

## K-Means Clustering Tutorial

By Kardi Teknomo, PhD

Preferable reference for this tutorial is

**Teknomo, Kardi. K-Means Clustering Tutorials.** <http://people.revoledu.com/kardi/tutorial/kMean/>

Last Update: July 2007

### What is K-Means Clustering?

Simply speaking it is an algorithm to classify or to group your objects based on attributes/features into K number of group. 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.

Example: Suppose we have 4 objects as your training data point and each object have 2 attributes. Each attribute represents coordinate of the object.

Object	Attribute 1 (X): weight index	Attribute 2 (Y): pH
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4

Thus, we also know before hand that these objects belong to two groups of medicine (cluster 1 and cluster 2). The problem now is to determine which medicines belong to cluster 1 and which medicines belong to the other cluster. Each medicine represents one point with two components coordinate.

### Numerical Example (manual calculation)

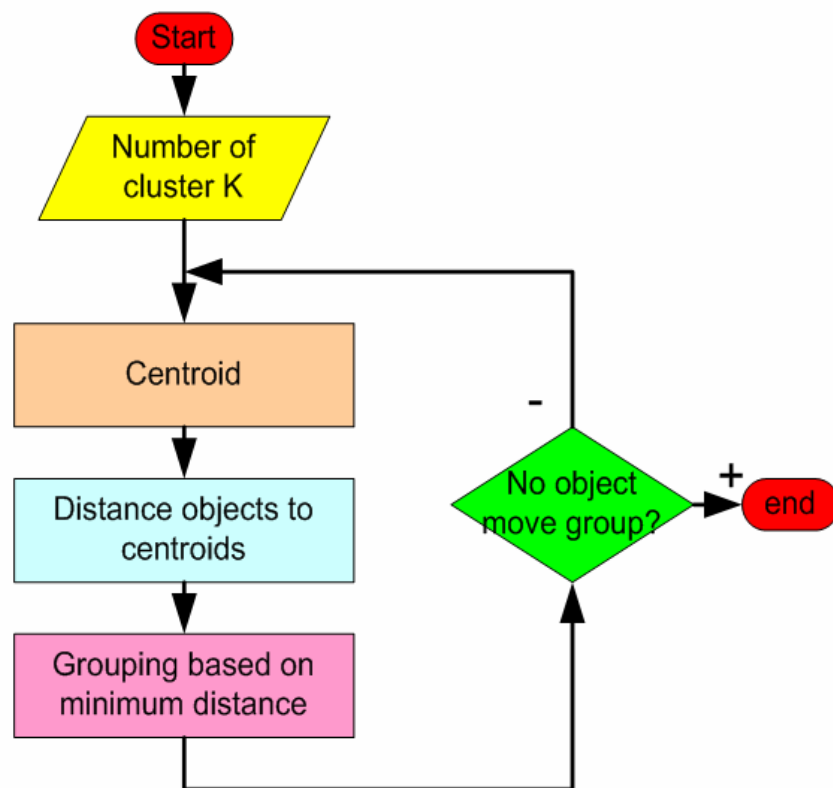
The basic step of k-means clustering is simple. In the beginning, we determine number of cluster K and we assume the centroid or center of these clusters. We can take any random objects as the initial centroids or the first K objects can also serve as the initial centroids.

Then the K means algorithm will do the three steps below until convergence

Iterate until *stable* (= no object move group):

1. Determine the centroid coordinate
2. Determine the distance of each object to the centroids
3. Group the object based on minimum distance (find the closest centroid)

