e-PGPathshala

Subject : Computer Science

Paper: Machine Learning

Module: K-Means Clustering

Module No: CS/ML/24

Quadrant I – e-text

Welcome to the e-PG Pathshala Lecture Series on Machine Learning.In this module we will discuss an important algorithm used for clustering – the k-means clustering algorithm.

Learning Objectives:

The learning objectives of this module are as follows:

- To understand k-means clustering algorithm
- To outline the different variations of k-means clustering
- To discuss the strengths and weaknesses of k-means clustering algorithm

24.1 Partitional Algorithm

In the previous module we discussed hierarchical clustering where new clusters are found iteratively using previously determined clusters. In this module we will be discussing another type of clustering called partitional clustering where we discover all the clusters at the same time. One of the most important and popular clustering algorithm – the k-means algorithm is an example of partitional clustering. In partitional clustering the data points are divided into a finite number of *partitions*which are disjoint subsets of the set of data points, that is,each data point is assigned to exactly one subset. These type of clustering algorithms can be viewed as problems of iteratively relocating data points between clusters until an optimal partition of the data points has been obtained.

In the basic algorithm the data points are partitioned into k clusters and the partitioning criterion is optimized using methods such as minimizing the squared error. In the case of the basic iterative algorithm of K-means or K-medoids both of which belong to the partitional clustering categorythe convergence is local and the globally optimal solution is not always guaranteed.

The number of data points in any data set is finite and the number of distinct partitions is also finite. It is possible to tackle the problem of local minima by using exhaustive search methods.

24.2 K-Means Clustering

As the case with any partitional algorithm, basically the K-means algorithm is an iterative algorithm whichdivides the given data set into K disjoint groups. As already discussed K-means is the most widely used clustering techniques. This partitional method uses prototypes for representing the cluster. For a given K we need to find a partition of *K clusters* that optimizes the chosen partitioning criterion or cost function

The ***K-means*** algorithm is a heuristic method where each cluster is represented by the center of the cluster and the algorithm converges when the centroids of the clusters do not change. The K-means algorithm is the simplest partitioning method for clustering and widely used in data mining applications

24.2.1 Steps of K-Means

Initialize k values of centroids

The following two steps are repeated until the data points do not change partitions and there is no change in the centroid

- Partition the data points according to the current centroids. The similarity between each data point and eachcentroidis determined and the data points are moved to the partition to which it is most similar.
- The new centroidsof the data points in each partition is then calculated.

These steps are given in the flow diagram shown in Figure 24.1.

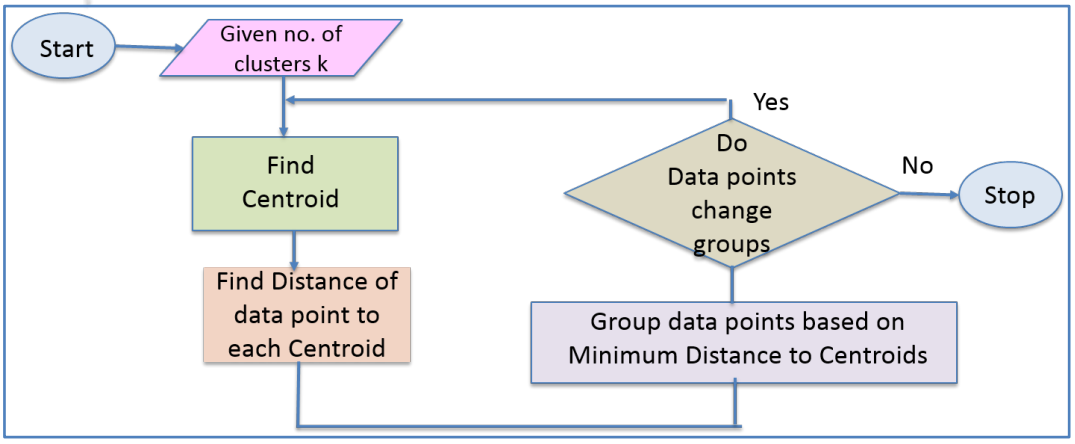


Figure 24.1 Flow Diagram of K-means algorithm

24.2.2 Algorithm for K-means Clustering

Now let us discuss in detail the K-means algorithm. Let $X = \{x_1, x_2, x_3, \ldots\ldots, x_n\}$ be the set of n data points and $V = \{v_1, v_2, \ldots\ldots, v_c\}$ be the set of centroids.

