K-Means

It is a clustering algorithm (Marker segmentation, Social network analysis, Organize Computing clusters, Astronomical data analysis)

The k-means algorithm is extremely **easy to implement** but is also **computationally very efficient** compared to other clustering algorithms, which might explain its popularity. While k-means is very good at identifying clusters of spherical shape, one of the **drawbacks of this clustering algorithm is that we have to specify the number of clusters k a prior**. An inappropriate choice for k can result in poor clustering performance. Later in this chapter, we will discuss the elbow method and silhouette plots, which are useful techniques to evaluate the quality of a clustering to help us determine the optimal number of clusters k.

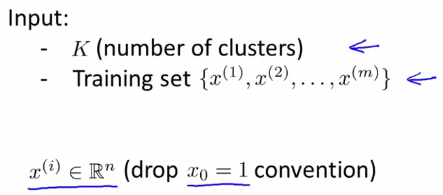
**k-means can be summarized by the following four steps:**

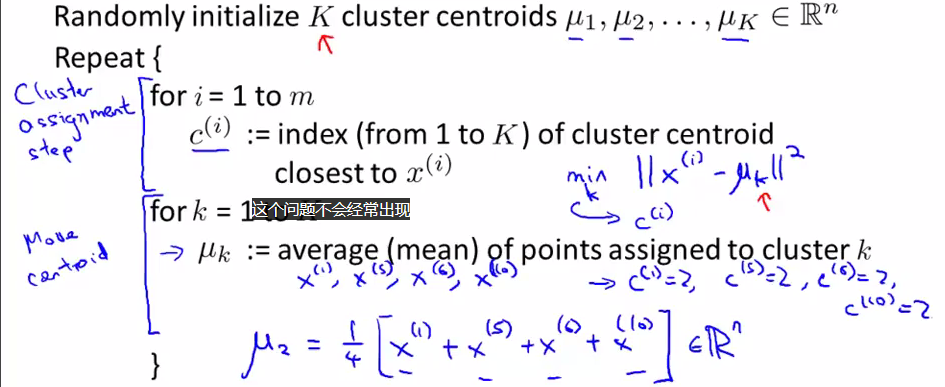
1. **Randomly pick k centroids from the sample points as initial cluster centers**
2. **Assign each sample to the nearest centroid **
3. **Move the centroids to the center of the samples that were assigned to it.**
4. **Repeat the steps b,c until the cluster assignment do not change or a user-defined tolerance or a maximum number of iterations is reached.**
5. **K-Means Algorithm**

Based on Euclidean distance metric, we can describe the k-means algorithm as a simple optimization problem, an iterative approach for minimizing the **within-cluster sum of squared errors (SSE)**, which is sometimes also called cluster intertia:

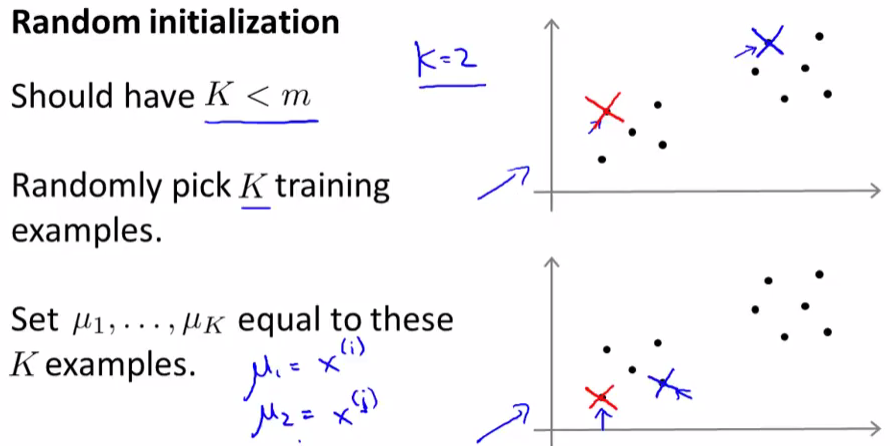


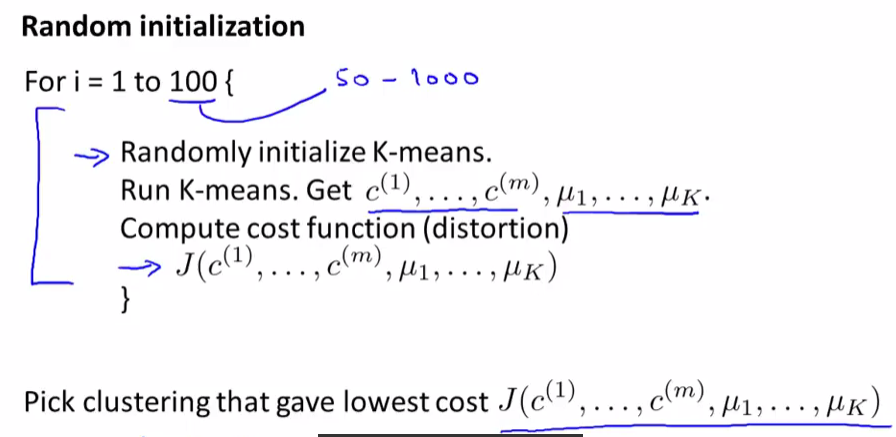






1. **Initialize K**





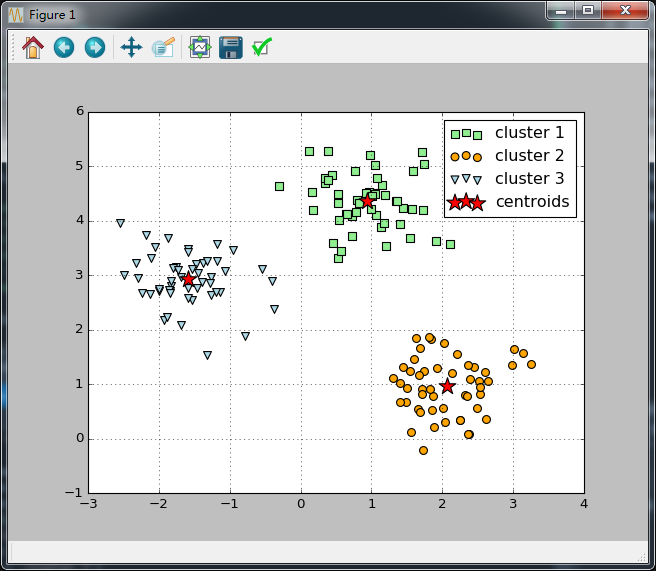
Another strategy is to place the initial centroids far away from each other via the k-means++ algorithm, which leads to better and more consistent results than the classic k-means.

The initialization in k-means++ can be summarized as follows:

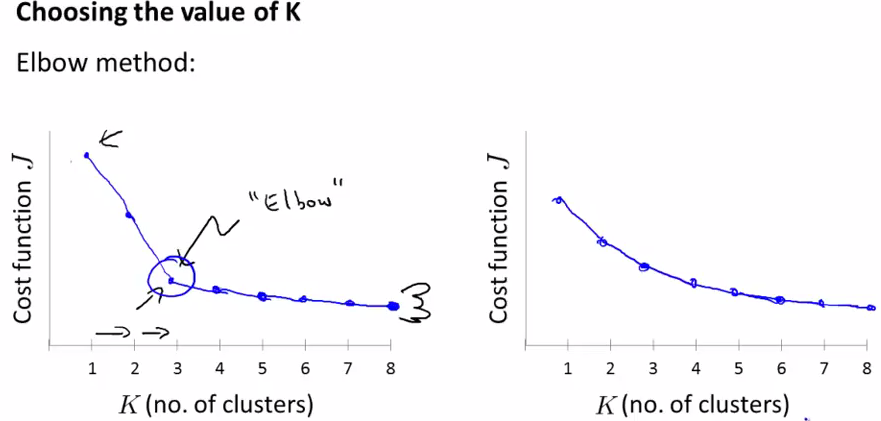
1. Initialize an empty set M to store the k centroids being selected
2. Randomly choose the first centroid  from the input samples and assign it to M
3. For each sample  that is not in M, find the minimum squared distance  to any of the centroids in M.
4. To randomly select the next centroid ,use a weighted probability distribution equal to 
5. Repeat steps 2 and 3 until k centroids are chosen.
6. Proceed with the classic k-means algorithm

