Clustering

Clustering attaches label to each observation or data points in a set. It is also called as “unsupervised classification” or “segmentation” or “grouping”. Intuitively, it assigns same label to a data points that are “close” to each other. Thus, clustering algorithms rely on a distance metric between data points.

Some of the business cases for Clustering includes

- Market Segmentation
- Product Segmentation
- Customer Segmentation
- Speaker modeling

Distance

Three famous metrics (to calculate the distance between two points) are Manhattan Distance, Euclidean distance and Minkowsi distance

N-dimensional Manhattan distance:

$$d(p,q) = \sum_{i=1}^{n} |p_i - q_i|$$

3 dimensional Euclidean distance:

$$p_1 = [x_1, y_1, z_1]$$
$$p_2 = [x_2, y_2, z_2]$$
$$d(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

N dimensional Euclidean distance:

$$p = [p_1, p_2, ..., p_n]$$
$$q = [q_1, q_2, ..., q_n]$$
$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + ... + (p_n - q_n)^2}$$

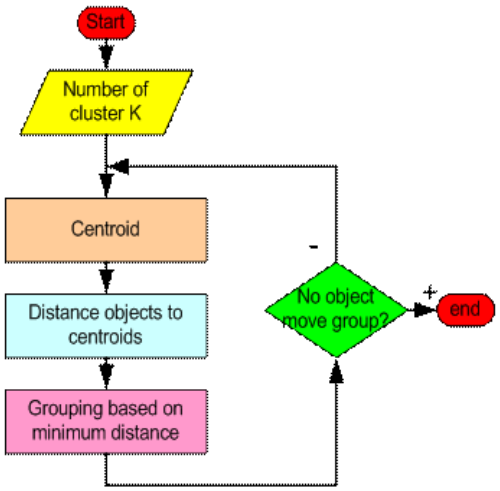
N-dimensional Minkowski distance

$$d(p,q) = (\sum_{i=1}^{n} |p_i - q_i|^c)^{1/c}$$

for c = 1 , c = 2, the Minkowski metric becomes equal to the Manhattan and Euclidean metrics respectively.

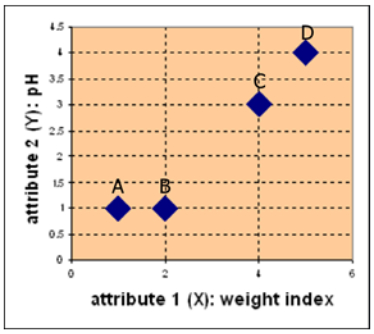
K-Means Clustering

- It is one of the most popular clustering algorithm. "*K*" stands for number of clusters, it is typically a user input to the algorithm; some criteria can be used to automatically estimate *K*
- *K*-means algorithm is iterative in nature
- Works only for numerical data
- Easy to implement

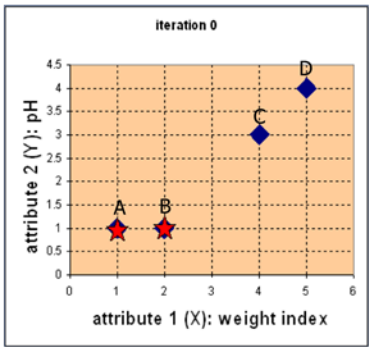


K-Means Clustering Steps

- **Step 1:** Begin with a decision on the value of k =number of clusters .
- **Step 2**: Choose random 'k' samples as initial centroids
- **Step 3:** Take each sample in sequence and compute its distance from the centroid. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster. Re-calculate the centroids of the new clusters
- **Step 4:** Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.



Suppose we have 4 types of medicines and each has two attributes (pH and weight index). Our goal is to group these objects into K=2 group of medicine

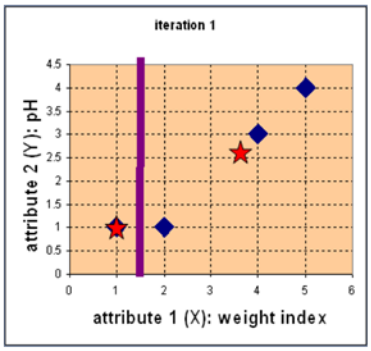


Step 1: Use initial seed points for partitioning and assign each object to the cluster with the nearest seed point

$$\mathbf{D}^0 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix} \quad \begin{matrix} \mathbf{c}_1 = (1,1) & group-1 \\ \mathbf{c}_2 = (2,1) & group-2 \end{matrix}$$

$$d(D, c_1) = \sqrt{(5-1)^2 + (4-1)^2} = 5$$

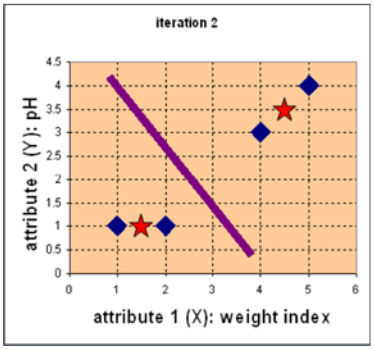
$$d(D, c_2) = \sqrt{(5-2)^2 + (4-1)^2} = 4.24$$