1 We first randomly select *'k'* cluster centroids.
2 We then calculate the distance between each data point and the cluster centroids.
3 We then find the cluster whose centroid is nearest to the data point. We then assign the data point to this cluster.
4 We then recalculate the new cluster centroid using:

$$J(V) = \sum_{i=1}^{k} \sum_{j=1}^{k_i} \left(\left\| x_i - v_j \right\| \right)^2$$

where, *'k_i'* represents the number of data points in i^{th} cluster.

5 We then recalculate the distance between each data point and the newly obtained cluster centroids.
6 If no data point has been reassigned then we stop, otherwise we repeat from step 3.

24.2.3 Example

In the example shown in Figure 24.2 we see that initially the cluster centroids are chosen at random as we are talking about an unsupervised learning technique where we are not provided with the labelled samples. Even after the first round using these randomly chosen centroids the complete set of data points are partitioned all at once. As you can see after round 2 the centroids have moved since the data points have been used to calculate the new centroids. The process is repeated and the centroids are relocated until convergence occurs when no data point changes partitions.

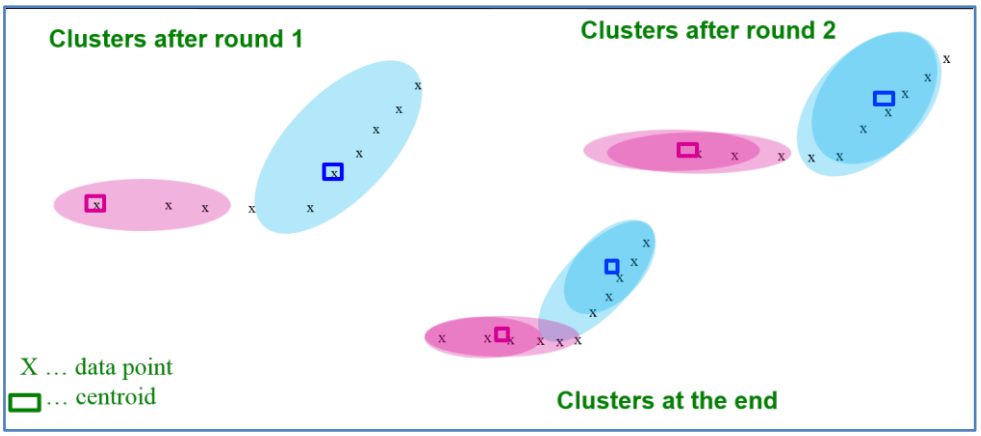


Figure 24.2 Example of the K-means Algorithm

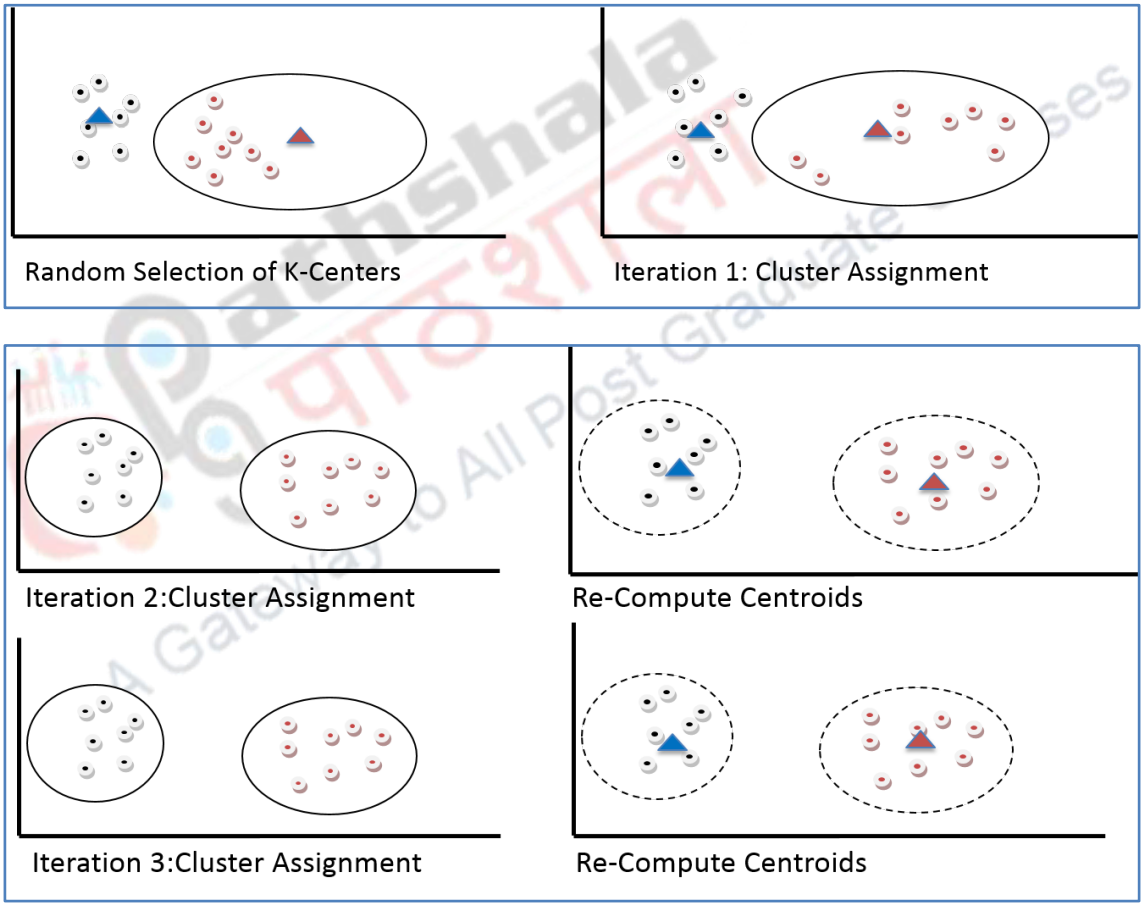


Figure 24.3 Steps of Cluster Assignment and Centroid Computation

24.3 Issues associated with the K-means Algorithm

24.3.1 Getting the K-right

The first question we need address is to choose the correct value of K that is how many partitions should we have. One way of selecting K is to first try

different values of K and studying the change in the average distance to centroid as K increases. As we can see from Figure 24.3 the average changes rapidly until the right value of K whenthe changes are slow.