<http://people.revoledu.com/kardi/tutorial/kMean/index.html>

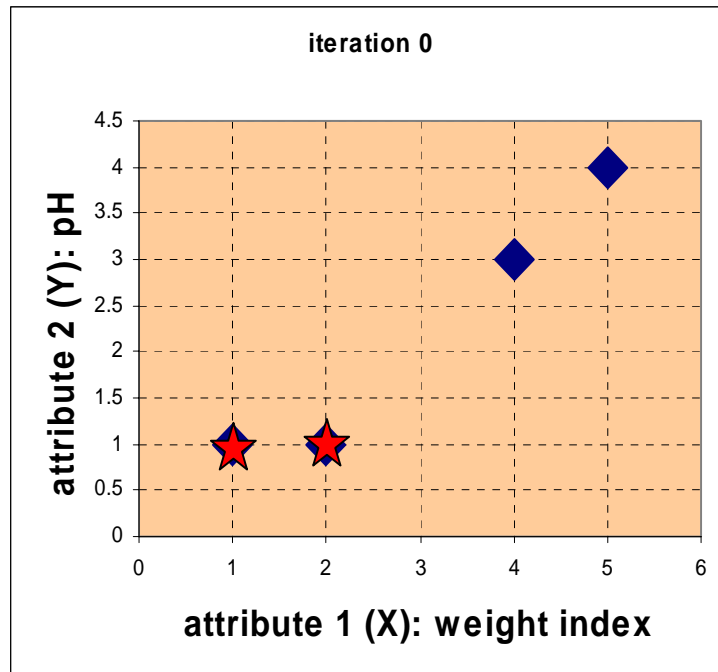


The numerical example below is given to understand this simple iteration.

Suppose we have several objects (4 types of medicines) and each object have two attributes or features as shown in table below. Our goal is to group these objects into  $K=2$  group of medicine based on the two features (pH and weight index).

Object	Feature 1 (X): weight index	Feature 2 (Y): pH
Medicine A	1	1
Medicine B	2	1
Medicine C	4	3
Medicine D	5	4

Each medicine represents one point with two features (X, Y) that we can represent it as coordinate in a feature space as shown in the figure below.



1. *Initial value of centroids:* Suppose we use medicine A and medicine B as the first centroids. Let  $\mathbf{c}_1$  and  $\mathbf{c}_2$  denote the coordinate of the centroids, then  $\mathbf{c}_1 = (1, 1)$  and  $\mathbf{c}_2 = (2, 1)$
2. *Objects-Centroids distance:* we calculate the distance between cluster centroid to each object. Let us use Euclidean distance, then we have distance matrix at iteration 0 is

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

A	B	C	D	
1	2	4	5	X
1	1	3	4	Y

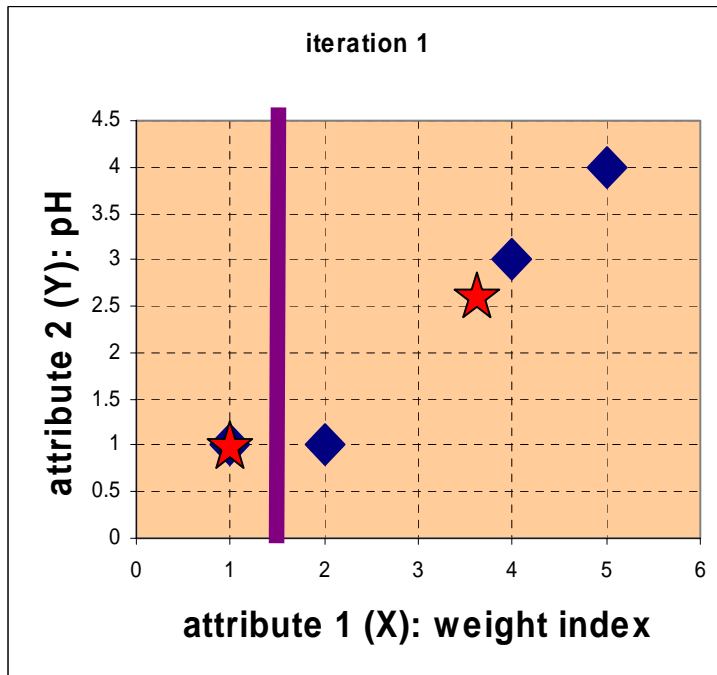
Each column in the distance matrix symbolizes the object. The first row of the distance matrix corresponds to the distance of each object to the first centroid and the second row is the distance of each object to the second centroid. For example, distance from medicine C = (4, 3) to the first centroid  $\mathbf{c}_1 = (1, 1)$  is  $\sqrt{(4-1)^2 + (3-1)^2} = 3.61$ , and its distance to the second centroid  $\mathbf{c}_2 = (2, 1)$  is  $\sqrt{(4-2)^2 + (3-1)^2} = 2.83$ , etc.