Step 2: Compute new centroids of the current partition and compute the distance of all objects to the new centroids

$$\mathbf{D}^1 = \begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix} \quad \begin{matrix} \mathbf{c}_1 = (1,1) & group-1 \\ \mathbf{c}_2 = (\frac{11}{3}, \frac{8}{3}) & group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \\ \begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} & \begin{matrix} X \\ Y \end{matrix} \end{matrix}$$



Step 3: Repeat the first two steps until its convergence

How to Choose K?

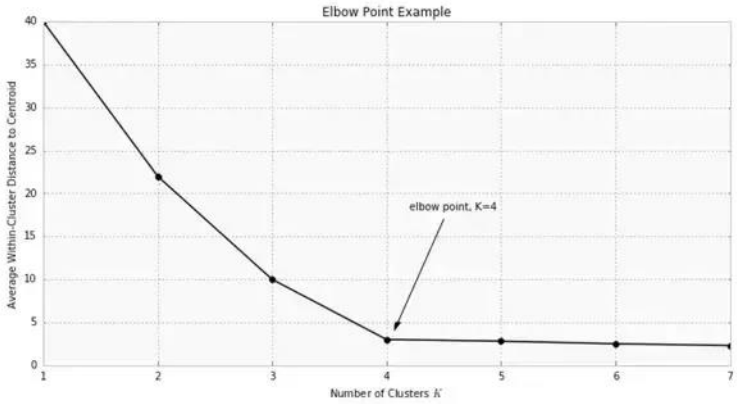
Elbow Curve of Within Cluster Distance

One method to validate the number of clusters is the elbow method. The idea of the 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)

$$\sum_{k=1}^{K} \sum_{i \in S_k} \sum_{j=1}^{p} (x_{ij} - \bar{x}_{kj})^2$$

where S_k is the set of observations in the *k*th cluster and $\bar{x}_{kj}$ is the *j*th variable of the cluster center for the *k*th cluster.

Then, 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.



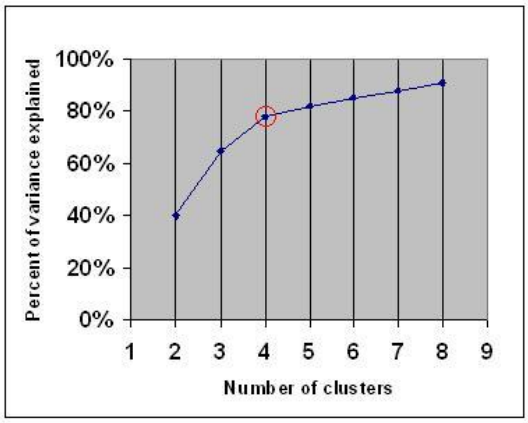
Elbow Curve of Variance Explained

It looks at the percentage of variance explained as a function of the number of clusters: One should choose a number of clusters so that adding another cluster doesn't give much better modeling of the data. More precisely, if one plots the percentage of variance explained by the clusters against the number of clusters, the first clusters will add much information (explain a lot of variance), but at some point the marginal gain will drop, giving an angle in the graph. The number of clusters is chosen at this point, hence the "elbow criterion". This "elbow" cannot always be unambiguously identified. Percentage of variance explained is the ratio of the between-group variance to the total variance, also known as an F-test.

$$Variance\ Explained = \frac{Between\ Cluster\ Sum\ of\ Square}{Total\ Square\ of\ Error}$$

$$T = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} d(x_i, x_j) \qquad T = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\Big(\sum_{C(j)=k} d(x_i, x_j) + \sum_{C(j)\neq k} d(x_i, x_j)\Big) = W(C) + B(C)$$

$$W(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(j)=k} d(x_i, x_j) \qquad B(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(j)\neq k} d(x_i, x_j)$$



Silhouette Score

The Silhouette Coefficient is calculated using the mean intra-cluster distance (a) and the mean nearest-cluster distance (b) for each sample. The Silhouette Coefficient for a sample is **(b - a) / max(a, b)** where b is the distance between a sample and the nearest cluster that the sample is not a part of.

The best value is 1 and the worst value is -1. Values near 0 indicate overlapping clusters. Negative values generally indicate that a sample has been assigned to the wrong cluster, as a different cluster is more similar