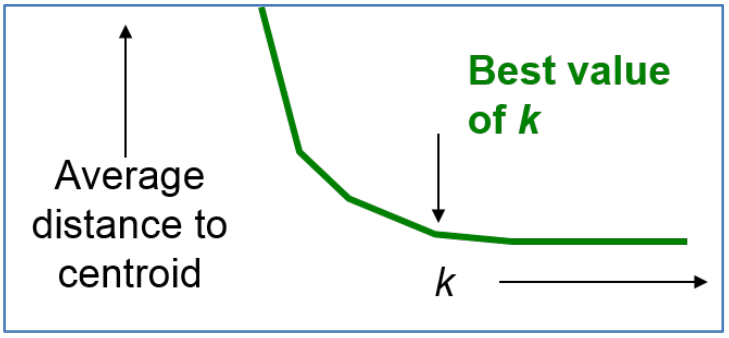


Figure 24.3 Choosing the right value of K

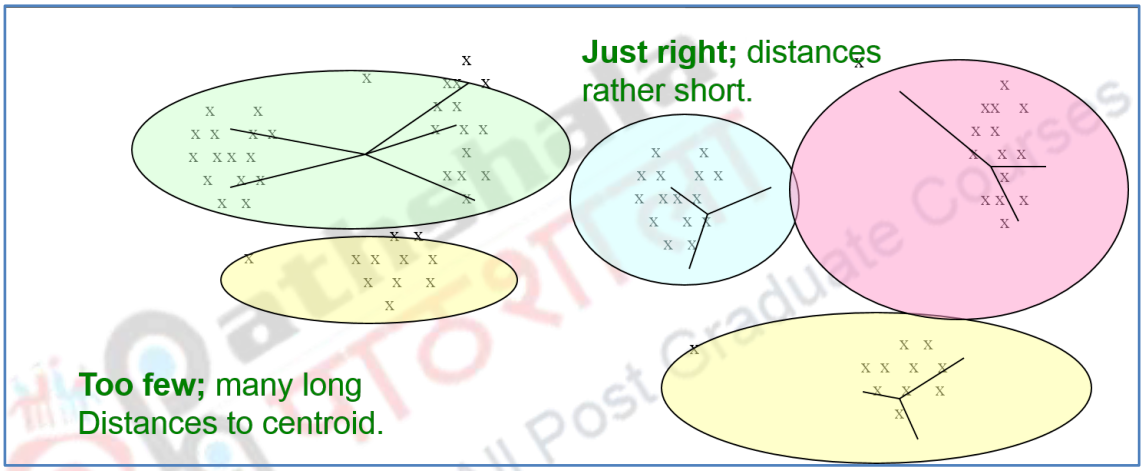


Figure 24.4 Illustration of Different Values of K

Figure 24.4 shows the case of 2 clusters which is too few since the distances to the centroids is too long while when there are 3 clusters the distances are short and just right.

24.3.2 Problems with selecting initial points

One of the issues associated with K-means clustering is the initial choice of the the K-centroids. Though we have explained that these initial centroids can be chosen at random this may not lead to fast convergence. If there are K clusters then the chance of selecting one centroid from each cluster is small and the chance becomes even lower when K is large. If we assume that clusters are of same size, n, that is each cluster has n data points then the number of ways of selecting K centroids is large and the probability that we choose the K across clusters is small.

$$P = \frac{\text{number of ways to select one centroid from each cluster}}{\text{number of ways to select Kcentroids}} = \frac{K!n^k}{(Kn)^K} = \frac{K!}{K^K}$$

For example, if K = 10, then probability = $10!/10^{10} = 0.00036$

Sometimes the randomly chosen initial centroids will readjust themselves in the right direction but there are chances that they do not (Figure 24.5).Therefore this initial cluster problem needs to be addressed. One way to do this is to conduct multiple runs but even then due to the low probability, this approach may not work.Another method we sample and use hierarchical clustering to determine the initial centroids. Another approach is first select more than K initial centroids and then select among these initial centroids which are the most widely separated. Another method is carry out pre-processing where we can normalize the data and eliminate outliers. On the other hand in post-processing we can eliminate small clusters that may represent outliers and split or merge clusters

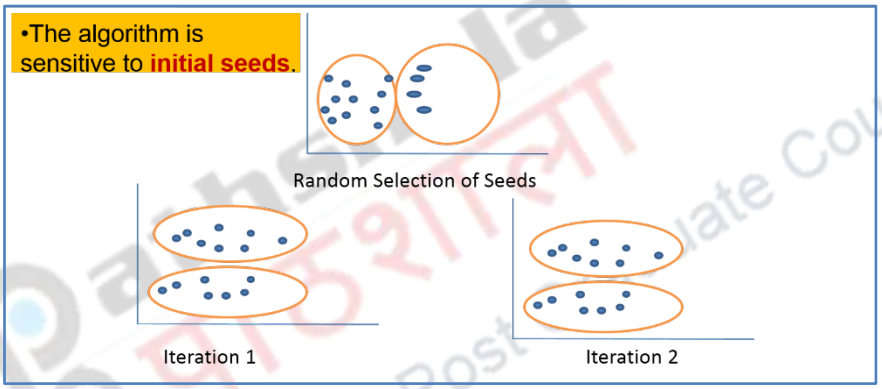


Figure 24.5 Effect of selection of Initial Seeds

according to the error in the clustering measured using in Mean Squared Error (MSE).

Bisecting K-means is a variant of K-means where the clusters are bisected using basic K-means and one of the clusters with the lowest MSE is selected from the bi-section.