3. *Objects clustering:* We assign each object based on the minimum distance. Thus, medicine A is assigned to group 1, medicine B to group 2, medicine C to group 2 and medicine D to group 2. The element of Group matrix below is 1 if and only if the object is assigned to that group.

$$\mathbf{G}^0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \quad \begin{array}{l} \text{group-1} \\ \text{group-2} \end{array}$$

A	B	C	D
1	0	0	0
0	1	1	1

4. *Iteration-1, determine centroids:* Knowing the members of each group, now we compute the new centroid of each group based on these new memberships. Group 1 only has one member thus the centroid remains in  $\mathbf{c}_1 = (1,1)$ . Group 2 now has three members, thus the centroid is the average coordinate among the three members:  $\mathbf{c}_2 = (\frac{2+4+5}{3}, \frac{1+3+4}{3}) = (\frac{11}{3}, \frac{8}{3})$ .



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, we have distance matrix at iteration 1 is

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

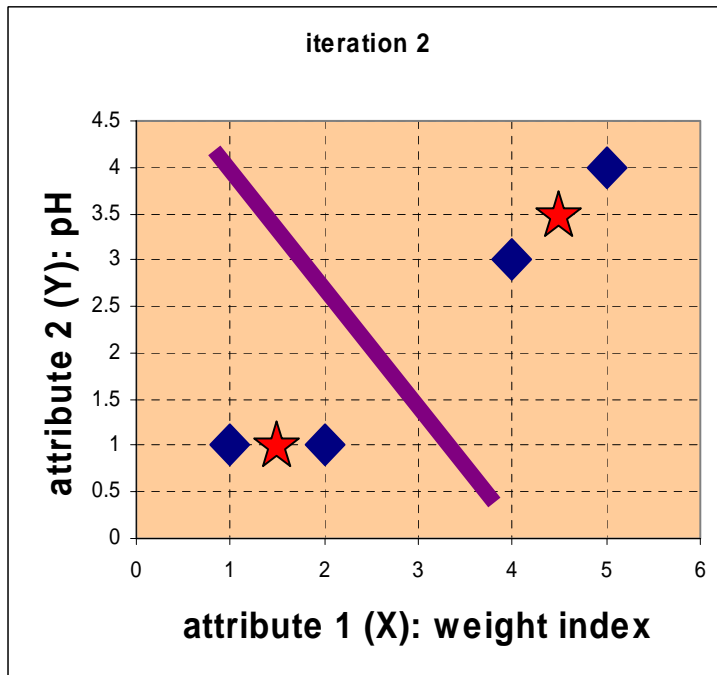
	A	B	C	D	
$\begin{bmatrix} 1 & 2 & 4 & 5 \end{bmatrix}$	1	2	4	5	X
$\begin{bmatrix} 1 & 1 & 3 & 4 \end{bmatrix}$	1	1	3	4	Y

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 medicine B to Group 1 while all the other objects remain. The Group matrix is shown below

$$\mathbf{G}^1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \begin{matrix} \text{group-1} \\ \text{group-2} \end{matrix}$$

	A	B	C	D
1	1	1	0	0
0	0	0	1	1

7. *Iteration 2, determine centroids:* Now we repeat step 4 to calculate the new centroids coordinate based on the clustering of previous iteration. Group1 and group 2 both has two members, thus the new centroids are  $\mathbf{c}_1 = (\frac{1+2}{2}, \frac{1+1}{2}) = (1\frac{1}{2}, 1)$  and  $\mathbf{c}_2 = (\frac{4+5}{2}, \frac{3+4}{2}) = (4\frac{1}{2}, 3\frac{1}{2})$



