Chapter-7

K-MEANS CLUSTERING
Chapter-7

K- MEANS CLUSTERING

7.1 K-Means Clustering

The k-means algorithm is a very popular algorithm for data clustering, since it is very simple to implement, it is fast, and it is fairly easy to understand. K-means clustering belongs to partitioning based clustering. It is an algorithm to classify the objects, which is based on attributes/features into K number of groups. K is positive integer number. The grouping is done by minimizing the sum of squares of distances between data and the corresponding cluster centroid. Thus the purpose of K-mean clustering is to classify the data.

The way k-means clustering works is that first the number of clusters (k) desired is specified, then the algorithm selects k cluster seeds (centers) which are located approximately uniformly in a multidimensional space. Each observation is then assigned to the nearest cluster mean to form temporary clusters. The cluster mean positions are then calculated and used as new cluster centers. The observations are then reallocated clusters according to the new cluster centers. This is repeated until no further change in the cluster centers occurs. The observations are assigned into clusters so that every observation belongs to at most one cluster.
7.1.1 Background Knowledge

Cluster center or centroid (mean of data points in cluster \( c \)) \( \bar{x}_c \)

\[
\bar{x}_c = \frac{\sum_{j}^{nc} \text{Data Point}_j^{(c)}}{n_c}
\]

Where: \( c = \text{cluster} \)

Data Point\(_j^{(c)}\) = an individual data point in cluster \( c \)

\( n_c = \text{total number of data points in cluster} \)

\( d \) is Euclidean distance:

\[
d(i, j) = \sqrt{(|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \ldots + |x_{ip} - x_{jp}|^2)}
\]

Which represent dissimilarity/similarity metric

Where: \( i = (x_{i1}, x_{i2}, X_{ip}) \) and \( j = (x_{j1}, x_{j2}, X_{jp}) \) are two \( p \)-dimensional data objects. Variability within a cluster \( C \) Error\(_c\), can be measured by the sum of squares for clusters:

\[
\text{Error}_c = \sum_{i}^{n_c} d_i^2
\]

Where:

\( s = \text{Euclidean distance between data point} \ i \) and designated cluster centre

\( c = \text{cluster} \)

\( n_c = \text{total number of data points in cluster} \ c \)

- The sum of the error of all the clusters that is for each cluster

\[
E = \sum_{c}^{k} \text{Error}_c
\]

Where: \( K = \text{total number of clusters} \)

7.2 K-means Algorithm

K-means is one of the simplest unsupervised learning algorithm that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set
through a certain fixed number of clusters (assume K clusters). The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no data set point is remaining, the first step is completed and an early grouping is done. After that, we need to recalculate K new centroids of the clusters resulting from the previous step. After we have these K new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the K centroids change their location step by step until no more changes are done. In other words centroids do not move any more. [1]

Finally, this algorithm aims at minimizing an objective function; in this case a sum squared error function. The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} || x_i^{(j)} - c_j ||^2$$

Where $$|| x_i^{(j)} - c_j ||^2$$ is a chosen distance measure between a data point $$x_i^{(j)}$$and the cluster centre $$C_j$$, is an indicator of the distance of the n data points from their respective cluster centres.
The algorithm is composed of the following steps:

**Figure 7.2.1: Flow Chart of K-mean Algorithm**

K-means algorithm does not necessarily find the most optimal configuration corresponding to the minimum global objective function. The algorithm is also significantly sensitive to the initial randomly selected cluster centers. The K-means algorithm can be run multiple times to reduce this effect.

**Step 1:** Algorithm will begin with the number of clusters, here K=2.

**Step 2:** Put any initial partition that classifies the data into K clusters. You may assign the training samples randomly or systematically as the following:

i. Take the first K training sample as single-element clusters.
ii. Assign each of the remaining (N-K) training samples to the cluster with the nearest centroid. After each assignment, recompute the centroid of the gaining cluster.

**Step 3:** Take each sample in sequence and compute its distance from the centroid of each of the clusters. If a sample is not currently in the cluster with the closest centroid, switch this sample to that cluster and update the centroid of the cluster gaining the new sample and the cluster losing the sample.

**Step 4:** Repeat step 3 until convergence is achieved, that is until a pass through the training sample causes no new assignments.