24.3.3 Stopping/Convergence Criterion

The stopping criteria can be defined in any one of the following ways. We can stop the iteration when there are no or a minimum number of re-assignments of data points to different clusters. We can also define the stopping criteria to be when there is no (or minimum) change of centroids, or in terms of error as minimum decrease in the sum of squared error (SSE). We will discuss this error later on in this module in the section on evaluation.

24.3.3.1 Example

The numerical example given below can be used to understand the K-means algorithm. Suppose we have 4 types of medicines and each has two attributes pH and weight index (Figure 24.6). Our goal is to group these objects into *K*=2 group of medicines.

Medicine	Weight	pH-Index
A	1	1
B	2	1
C	4	3
D	5	4

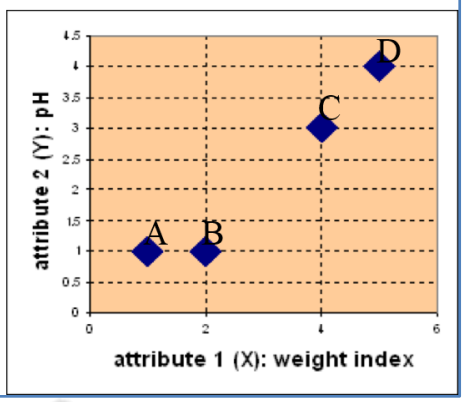
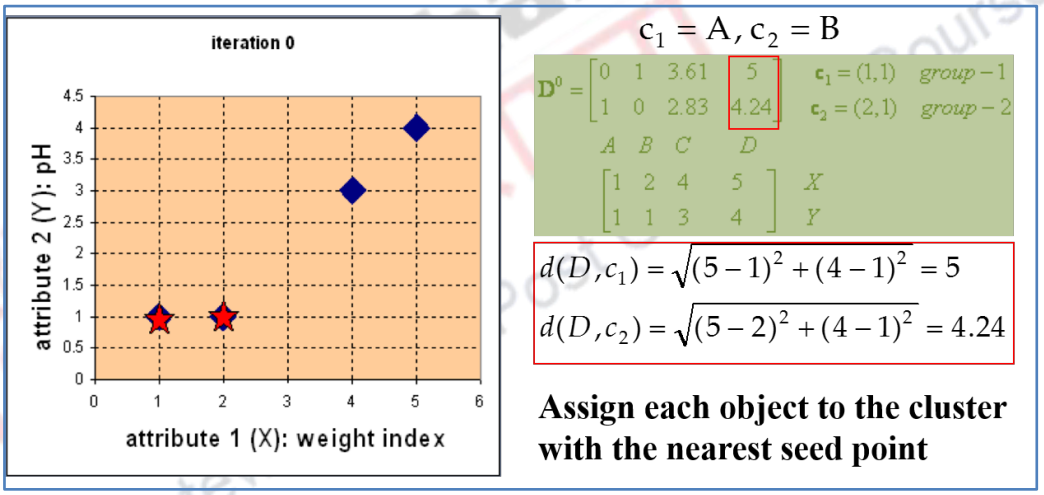
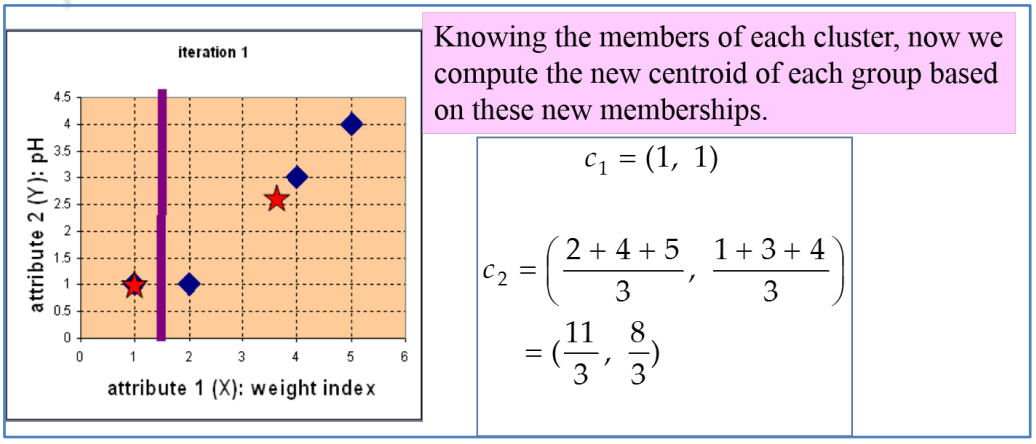


