***K-Means’ Theorem***

The k-means algorithm is an iterative method for clustering a set of N points (**vectors**) into k groups or clusters of points. It is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining.

k-means is one of the simplest **unsupervised learning** algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume **k clusters**) fixed apriori.

**Apriori.Algorithm:**

Apriori is an algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database.

The frequent item sets determined by Apriori can be used to determine association rules which **highlight general trends in the database**: this has applications in domains such as market basket analysis.

Apriori is designed to operate on databases containing transactions (for example, **collections of items bought by customers,** or **details of a website frequentation**). Other algorithms are designed for finding association rules in data having no transactions (Winepi and Minepi), or having no timestamps (DNA sequencing).

**Elbow Method:**

Elbow method is to run k-means clustering on the dataset for a range of values of k (say, k from 1 to 10 in the examples above), and for each value of k calculate the **sum of squared errors** (SSE)

var sse = {};

for (var k = 1; k <= maxK; ++k) {

sse[k] = 0;

clusters = kmeans(dataset, k);

clusters.forEach(function(cluster) {

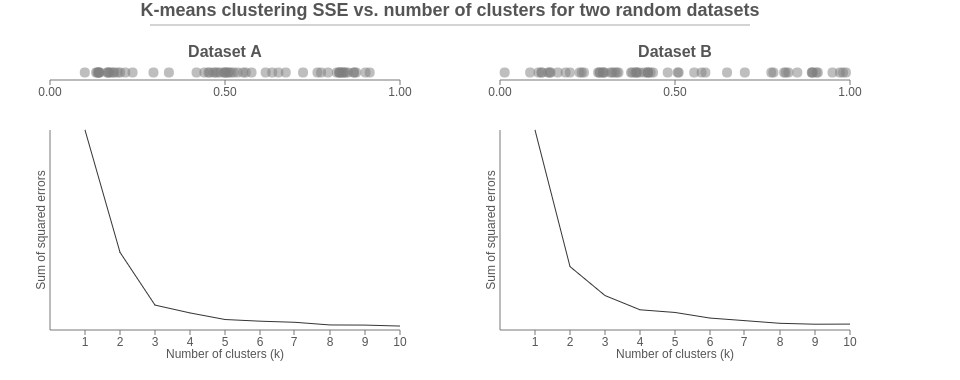
mean = clusterMean(cluster);

cluster.forEach(function(datapoint) {

sse[k] += Math.pow(datapoint - mean, 2);

});

});}

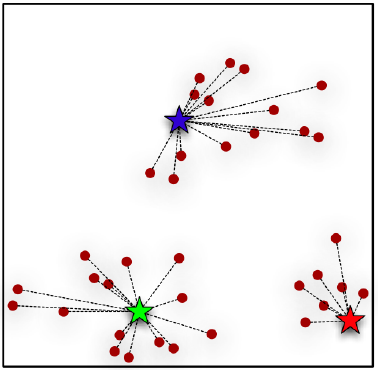
****

Plot a line chart of the SSE for each value of k. If the line chart looks like an arm, then the "**elbow**" on the arm is the value of k that is the best. The idea is that we want a small SSE, but that the SSE tends to decrease toward 0 as we increase k (the SSE is 0 when k is equal to the number of data points in the dataset, because then each data point is its own cluster, and there is no error between it and the center of its cluster). So our goal is to choose a small value of k that still has a low SSE, and the elbow usually represents where we start to have diminishing returns by increasing k.

Dataset A on the left. At the top we see a number line plotting each point in the dataset, and below we see an elbow chart showing the SSE after running k-means clustering for k going from 1 to 10. it is pretty clear elbow at **k= 3**, indicating that 3 is the best number of clusters.

**Find closest centroid**

Find the closest centroid to each point, and group points that share the same closest centroid.



Randomly initialize three points called the cluster centroids.

K-means is an iterative algorithm and it does two steps

1. Cluster assignment step

2. Move centroid step.