If the number of data is less than the number of cluster then assign each data as the centroid of the cluster. Each centroid will have a cluster number. If the number of data is bigger than the number of cluster, for each data, calculate the distance to all centroid and get the minimum distance. This data is said belong to the cluster that has minimum distance from this data.

Since the location of the centroid is not sure, we need to adjust the centroid location based on the current updated data. Then assign all the data to this new centroid. This process is repeated until no data is moving to another cluster anymore. Mathematically this loop can be proved to be convergent.

The convergence will always occur if the following condition satisfied:

- Each switch in step 2 the sum of distances from each training sample to that training sample's group centroid is decreased.
- There are only finitely many partitions of the training examples into k clusters.
7.2.1 K-Means Algorithm Properties

- There are always K clusters as K must be specified in advance.
- There is always at least one item in each cluster.
- The clusters are non-hierarchical and they do not overlap.
- Every member of a cluster is closer to its cluster than any other cluster because closeness does not always involve the 'center' of clusters.
- Each cluster is associated with a centroid (center point)
- Each point is assigned to the cluster with the closest centroid

7.2.2 An Example

Suppose the four objects (types of medicines) and each object have two attributes or features as shown in table below [3].

<table>
<thead>
<tr>
<th>Object</th>
<th>Feature (X)</th>
<th>Feature (Y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object A</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Object B</td>
<td>20</td>
<td>5</td>
</tr>
<tr>
<td>Object C</td>
<td>40</td>
<td>15</td>
</tr>
<tr>
<td>Object D</td>
<td>50</td>
<td>40</td>
</tr>
</tbody>
</table>

| Table 7.2.2.1: Set of Object

Four objects belong to two groups of object (cluster 1 and cluster 2) is known before hand. The problem now is to determine which object belong to cluster 1 and which object belongs to the other cluster. [4]

The goal is to group these objects into K=2 group of object based on the two features. Each object represents one point with two attributes (X, Y) that can represent it as coordinate in an attribute space as shown in the figure below.
1. **Initial value of centroids:** Suppose use the object A and object B as the first centroids. Let $C_1$ and $C_2$ denote the coordinate of the centroids, then $C_1 = (10,10)$ and $C_2 = (20,5)$

2. **Objects-Centroids distance:** calculate the distance between cluster centroid to each object and use Euclidean distance and then distance matrix at iteration 0 is

$$
D^0 = \begin{bmatrix}
0 & 11.18 & 30.41 & 50 \\
11.18 & 0 & 20.61 & 46.3
\end{bmatrix}
$$

Each column in the distance matrix symbolizes the object. The first row of the distance matrix corresponds to the distance of each object to the first centroid and the second row is the distance of each object to the second centroid. For example, distance from object $C = (40, 15)$ to the first centroid $C_1 = (10,10)$ is $\sqrt{(40-10)^2 + (15-10)^2} = 30.41$ and its distance to the second centroid $C_2 = (20,5)$ is $\sqrt{(40-20)^2 + (15-5)^2} = 20.61$ etc.
3. **Objects clustering:** Assign each object based on the minimum distance. Thus, object A is assigned to group 1, object B to group 2, object C to group 2 and object D to group 2. The element of Group matrix below is 1 if and only if the object is assigned to that group.

\[
G^0 = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1
\end{bmatrix}
group -1 \\
A & B & C & D
\]

4. **Iteration-1, determine centroids:** Knowing the member of each group, now compute the new centroid of each group based on these new memberships. Group 1 only has one member thus the centroid remains in \(C_1 = (10, 10)\). Group 2 now has three members, thus the centroid is the average coordinate among the three members:

\[
c_2 = \left( \frac{20 + 40 + 50}{3}, \frac{5 + 15 + 40}{3} \right) = \left( \frac{110}{3}, \frac{60}{3} \right)
\]

![Figure 7.2.2.2: K-mean algorithm iteration 1](image)

5. **Iteration-1, Objects-Centroids distances:** The next step is to compute the distance of all objects to the new centroids. Similar to step 2, distance matrix at iteration 1 is
\[
D' = \begin{bmatrix}
0 & 11.18 & 30.41 & 5 \\
19.43 & 21.93 & 6.01 & 20.04
\end{bmatrix}
\quad c_1 = \left(\frac{10}{3}, \frac{60}{3}\right) \text{ group - 1}
\]

\[
A \quad B \quad C \quad D
\begin{bmatrix}
10 & 20 & 40 & 50 \\
10 & 5 & 15 & 40
\end{bmatrix} X
\]

6. **Iteration-1, Objects clustering:** Similar to step 3, we assign each object based on the minimum distance. Based on the new distance matrix, we move the object B to Group 1 while all the other objects remain. The Group matrix is shown below

\[
G' = \begin{bmatrix}
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1
\end{bmatrix}
\]

7. **Iteration 2, determine centroids:** Now repeat step 4 to calculate the new centroids coordinate based on the clustering of previous iteration.