Figure 24.6 Mapping of the Data points into the Data Space

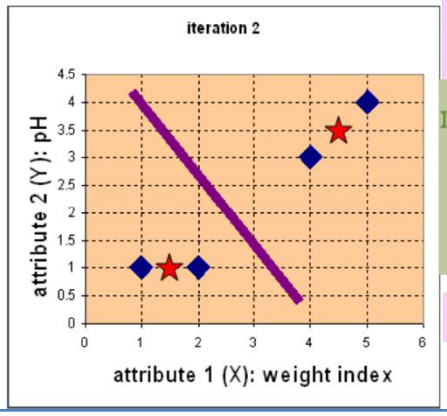


(a)



(b)

Step 2: Renew membership based on new centroids



Compute the distance of all objects to the new centroids

$$D^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \begin{matrix} c_1 = (1\frac{1}{2}, 1) & group-1 \\ c_2 = (4\frac{1}{2}, 3\frac{1}{2}) & group-2 \end{matrix}$$

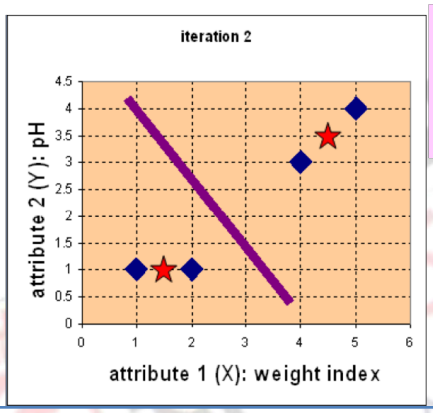
$$\begin{matrix} A & B & C & D \end{matrix}$$

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

Assign the membership to objects

(c)

Step 3: Repeat the first two steps until its convergence



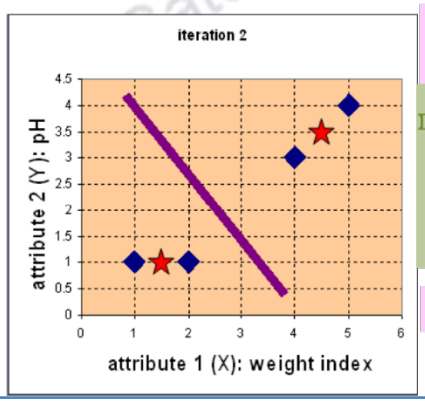
Knowing the members of each cluster, now we compute the new centroid of each group based on these new memberships.

$$c_1 = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = (1\frac{1}{2}, 1)$$

$$c_2 = \left(\frac{4+5}{2}, \frac{3+4}{2}\right) = (4\frac{1}{2}, 3\frac{1}{2})$$

(d)

Step 3: Repeat the first two steps until its convergence



Compute the distance of all objects to the new centroids

$$D^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \begin{matrix} c_1 = (1\frac{1}{2}, 1) & group-1 \\ c_2 = (4\frac{1}{2}, 3\frac{1}{2}) & group-2 \end{matrix}$$

$$\begin{matrix} A & B & C & D \end{matrix}$$

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

Assign the membership to objects

(e)

Figure 24.7 The Iterations for the example in Figure 24.5

Figure 24.7 (a) shows the random selection of the initial centroids as C_1 = (1,1) for cluster A and C_2=(2,1) for cluster B. We illustrate with data point D and calculate the distance of D with each of the randomly selected centroids (also sometimes called as seed points). We find that D is closer to $C_1$and we assign D to that cluster. Now Figure 24.7 (b) shows the assignment of each data point to one of the clusters. In our example only one data point D is assigned toAso it's centroid does not change. But however cluster B has three data points associated with it and the new centroid C_2now becomes (11/3, 8/3). Now we compute the new assignments as shown in Figure 24.7 (c). Figure 24.7 (d) shows the new values of C_1 and C_2. Figure 24.7 (e) shows the results after convergence.