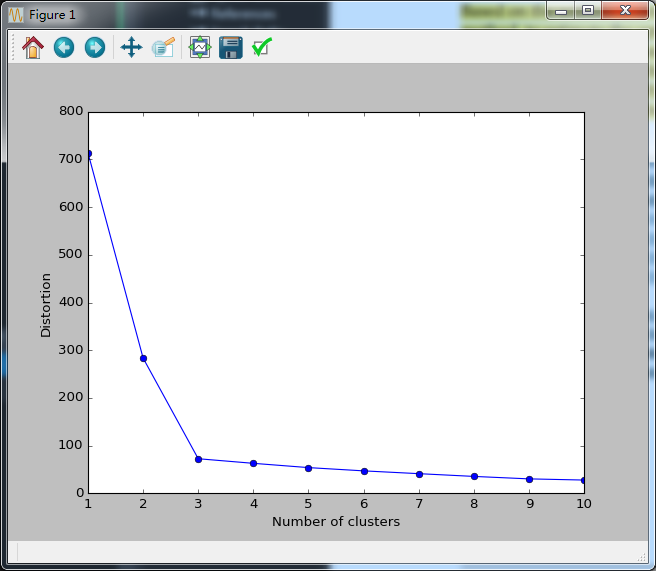
Another problem with k-means is that one or more clusters can be empty.

Solution: If a cluster is empty, the algorithm will search for the sample that is farthest away from the centroid of the empty cluster. Then it will reassign the centroid to be this farthest point.



1. **Choosing the number of Clusters**
2. Based on the within-cluster SSE, we can use a graphical too, the so-called **elbow method**, to estimate the optimal number of clusters k for a given task. Intuitively we can say that , if k increases , the distortion will decrease. This is because the samples will be closer to the centroids they are assigned to. The idea behind the elbow method is to identify the value of k where the distortion begins to increase most rapidly, which will become more clear if we plot distortion for different values of k.





1. Quantifying the quality of clustering viasilhouette plots

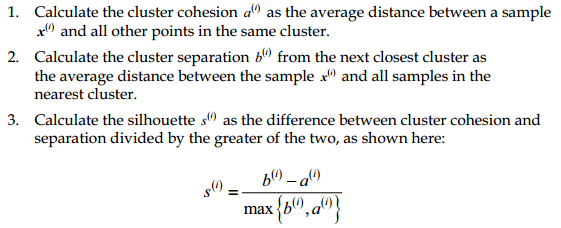
Another intrinsic metric to evaluate the quality of a clustering is silhouette analysis, which can also be applied to clustering algorithms other than k-means.

Sihouette analysis can be used as a graphical tool to plot a measure of how tightly grouped the samples in the clusters are. To calculate the silhouette coefficient of a single sample in our dataset, we can apply the following three steps:

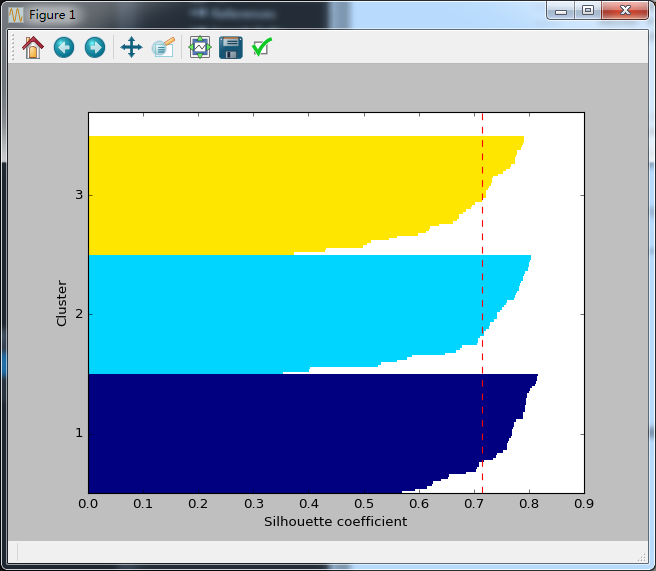
**Hard Versus soft clustering**

Hard clustering describes a family of algorithms where each sample in a dataset is assigned to exactly one cluster, as in the k-means algorithm. In contrast, algorithms for soft clustering (sometimes also called fuzzy clustering) assign a sample to one or more clusters. A popular example of soft clustering is the fuzzy C-means (FCM) algorithm (also called soft k-means or fuzzy k-means).

More about Fuzzy k-means, please refer to chapter 11 of python machine learning and Other related materials.



The silhouette coefficient is bounded in the range -1 to 1. Based on the preceding formula, we can see that the silhouette coefficient is 0 if the cluster separation and chhesion are equal. Furthermore , we get close to an ideal silhouette coefficient of 1 if b>>a, since b quantifies how dissimilar a sample is to other cluster, and a tells us how similar it is to the other samples in its own cluster ,respectively.



Refrence:

1. Machine Learning , Ng Video
2. Python Machine learning, book