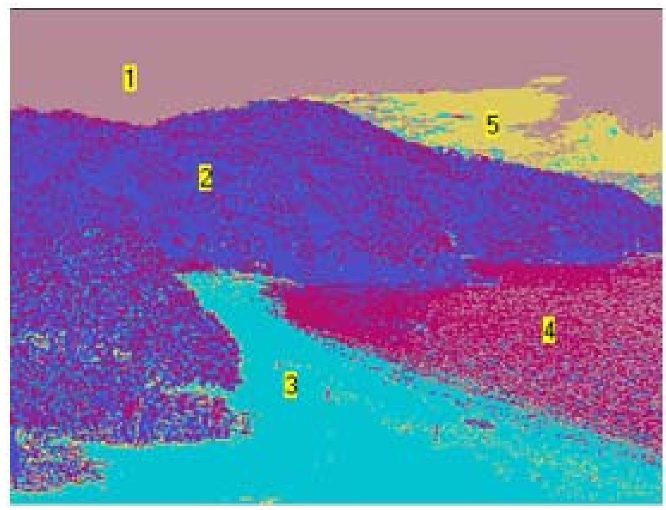
Kardi Teknomo

K-MEANS CLUSTERING TUTORIAL



Revoledu.com

K Means Tutorial by Kardi Teknomo

Published by Revoledu.com
Online edition is available at Revoledu.com
Last Update: October 2012

Table of Contents

What is K-Means Clustering?

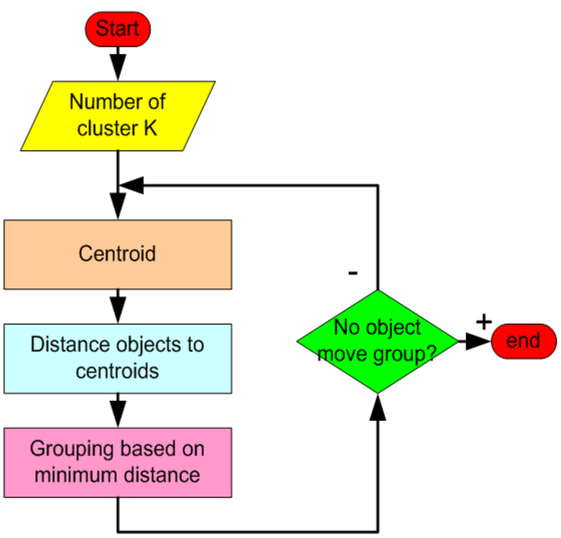
K means clustering algorithm was developed by J. MacQueen (1967) and then improved by J. A. Hartigan and M. A. Wong around 1975. Simply speaking k means clustering is an algorithm to classify or to group your objects based on attributes/features into K number of group. K is positive integer number. The 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.

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.

K means algorithm is using the three steps below until it converges.

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)

The flow chart of the algorithm is given below.



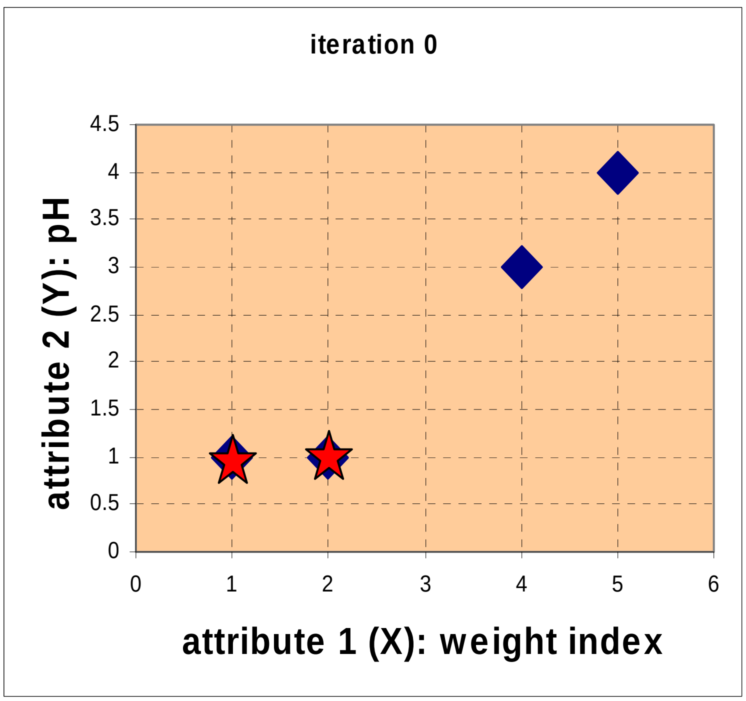
K Means Numerical Example (manual calculation)

Let us illustrate the k means algorithm with a numerical example.