8. *Iteration-2, Objects-Centroids distances:* Repeat step 2 again, we have new distance matrix at iteration 2 as

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

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

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

$$G^2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad \begin{array}{l} \text{group-1} \\ \text{group-2} \end{array}$$

$$\begin{array}{cccc} A & B & C & D \end{array}$$

We 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. We get the final grouping as the results

Object	Feature 1 (X): weight index	Feature 2 (Y): pH	Group (result)
Medicine A	1	1	1
Medicine B	2	1	1
Medicine C	4	3	2
Medicine D	5	4	2

***How the K-Mean Clustering algorithm works?***

If the number of data is less than the number of cluster then we 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, we 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 we are not sure about the location of the centroid, we need to adjust the centroid location based on the current updated data. Then we 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 convergent.

As an example, I have made a Visual Basic and Matlab code. You may download the complete program in <http://www.planetsourcecode.com/xq/ASP/txtCodeId.26983/IngWId.1/qx/vb/scripts/ShowCode.htm>, or in its official web page in <http://people.revoledu.com/kardi/tutorial/kMean/download.htm>

The number of features is limited to two only but you may extent it to any number of features. The main code is shown here.

```
Sub kMeanCluster (Data() As Variant, numCluster As Integer)
' main function to cluster data into k number of Clusters
' input: + Data matrix (0 to 2, 1 to TotalData); Row 0 = cluster, 1 =X, 2= Y; data in columns
'       + numCluster: number of cluster user want the data to be clustered
'       + private variables: Centroid, TotalData
' output: o) update centroid
'         o) assign cluster number to the Data (= row 0 of Data)
Dim i As Integer
Dim j As Integer
Dim X As Single
Dim Y As Single
Dim min As Single
Dim cluster As Integer
Dim d As Single
Dim sumXY()
Dim isStillMoving As Boolean

isStillMoving = True

If totalData <= numCluster Then
    Data(0, totalData) = totalData ' cluster No = total data
    Centroid(1, totalData) = Data(1, totalData) ' X
    Centroid(2, totalData) = Data(2, totalData) ' Y
Else
    'calculate minimum distance to assign the new data
    min = 10 ^ 10 'big number
    X = Data(1, totalData)
    Y = Data(2, totalData)
    For i = 1 To numCluster
        d = dist(X, Y, Centroid(1, i), Centroid(2, i))
        If d < min Then
            min = d
            cluster = i
        End If
    Next i
    Data(0, totalData) = cluster

    Do While isStillMoving
        ' this loop will surely convergent

        'calculate new centroids
        ReDim sumXY(1 To 3, 1 To numCluster) ' 1 =X, 2=Y, 3=count number of data
        For i = 1 To totalData
```

<http://people.revoledu.com/kardi/tutorial/kMean/index.html>

```

sumXY(1, Data(0, i)) = Data(1, i) + sumXY(1, Data(0, i))
sumXY(2, Data(0, i)) = Data(2, i) + sumXY(2, Data(0, i))
sumXY(3, Data(0, i)) = 1 + sumXY(3, Data(0, i))
Next i
For i = 1 To numCluster
  Centroid(1, i) = sumXY(1, i) / sumXY(3, i)
  Centroid(2, i) = sumXY(2, i) / sumXY(3, i)
Next i

'assign all data to the new centroids
isStillMoving = False
For i = 1 To totalData
  min = 10 ^ 10          'big number
  X = Data(1, i)
  Y = Data(2, i)
  For j = 1 To numCluster
    d = dist(X, Y, Centroid(1, j), Centroid(2, j))
    If d < min Then
      min = d
      cluster = j
    End If
  Next j
  If Data(0, i) <> cluster Then
    Data(0, i) = cluster
    isStillMoving = True
  End If
Next i
Loop
End If
End Sub

