K-Means Clustering

Unsupervised Learning

- Many times we want to group data points together that are similar looking/behaving
  - Customer behavior
  - Music listening/movie watching behavior

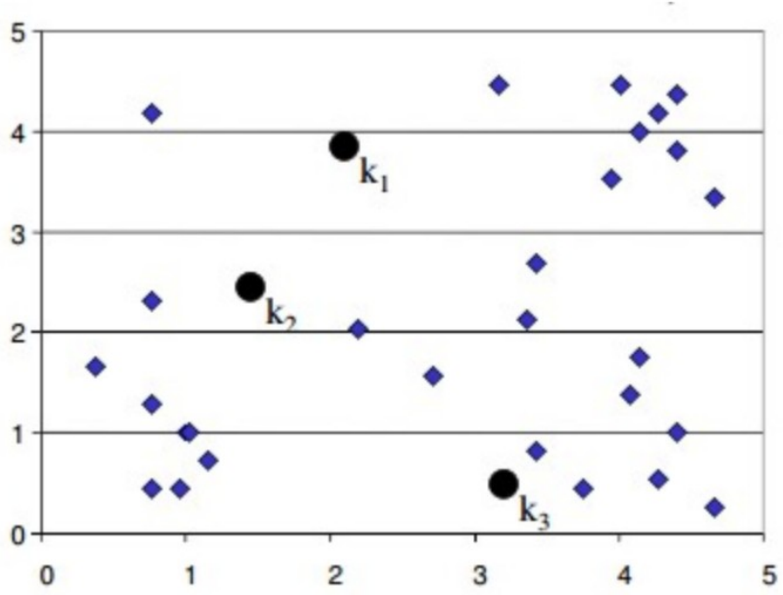
- Unsupervised learning is used when you don’t have a training set of labeled outputs (i.e., you don’t know which group each member falls into)

- Need to look for grouping or patterns in the data

K-Means Clustering: Step 1

- First, you guess how many clusters there are in the data

- Then you guess the first set of cluster centers (k_1, k_2, k_3)

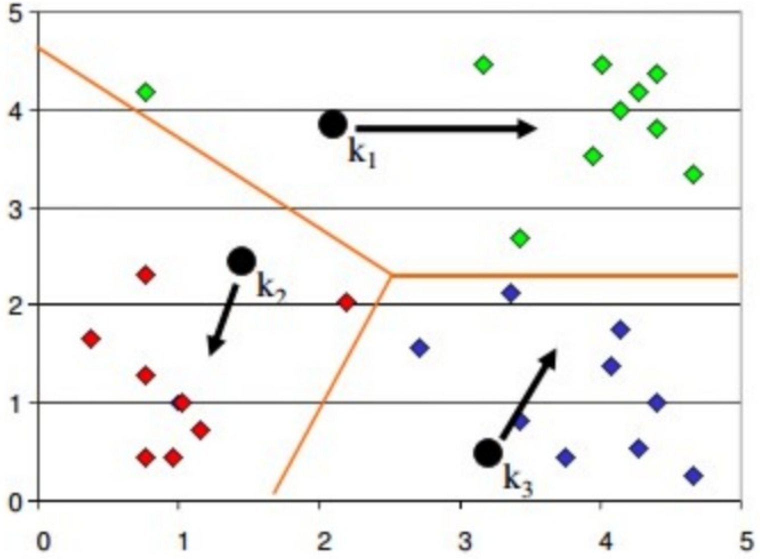


K-Means Clustering: Step 2

- Group the data points by which cluster center is closest
  - Usually use Euclidean distance

$$D = \sqrt{\left(x_2 - x_1\right)^2 + \left(y_2 - y_1\right)^2}$$

- Shift the cluster center to the mean location of the group of data points in each cluster

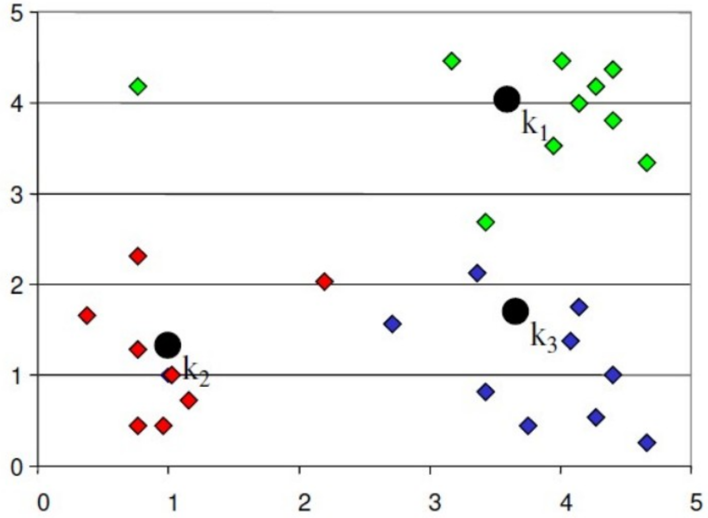


K-Means Clustering: Step 3

- New cluster centers are shown to the right after shift to mean location

- Process is repeated back to Step 2

- Clustering process stops when there is no change in membership of each cluster



Example

https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

Initial Cluster Centers

The k-means algorithm can be sensitive to the initial cluster center values. Therefore, k-means is often run multiple times with different initial cluster values and the best result is kept.