Suppose we have several objects (4 types of medicines) and each object have 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). These 4 objects are called *training data points*.

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

Each object in our example has 2 attributes. We can represent each attribute as a coordinate in 2-dimensional chart. Each medicine represents one point with two components coordinate. We call the two components of the coordinate as (X, Y). Thus, we can represent objects as points in a feature space as shown in the figure below.



To use k means algorithm, we must *know beforehand* that these objects belong to two groups of medicine (let name these two groups as cluster 1 and cluster 2). The stars represent the initial location of the two centroids. The problem now is to determine which medicines belong to cluster 1 and which medicines belong to the other cluster.

Step by steps k means algorithm is as follow:

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

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

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

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

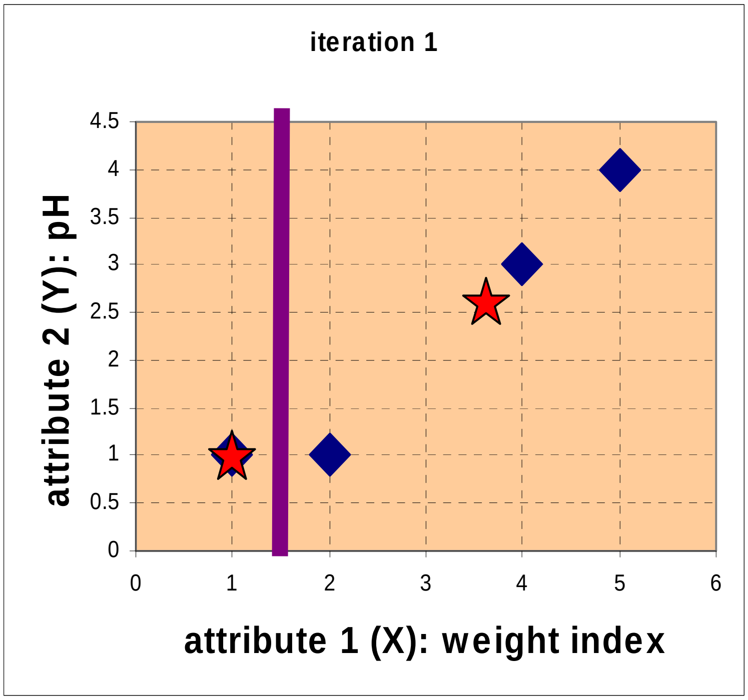
Each column in the distance matrix symbolizes the object. The first row of the distance matrix corresponds to the distance of each object to the first centroid and the second row is the distance of each object to the second centroid. For example, distance from medicine C = (4, 3) to the first centroid $\mathbf{c}_1 = (1,1)$ 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.

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

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

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

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

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

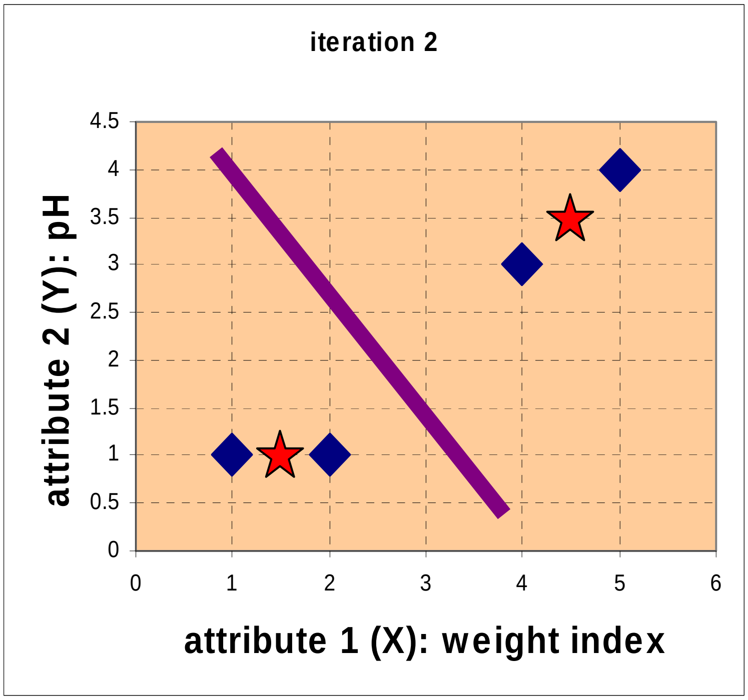
$$\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \begin{array}{l} X \\ Y \end{array}$$