```

The schematic of 3 matrix variables are given below

Data

	1	2	3	...	Total data	
0						Cluster number
1						X
2						Y

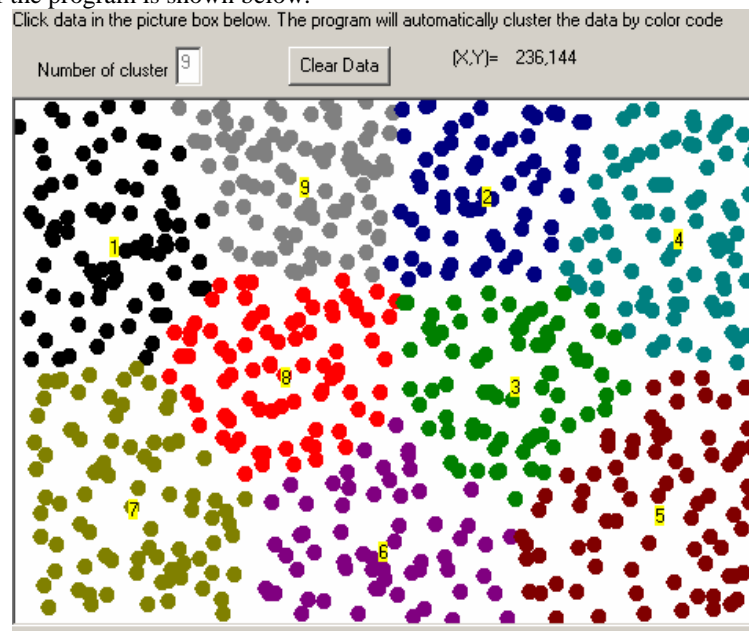
## SumXY

	1	2	3	...	Cluster number
1					X
2					Y
3					Count number of data in the cluster

## Centroid

	1	2	3	...	Cluster number
1					X
2					Y

The screen shot of the program is shown below.



When User click picture box to input new data (X, Y), the program will make group/cluster the data by minimizing the sum of squares of distances between data and the corresponding cluster centroid. Each dot is representing an object and the coordinate (X, Y) represents two attributes of the object. The colors of the dot and label number represent the cluster. You may try how the cluster may change when additional data is inputted.

<http://people.revoledu.com/kardi/tutorial/kMean/index.html>



For you who like to use Matlab, Matlab Statistical Toolbox contains a function name **kmeans**. If you do not have the statistical toolbox, you may use my code below. The **kMeanCluster** and **distMatrix** can be downloaded as text files in [http://people.revoledu.com/kardi/tutorial/kMean/matlab\\_kMeans.htm](http://people.revoledu.com/kardi/tutorial/kMean/matlab_kMeans.htm). Alternatively, you may simply type the code below.

```
function y=kMeansCluster(m,k)
    %%%%%%%%%%%%%%%
    % kMeansCluster - Simple k means clustering algorithm
    % Author: Kardi Teknomo, Ph.D.
    %
    % Purpose: classify the objects in data matrix based on the attributes
    % Criteria: minimize Euclidean distance between centroids and object points
    % For more explanation of the algorithm, see http://people.revoledu.com/kardi/tutorial/kMean/index.html %
    % Output: matrix data plus an additional column represent the group of each object %
    %
    % Example: m = [ 1 1; 2 1; 4 3; 5 4] or in a nice form
    %           m = [ 1 1;
    %                 2 1;
    %                 4 3;
    %                 5 4]
    %           k = 2
    % kMeansCluster(m,k) produces m = [ 1 1 1;
    %                                   2 1 1;
    %                                   4 3 2;
    %                                   5 4 2]
    % Input:
    % m - matrix data: objects in rows and attributes in columns
    % k - number of groups
    %
    % Local Variables
    % c - centroid coordinate size (1:k, 1:maxCol)
    % g - current iteration group matrix size (1:maxRow)
    % i - scalar iterator
    % maxCol - scalar number of rows in the data matrix m = number of attributes
    % maxRow - scalar number of columns in the data matrix m = number of objects
    % temp - previous iteration group matrix size (1:maxRow)
    % z - minimum value (not needed)
    %%%%%%%%%%%%%%%

    [maxRow, maxCol]=size(m);
    if maxRow<=k,
        y=[m, 1:maxRow];
    else

        % initial value of centroid
        for i=1:k
            c(i,:)=m(i,:);
        end

        temp=zeros(maxRow,1); % initialize as zero vector

        while 1,
            d=DistMatrix(m,c); % calculate objects-centroid distances
            [z,g]=min(d,[],2); % find group matrix g
            if g==temp,
                break; % stop the iteration
            else
                temp=g; % copy group matrix to temporary variable
            end

            for i=1:k
                c(i,:)=mean(m(find(g==i,:),:));
            end
        end
    end
end
```