24.4 Evaluating K-means Clusters

The most common measure is Sum of Squared Error (SSE). This is defined as follows:

For each point, the error is the distance to the nearest cluster

$$SSE = \sum_{j=1}^{k} \sum_{\mathbf{x} \in C_j} dist(\mathbf{x}, \mathbf{m}_j)^2$$

where C_j is the *j*th cluster, m_j is the centroid of cluster C_j (the mean vector of all the data points in C_j), and *dist*(x, m_j) is the distance between data point x and centroid m_j.

Given two clusterings (clustering is the set of clusters formed), we can choose the one with the smallest error. One straight forward way to reduce SSE is to increase K, the number of clusters. A good clustering with smaller K can have a lower SSE than a poor clustering with higher K

24.5 Strengths and Weaknesses of K-Means

24.5.1 Strengths of K-means

K-means is the most popular clustering algorithm The major strengths of K-means is that it is easy to understand and implement. It generally has a time complexity of the *O*(*tkn*), where *n* is the number of data points, K is the number of clusters, and *t* is the number of iterations. Since both *K* and *t* are small, *K*-

means is considered a linear time algorithm. It terminates at a local optimum if SSE is used. However the global optimum is hard to find due to complexity, which is its weakness.

24.5.2 Weaknesses of K-means

- The algorithm is only applicable for data where the concept of mean can be defined. For categorical data, where K-means is called as K-mode - the centroid is represented by most frequent values. The user needs to specify the value of K.
- A different initialization (selection of centroids) may sometimes produce a different clustering. The algorithm itself requires the labeling and interpretation of the clusters to be carried out in subsequent phase.
- The algorithm is sensitive to outliers, where outliers are data points that are very far away from other data points. These outliers could be errors in the data recording or some special data points with very different values. Including these outliers in the calculation of the centroid may affect the whole clustering process (Figure 24.8). One method to deal with outliers is to remove some data points in the clustering process that are much further away from the centroids than other data points. We may prefer to

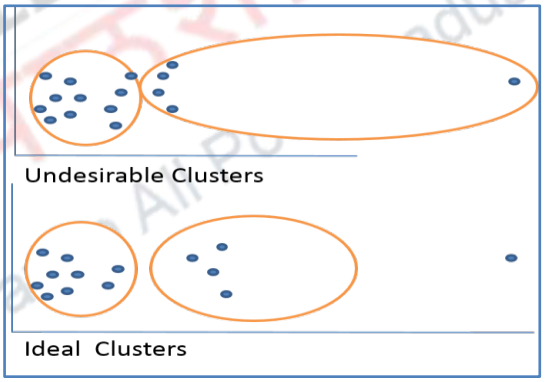


Figure 24.8 Influence of Outliers

monitor these possible outliers over a few iterations and then decide to remove them. Another method is to perform random sampling. Since in sampling we only choose a small subset of the data points, the chance of selecting an outlier is very small. Then we assign the rest of the data points to the clusters by distance or similarity comparison, or classification.

24.6 Applications

Some of the common applications are optical character recognition, biometrics, diagnostic systems, military applications, document clustering, etc..

24.7 Why K-Means?

Despite many of it's weaknesses, *K*-means is still the most popular algorithm due to its simplicity, efficiency and because other clustering algorithms have their own lists of weaknesses. In general there is no clear evidence that any other clustering algorithm performs better in general although some algorithms may be more suitable for some specific types of data or applications. Comparing different clustering algorithms is a difficult task since no knowledge of the correct clusters is available.

24.8 K-means Variations

Clustering typically assumes that each instance is given a definite or hard assignment to exactly one cluster. This means no uncertainty is allowed in class membership or for an instance to belong to more than one cluster. A variation called s*oft clustering* gives probabilities that an instance belongs to each of the set of clusters. In this case each data point is assigned a probability distribution across a set of discovered categories (probabilities of all categories must sum to 1).

Summary

- Discussed K-means Algorithm in detail
- Explained K-means Algorithm with an example
- Outlined the strengths & weaknesses of K-means Algorithm