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

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

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

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

$$\mathbf{D}^2 = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix} \quad \begin{array}{ll} \mathbf{c}_1 = (1\tfrac{1}{2}, 1) & group-1 \\ \mathbf{c}_2 = (4\tfrac{1}{2}, 3\tfrac{1}{2}) & group-2 \end{array}$$

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

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

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

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

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

We obtain result that $\mathbf{G}^2 = \mathbf{G}^1$. 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 does the K-Mean Clustering algorithm work?

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 identification 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.

To illustrate how the k means algorithm works, this tutorial is accompanied by several files which are codes in Matlab, in Visual Basic and in Microsoft Excel.

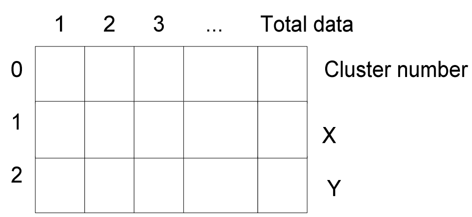
The number of features is limited to two only but you may extent it to any number of features. The main code is shown here.

```
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
' ouput: 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
```

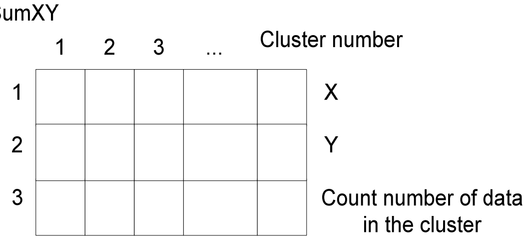
```
   Do While isStillMoving
   ' this loop will surely convergent

      '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
         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



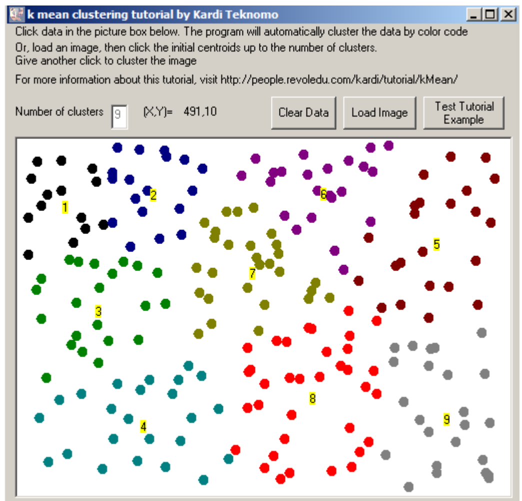
SumXY



Centroid

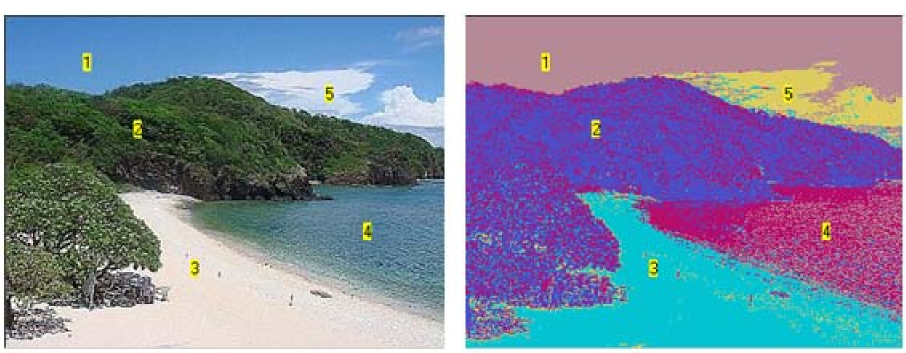
	1	2	3	...		Cluster number
1						X
2						Y

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.

The VB program also has a simple application to image processing. When you run the program, you may load any picture and then clicks in the image, the initial centroids. After you the click as many as the number of clusters you have specified, then the next click on the image will run the program. Sample image below shows the sample picture that you have in the companion files that have been clustered into 5 groups.



When you set different initial values, you may get different results.

For those of you who like to use Matlab, you can use Matlab Statistical Toolbox which contains a function name **kmeans**. If you do not have the statistical toolbox, you may use my code below. The **kMeanCluster** and **distMatrix** is provided together with this text.