<http://people.revoledu.com/kardi/tutorial/kMean/index.html>

```

end
y=[m,g];
end

```

The Matlab function kMeansCluster above call function DistMatrix as shown in the code below.

```

function d=DistMatrix(A,B)
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % DISTMATRIX return distance matrix between point A=[x1 y1] and B=[x2 y2]
    % Author: Kardi Teknomo, Ph.D.
    % see http://people.revoledu.com/kardi/
    %
    % Number of point in A and B are not necessarily the same.
    % It can be use for distance-in-a-slice (Spacing) or distance-between-slice (Headway),
    %
    % A and B must contain two column,
    % first column is the X coordinates
    % second column is the Y coordinates
    % The distance matrix are distance between points in A as row
    % and points in B as column.
    % example: Spacing= dist(A,A)
    % Headway = dist(A,B), with hA ~= hB or hA=hB
    % A=[1 2; 3 4; 5 6]; B=[4 5; 6 2; 1 5; 5 8]
    % dist(A,B)= [ 4.24 5.00 3.00 7.21;
    % 1.41 3.61 2.24 4.47;
    % 1.41 4.12 4.12 2.00 ]
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    [hA,wA]=size(A);
    [hB,wB]=size(B);
    if hA==1 & hB==1
        d=sqrt(dot((A-B),(A-B)));
    else
        C=[ones(1,hB);zeros(1,hB)];
        D=flipud(C);
        E=[ones(1,hA);zeros(1,hA)];
        F=flipud(E);
        G=A*C;
        H=A*D;
        I=B*E;
        J=B*F;
        d=sqrt((G-I).^2+(H-J).^2);
    end
end

```

### *Is the algorithm always convergence?*

Anderberg as quoted by [1] suggest the convergent k-mean clustering algorithm:

**Step 1.** Begin with any initial partition that groups the data into  $k$  clusters. One way to do this is:

- a. Take the first  $k$  training examples as single-element clusters
- b. Assign each of the remaining  $N - k$  examples to the cluster with the nearest centroid. After each assignment, recomputed the centroid of the gaining cluster.

**Step 2.** Take each example  $\mathbf{p}_j$  in sequence and compute its distance from the centroid of each of the  $k$  clusters. If  $\mathbf{p}_j$  is not currently in the cluster with the closest centroid, switch  $\mathbf{p}_j$  to that cluster and update the centroid of the cluster gaining  $\mathbf{p}_j$  and the cluster losing  $\mathbf{p}_j$ .

**Step 3.** Repeat step 2 until convergence is achieved, that is until a pass through the training examples causes no new assignments.

It is proven that the convergence will always occur if the following condition satisfied:

<http://people.revoledu.com/kardi/tutorial/kMean/index.html>

1. Each switch in step 2 decreases the sum of the squared distances from each training example to that training example's group centroid.
2. There are only finitely many partitions of the training examples into  $k$  cluster.

#### ***What are the applications of K-mean clustering?***

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, you have several objects and each object have several attributes and you want to classify the objects based on the attributes, then you can apply this algorithm.

