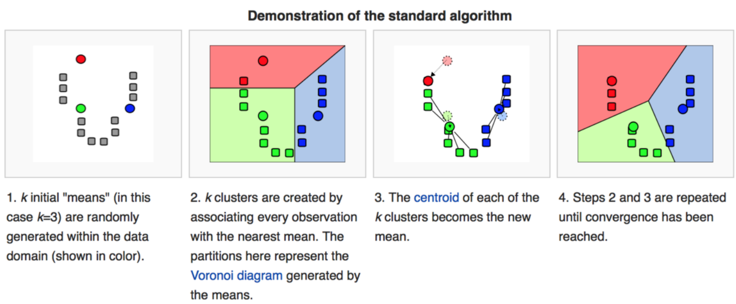
*A computer-generated program showing k-means clustering* [3]

K-means algorithm iteratively **minimizes** the distances between every data point and its centroid in order to find the optimal solution for all the data points.

1. $k$ random points of the data set are chosen to be centroids.
2. Distances between every data point and the $k$ centroids are calculated and stored.
3. Based on distance calculates, each point is assigned to the nearest cluster
4. New cluster centroid positions are updated: similar to finding a mean in the point locations
5. If the centroid locations changed, the process repeats from step 2, until the calculated new center stays the same, signals that the clusters' members and centroids are now set.

Finding the minimal distances between all the points implies that data points have been separated in order to form compact clusters possible, with the least variance within them. In other words, no other iteration could have a lower a distance between the centroids and the data points found within them.

EXAMPLE




*A graphical look at k-means clustering* [4]

## Technical Analysis

The K-means algorithm defined above aims at minimizing an objective function, which in this case is the squared err

The objective function for the K-means clustering algorithm is the squared error function:

$$J = \sum_{i=1}^{k} \sum_{j=1}^{n} (||x_i - v_j||)^2 = 1$$

where,

$||x_i - v_j||$ is the Eucledian distance between a point, $x_i$, and a centroid, $v_j$, iterated over all *k* points in the $i^{th}$ clu *n* clusters.

[5]

In simpler terms, the objective function attempts to pick centroids that minimize the distance to all points belonging respective cluster so that the centroids are more symbolic of the surrounding cluster of data points.

**Pseudocode**

```
INPUT:
      E = set of e data points
      K = number of clusters
      I = Iterations desired // this is necessary as full convergence is extremely costly.
OUTPUT:
      C = set of c cluster centroids
      L = set of distances, l from e to assigned centroid.

For c in C:
      Randomly assign centroid c to be at some e.

For e in E:
      Calculate distance from e to all centroids c.
      Assign each e to centroid c with min. distance. Store in L.

i = 0.
minDistance = Inf

While i < I:
      For c in C:
            Compute the average location of all e assigned to cluster c.
            Reassign centroid c to new location.
    For e in E:
            Calculate distance from e to all centroids c.
    If minDistance != l:
                Assign each e to centroid c with min. distance = L.
    Else:
        End
Return assignments
```

Why use the squared error function rather than, say, the absolute error? The reason is because the squared error has mathematical properties than absolute error. For further reference on these properties, check out this blog post by B

## Complexity

Although the algorithm seems quite simple, finding the optimal solution to the problem for observations in either *d* or for *k* clusters is NP-Hard. However, if *k* and *d* are fixed, the problem can be solved in time $O(n^{dk+1} log(n))$ using Algorithm, a common k-clustering algorithm, where *n* is the number of entities to be clustered.[6]. However, the runn this algorithm on nicely clustered data can be quite small, as minimization of the objective function will occur quickly

Because the algorithm computes the distances between each of the *k* cluster centers and their respective data point single iteration, a few iterations are enough to make further adjustments not worth the time complexity trade-off. In words, because further iterations won't change the assignments of the majority of data points but their distances sti be calculated, the algorithm becomes inefficient if convergence is the goal. For this reason, several variations of Lloy clustering algorithm have been developed in order to speed up the process at later stages; where these variations in the triangle-inequality, amongst others.

## Example

Before applying k-means clustering to a data set, data has to go from characteristics of an object to numerical data t analyzed.

EXAMPLE

### Categorizing Baseball Players

How would the baseball coach use k-means clustering to predict whether new recruits will be good? Each trait of a can be represented as a feature vector. By converting characteristics into numbers in a feature vector, the players b comparable in a vector space so that their differences can be better quantified.

The coach has the same type of information on both current players and new potential ones. Using k-means cluste the entire set of data points, he can figure out which of his current, known-level (remarkable, mediocre, dismal) pla closest to the new ones, and which of the new players would fill the most voids within his team.

## When to Use K-Means Clustering

K-Means clustering is a fast, robust, and simple algorithm that gives reliable results when data sets are distinct or we from each other in a linear fashion. It is best used when the number of cluster centers, is specified due to a well-defi types shown in the data. However, it is important to keep in mind that K-Means clustering may not perform well if it heavily overlapping data, if the Euclidean distance does not measure the underlying factors well, or if the data is nois outliers [7].

## References


1. Brickley, D. *1k_overview*. Retrieved from https://www.flickr.com/photos/danbri/6233990550/in/photolist-auSQcG-8 EmVCps-ptEDsu-7m1WFX-5EFMse-i4x1v-egRfqk-81wUmZ-a4D73j-87Yxiy-tty5TD-bAhHzE-5tMAuK-7MfnED-7rbgg 7rfbmm-rq7j4f-a1hTYE-gxCMpH-57XUAW-a6Nx34-8hAC1D-ounzPd-dybf7L-fCTKB3-dybeZd-dy5Mut-dybfdb-ow6 wcCF9s-cqjRDS-rrXXND-dybfjm-fCBeqB-dy5MGx-fCTM13-nouGAk-71tzae-4xwxcC-k2tddi-8kfyKH-8hABZB-7y4a ou3Zst-ftDTEu-osZ261-oun8p5-cSNeMy-4eJiRW
2. Naik, A. *Clustering Algorithm Applications*. Retrieved from https://sites.google.com/site/dataclusteringalgorithms/clustering-algorithm-applications
3. Petersen, J. *K-means*. Retrieved from http://pypr.sourceforge.net/kmeans.html#k-means-example
4. None, N. *k-means clustering*. Retrieved May 02, 2016, from https://en.wikipedia.org/wiki/K-means_clustering
5. Matteucci, M. *K-Means Clustering*. Retrieved June 14, 2016, from http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/kmeans.html
6. Inaba, M. (1994). Applications of weighted Voronoi diagrams and randomization to variance-based k-clustering. *A Symposium on Computational Geometry*, *10th*, 332-339.
7. Naik, a. *k-Means Clustering Algorithm*. Retrieved June 14,2016, from https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm


**Cite as:** k-Means Clustering. *Brilliant.org*. Retrieved 11:43, October 15, 2019, from https://brilliant.org/wiki/k-means-clustering/