```
function [y,c]=kMeansCluster(m,k,isRand)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  kMeansCluster - Simple k means clusteringalgorithm
% 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
% 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
%   isRand - optional, if using random initialization isRand=1,
%            otherwise input any number (default)it will assign
%            the first k data as initial centroids
%
% 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)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%c
close all
clc
if nargin<1,
     m = [ 1 1; 2 1; 4 3; 5 4];
end
if nargin<2,
    k=2;
end
if nargin<3,
    isRand=1;
end

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

       % initial value of centroid
    if isRand,
        p = randperm(size(m,1));      % random initialization
        for i=1:k
            c(i,:)=m(p(i),:);
       end
    else
        for i=1:k
           c(i,:)=m(i,:);        % sequential initialization
       end
```

```
    end

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

        while 1,
         d=DistMatrix(m,c);  % calculate objcets-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

        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 points in A=[x1 y1 ... w1] and in B=[x2 y2 ...
w2]
 % Copyright (c) 2005 by Kardi Teknomo,  http://people.revoledu.com/kardi/
 %
 % Numbers of rows (represent points) 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 the same number of columns (represent variables of n dimensions),
 % first column is the X coordinates, second column is the Y coordinates, and so on.
 % The distance matrix is distance between points in A as rows
 % and points in B as columns.
 % example: Spacing= dist(A,A)
 % Headway = dist(A,B), with hA ~= hB or hA=hB
 %          A=[1 2 3; 4 5 6; 2 4 6; 1 2 3]; B=[4 5 1; 6 2 0]
 %          dist(A,B)= [ 4.69   5.83;
 %                       5.00   7.00;
 %                       5.48   7.48;
 %                       4.69   5.83]
 %
 %          dist(B,A)= [ 4.69   5.00     5.48    4.69;
 %                       5.83   7.00     7.48    5.83]
 %%%%%%%%%%%%%%%%%%%%%%%%%%%
clc
if nargin<1,
    A=[1 2 3;
       4 5 6;
       2 4 6;
       1 2 3];
end
if nargin<2,
    B=[4 5 1;
       6 2 0;];
end
[hA,wA]=size(A);
[hB,wB]=size(B);
if wA ~= wB,  error(' second dimension of A and B must be the same'); end
for k=1:wA
  C{k}= repmat(A(:,k),1,hB);
  D{k}= repmat(B(:,k),1,hA);
end
```