#### ***What are the weaknesses of K-Mean Clustering?***

Similar to other algorithm, K-mean clustering has many weaknesses:

- When the numbers of data are not so many, initial grouping will determine the cluster significantly.
- The number of cluster,  $K$ , must be determined before hand.
- We never know the real cluster, using the same data, if it is inputted in a different way may produce different cluster if the number of data is a few.
- We never know which attribute contributes more to the grouping process since we assume that each attribute has the same weight.

One way to overcome those weaknesses is to use K-mean clustering only if there are available *many* data.

#### ***How if we have more than 2 attributes?***

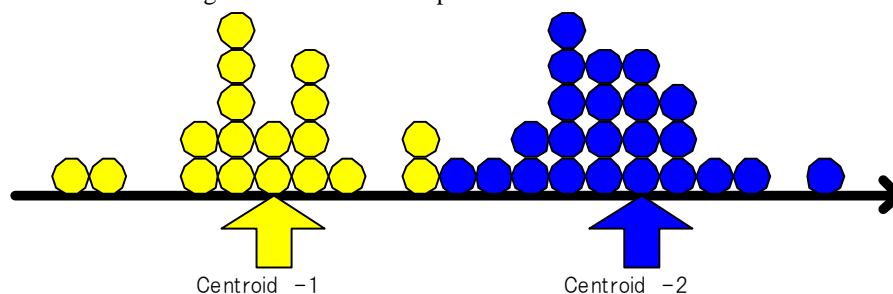
To generalize the k-mean clustering into  $n$  attributes, we define the *centroid* as a vector where each component is the average value of that component. Each component represents one attribute. Thus each point number  $j$  has  $n$  components or denoted by  $p_j(x_{j1}, x_{j2}, x_{j3}, \dots, x_{jm}, \dots, x_{jn})$ . If we have  $N$  training points, then the  $m$  component of centroid can be calculated as:

$$\bar{x}_m = \frac{1}{N} \sum_j x_{jm}$$

The rest of the algorithm is just the same as above.

#### ***What is the minimum number of attribute?***

As you may guess, the minimum number of attribute is one. If the number of attribute is one, each example point represents a point in a distribution. The k-mean algorithm becomes the way to calculate the mean value of  $k$  distributions. Figure below is an example of  $k = 2$  distributions.



***Where is the learning process of  $k$  - mean clustering?***

Each object represented by one attribute point is an example to the algorithm and it is assigned automatically to one of the cluster. We call this “unsupervised learning” because the algorithm classifies the object automatically only based on the criteria that we give (i.e. minimum distance to the centroid). We don’t need to supervise the program by saying that the classification was correct or wrong. The learning process is depending on the training examples that you feed to the algorithm. You have two choices in this learning process:

1. Infinite training. Each data that feed to the algorithm will automatically consider as the training examples. The VB program above is on this type.
2. Finite training. After the training is considered as finished (after it gives about the correct place of mean). We 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.

For example using one attribute: during learning phase, we assigned each example point to the appropriate cluster. Each cluster represents one distribution. After finishing the training phase, if we are given a point, the algorithm can assign this point to one of the existing distribution. If we use infinite training, then any point given by user is also classified to the appropriate distribution and it is also considered as a new training point.

***Are there any other resources for K-mean Clustering?***

There are many books and journals or Internet resources discuss about K-mean clustering, your search must be depending on your application. Here are some lists: of my references: for this tutorial.

1. Gallant, Stephen I., Neural Network Learning and expert systems, the MIT press, London,1993, pp. 134-136.
2. Anderberg, M.R., Cluster Analysis for Applications, Academic Press, New York, 1973, pp. 162-163.
3. Costa, Luciano da Fontoura and Cesar, R.M., Shape Analysis and Classification, Theory and Practice, CRC Press, Boca Raton, 2001, pp 577-615.

For more updated information about this tutorial, visit the official page of this tutorial:

<http://people.revoledu.com/kardi/tutorial/kMean/index.html>

<http://people.revoledu.com/kardi/tutorial/kMean/index.html>