Group 1 and group 2 both have two members, thus the new centroids are:

\[
C_1 = \left(\frac{10 + 20}{2}, \frac{10 + 5}{2}\right) = and \quad C_2 = \left(\frac{40 + 50}{2}, \frac{15 + 40}{2}\right)
\]

![Figure 7.2.2.3: K-mean algorithm iteration 2](image-url)
8. **Iteration-2, Objects-Centroids distances**: Repeat step 2 again, then New distance matrix at iteration 2 as:

\[
D^2 = \begin{bmatrix}
5.59 & 5.59 & 26.1 & 47.76 \\
39.13 & 33.63 & 13.46 & 13.46 \\
\end{bmatrix}
\]

\(c_1 = (15, 7.5)\) **group** - 1

\(c_2 = (45, 27.5)\) **group** - 2

\[
A \quad B \quad C \quad D
\]
\[
X \\
Y
\]

9. **Iteration-2, Objects clustering**: Again, assign each object based on the minimum distance.

\[
G^2 = \begin{bmatrix}
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 \\
\end{bmatrix}
\]

\(group\) - 1

\(group\) - 2

\[
A \quad B \quad C \quad D
\]

Obtain result that \(G^2=G^1\). Comparing the grouping of last iteration and this iteration reveals that the objects does not move group anymore.

Thus, the computation of the K-mean clustering has reached its stability and no more iteration is needed and gets the final grouping as the results.

<table>
<thead>
<tr>
<th>Object</th>
<th>Attribute 1 (X)</th>
<th>Attribute 2 (Y):</th>
<th>Group (result)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object A</td>
<td>10</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>Object B</td>
<td>20</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Object C</td>
<td>40</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>Object D</td>
<td>50</td>
<td>40</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 7.2.2.2: Result of clustering
7.3 Applications

There are a lot of applications of the K-mean clustering, range from unsupervised learning of neural network, Pattern recognitions, Classification analysis, Artificial intelligent, image processing, machine vision, etc.

In principle, there are several objects and each object have several attributes and classify the objects based on the attributes then apply this algorithm.

Here is the principle of application of K-means clustering to machine learning or data mining. Each object represents one attribute and it is automatically assign to one of the cluster. We call this "unsupervised learning" because the algorithm classifies the object automatically only based on the criteria (i.e. minimum distance to the centroid). No need to supervise the program by saying that the classification was correct or wrong. The learning process is depending on the training examples that feeded to the algorithm.

Two choices in this learning process are following:
1. **Infinite training:** Each data that feed to the algorithm will automatically consider as the training examples.
2. **Finite training:** After the training is considered as finished (after it gives about the correct place of mean). Start to make the algorithm to work by classifying the cluster of new points. This is done simply by assign the point to the nearest centroid without recalculate the new centroid. Thus after the training finished, the centroid are fixed points.

**Advantages to Using this Technique**

- With a large number of variables, K-Means may be computationally faster than hierarchical clustering (if K is small).
• K-Means may produce tighter clusters than hierarchical clustering, especially if the clusters are globular.

7.4 Weaknesses of k-mean

• When the data set is not large, initial grouping will determine the cluster significantly.
• The number of cluster K must be determined before hand.
• Sensitive to initial condition i.e. Different initial condition may produce different result of cluster.
• This is not known which attribute contributes more to the grouping process since; assume that each attribute has the same weight.
• Resultant cluster shape will be circular which is based on distance.
• Difficulty in comparing the quality of the clusters produced (e.g. for different initial partitions or values of K affect outcome).
• Does not work well with non-globular clusters (do not have well defined centers)
• Different initial partitions can result in different final clusters. It is helpful to rerun the program using the same as well as different K values to compare the results achieved.

REFERENCES