```
S=zeros(hA,hB);
for k=1:wA
  S=S+(C{k}-D{k}').^2;
end
d=sqrt(S);
```

The Matlab code can be run for any dimensions, not only for two dimensions as in the numerical example.

The file companion of this tutorial also contains a spreadsheet file. You can use the spreadsheet example simply by deleting the control of cell that says “Delete Me”.

Control | delete me

	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Kardi's Example				Control	delete me									
2															
3									square distance from center				Binary of Min distance		
4	Object	feature 1	feature 2		Initial values				Object	1	2		Object	1	2
5	1	2	2		center	feature 1	feature 2		1	0.0	1.3		1	1	0
6	2	1.5	1		1	2	2		2	1.3	0.0		2	0	1
7	3	4	3		2	1.5	1		3	5.0	10.3		3	1	0
8	4	5	4						4	13.0	21.3		4	1	0
9															
10					Computed center										
11					center	feature 1	feature 2								
12					1	2	2								
13					2	1.5	1								
14															
15															
16															
17															
18					This Workbook is a companion of Kardi Teknomo's tutorial on K Mean Clustering										

Is k means algorithm always convergence?

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

Step 1. Begin with a decision on the value of k = number of clusters

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:

1. Take the first k training sample as single-element clusters
2. Assign each of the remaining (N-k) training samples to the cluster with the nearest centroid. After each assignment, recomputed the centroid of the gaining cluster.

Step 3. Take each sample $\mathbf{p}_j$ in sequence and compute its distance from the centroid of each of the k clusters. If the sample $\mathbf{p}_j$ is not currently in the cluster with the closest centroid, switch this sample $\mathbf{p}_j$ 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 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 to be convergent. The convergence will always occur if the following condition satisfied:

1. Each switch in step 3 the sum of distances from each training sample to that training sample's group centroid is decreased.
2. There are only finitely many partitions of the training examples into k clusters.

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 many other algorithms, 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 beforehand.
- We never know the real cluster, using the same data, if it is inputted in a different order may produce different cluster if the number of data is a few.
- Sensitive to initial condition. Different initial condition may produce different result of cluster. The algorithm may be trapped in the local optimum.
- We never know which attribute contributes more to the grouping process since we assume that each attribute has the same weight.

- The weakness of arithmetic mean is not robust to outliers. Very far data from the centroid may pull the centroid away from the real one.
- The result is circular cluster shape because based on distance.

One way to overcome those weaknesses is to use K-mean clustering only if there are available many data. To overcome outliers problem, we can use median instead of mean. To overcome weakness of k means, several algorithms had been proposed such as k medoids, fuzzy c mean and k mode.

Some people pointed out that K means clustering cannot be used for other type of data rather than quantitative data. This is not true! See how you can use multivariate data up to n dimensions (even mixed data type) here. The key to use other type of dissimilarity is in the distance matrix

What if I 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 $p_j(x_{j1}, x_{j2}, x_{j3}, ..., x_{jm}, ..., x_{jn})$. 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.

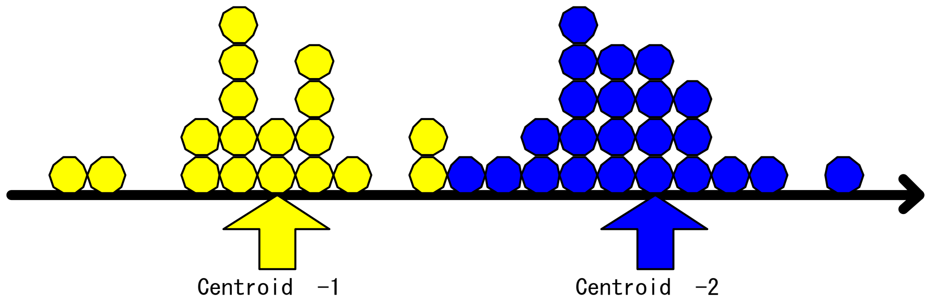
For example, we use four dimensional attributes. Point A has coordinate (0, 3, 4, 5) and point B (which is the centroid) has coordinate (7, 6, 3, -1).

The Euclidean Distance between point A and B is computed as

$$d_{BA} = \sqrt{(0-7)^2 + (3-6)^2 + (4-3)^2 + (5+1)^2}$$
$$= \sqrt{49+9+1+36} = 9.747$$

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?

Here is the principle of application of k-means clustering to machine learning or data mining:

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. Check the program code companion of this tutorial.
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.

What are the Difference of Supervised and Unsupervised Learning?

About	**Unsupervised learning**	**Supervised learning**
Other name	Cluster Analysis	Classification, Pattern Recognition
Training or learning period	• Object category is unknown • Rule of classification is given (generalized distance based)	Object category is known
Purposes of training	To know category of each object	To know the classification rule
After training (usage)	To classify object into a number of category	To classify object into a number of category
Example of Methods	K means clustering, hierarchical clustering, EM algorithm (Gaussian Mixture)	Discriminant analysis, K nearest neighbor, Decision tree, Multilayer Perceptron neural network

In unsupervised learning, the category of the object is unknown. However, we know the rule to classify (usually based on distance) and we also know the features (independent variables) that can describe the classification of the object. There is no training example to examine whether the classification is correct or not. Thus, the objects are assigned into groups merely based on the given rule.

In supervised learning, object groups and several training examples of objects that have been grouped are known. The model of classification is also given (for example, linear or quadratic) and we want to know the best fit parameters of the model that can best separate the objects based on the training samples.

The differences between unsupervised and supervised learning are only on the training session. After the parameters are determined, and we start to use the model, both models have the same usage to classify object into a number of category.

K means algorithm is one of the simplest partitions clustering method. More advanced algorithms related to k means are Expected Maximization (EM) algorithm especially Gaussian Mixture, Self-Organization Map (SOM) from Kohonen, Learning Vector Quantization (LVQ).

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 classical 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.
4. Check the latest update of resources of k means in http://people.revoledu.com/kardi/tutorial/kMean/index.html