In the sklearn implementation of k-means, the n_init parameter controls the number of times the k-means algorithm will be run with different centroid seeds. By default, this value is set to 10.

Pros & Cons

Pros

- Easy to implement and understand
- Scales easily to large data sets
- Works well with spherical clusters
- Efficient

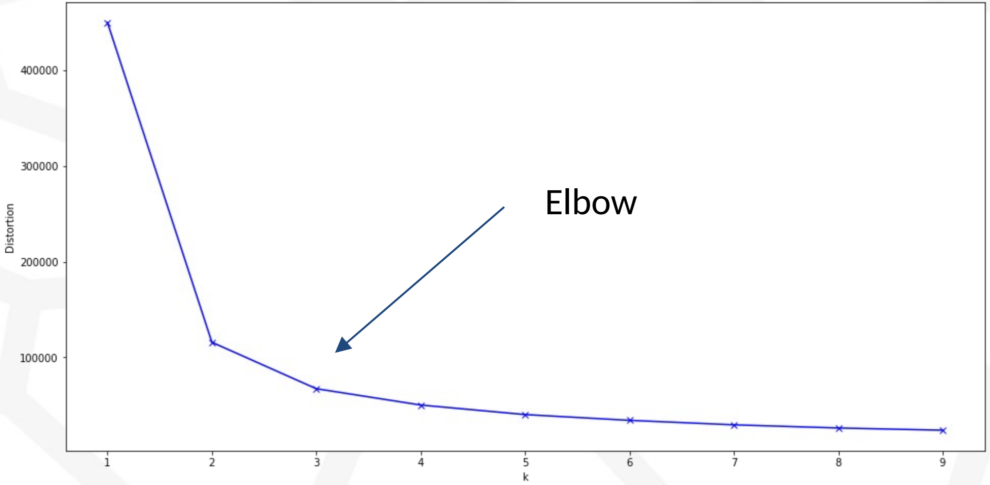
Cons

- Have to choose k manually. Results sensitive to k.
- Dependent on initial centroid values.
- Doesn’t work well with clusters of varying sizes
- Doesn’t work well with clusters of unusual shapes

Choosing K

When k is small, all points will belong to the same cluster. When k is large, all points will belong to their own clusters. Therefore, there is a “sweet spot” value of k.

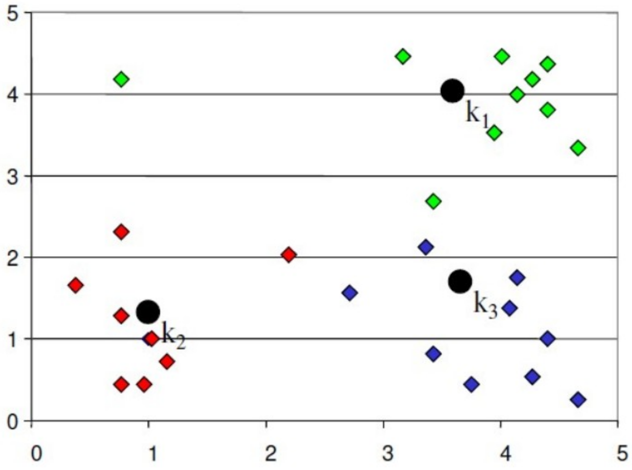
The most common way of choosing k is using the elbow method. Run k-means for a range of k values and calculate the distortion. Choose the value of k that falls in the “elbow”.


Elbow

What is Distortion?

Distortion is simply the within-cluster sum of squared error:

1. For each cluster, find the distance between each point and the centroid.
2. Square the distance and sum it.
3. Then sum the squared error across all clusters.



$$\sum_{\mathrm{P_i\ in\ Cluster\ 1}} \mathrm{distance}(\mathrm{P_i}, \mathrm{C_1})^2 + \sum_{\mathrm{P_i\ in\ Cluster\ 2}} \mathrm{distance}(\mathrm{P_i}, \mathrm{C_2})^2 + \sum_{\mathrm{P_i\ in\ Cluster\ 3}} \mathrm{distance}(\mathrm{P_i}, \mathrm{C_3})^2$$