

**CSC 323: MACHINE LEARNING**

**PERCEPTRON AND KMEANS ALGORITHMS REPORT**

Group work

K Means Algorithm

Explanation  
K means clustering algorithm was developed by J. MacQueen (1967) and was later improved by J. A. Hartigan and M. A. Wong around 1975. K means clustering algorithm to group objects based on attributes OR features into K number of groups. K is a positive integer. Grouping is done by minimizing the sum of squares of distances between a given data and the corresponding cluster centroid. It is one of the simplest partitioning clustering method.

Example  
Suppose we have 4 different medicines (A, B, C, D). Each medicine has 2 attributes (Weight index and pH).

|  |  |  |
| --- | --- | --- |
| **OBJECT** | Attribute1(x): weight Index | Attribute 2(y): pH |
| MEDICINE A | 1 | 1 |
| MEDICINE B | 2 | 1 |
| MEDICINE C | 4 | 3 |
| MEDICINE D | 5 | 4 |

Suppose we know beforehand that these objects belong to two clusters. (cluster 1 and 2).  
The problem now is to determine which medicines belong to which cluster.

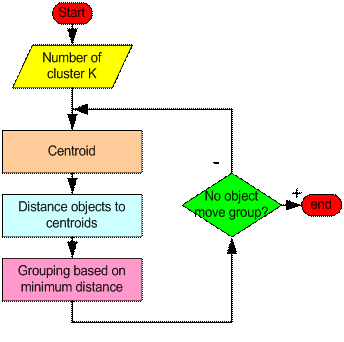
Pseudocode

***The basic steps of k-means clustering are as follows.***

* Determine number of clusters K.
* Pick the centroids (centre) of the clusters.
* Determine centroid’s coordinates for each cluster.
* Determine distance of each object in relation to the centroid.
* Group the object based on the minimum distance.

***\*Then the K means algorithm will repeat the last three steps until convergence.***

***k-means flowchart***



Exercise.

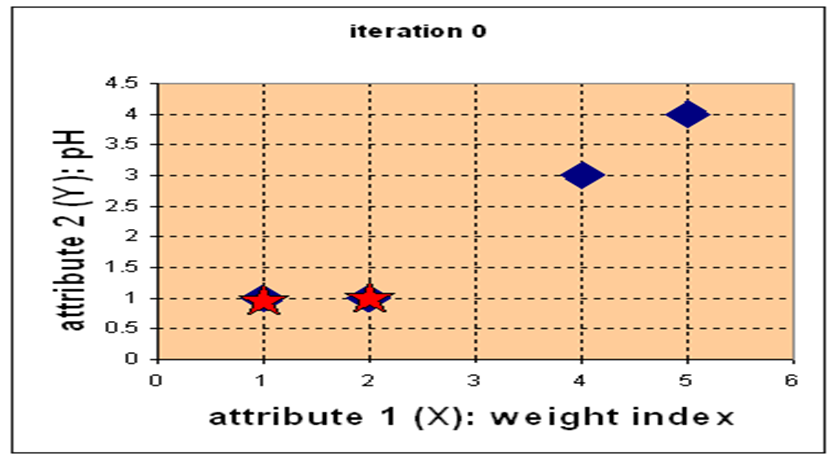
* Suppose we know before hand that these objects belong to two clusters (cluster 1 and cluster 2).
* The problem now is to determine which medicines belong to which cluster.

Solution

Taking the example through the Algorithm

1. Suppose our goal is to group these objects into 2 groups, based on the two features (pH and weight index) K=2.
2. Picking the centroids:

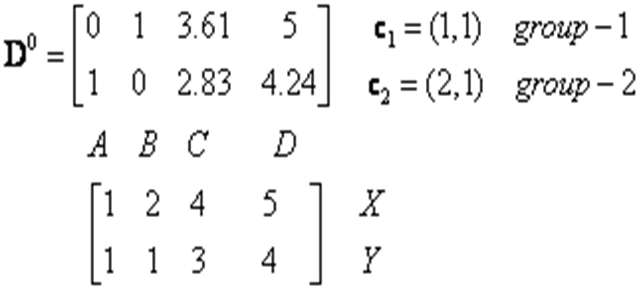
Initial value of centroids: Suppose we use medicine A and medicine B as the first centroids. Let C1 and C2 denote the coordinate of the centroids, thus C1 =(1,1) and C2 = (2,1).



1. Iterative stage

REPEAT the following steps until convergence.

* Determine the centroid’s coordinates for each cluster



* Determine Objects-to-Centroids distance (use Euclidean dist.)

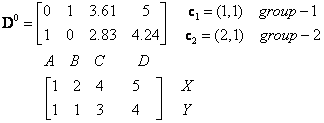
we calculate the distance between cluster centroid (C1 and C2 respectively) to each object ([1,1],[2,1], [4,3], and [5,4]).

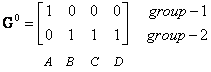
For example, distance from medicine C (4, 3) to the first centroid C1 is C:\Users\user\Desktop\Work\CSC 323\CSC 323_2015\2_Instance based Learning\K-Means_Clustering\NumericalExample_files\NumericalExample_clip_image014.gif, and its distance to the second centroid C2 is, .

***THIS RESULTS IN THE DISTANCE MATRIX, D0***

* 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.





Repeat the Iterative stage

* *Determine the NEW centroids* :

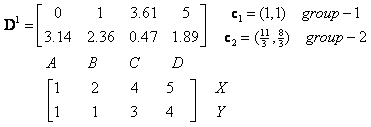
Knowing the members of each group, now we compute the new centroid of each group based on these new memberships.

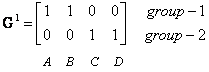
Group 1 only has one member thus the centroid remains as C1 (1,1)

Group 2 now has three members, thus the centroid is the average coordinate among the three members, (B[2,1], C[4,3] and D[5,4]: 

* Determine New Distance Matrix & New Group Matrix.

The new distance Matrix given the new centroids is



 Iterate until *stable* (= no object move group) i.e by Checking if an object changed its group status. Object B moves, hence ITERATE

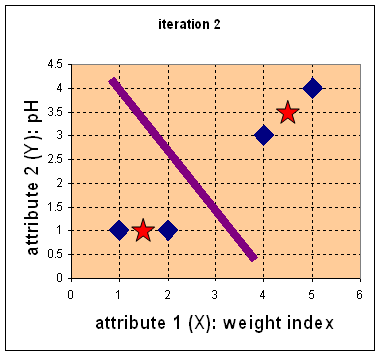
* Repeat the Iterative stage

*Determine the NEW 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 TWO members, A[1,1] and B[2,1],

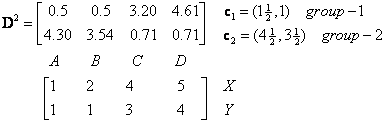
Group 2 now has TWO members, C[4,3] and D[5,4]:

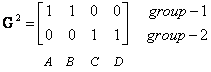
C:\Users\user\Desktop\Work\CSC 323\CSC 323_2015\2_Instance based Learning\K-Means_Clustering\NumericalExample_files\NumericalExample_clip_image026.gif

C:\Users\user\Desktop\Work\CSC 323\CSC 323_2015\2_Instance based Learning\K-Means_Clustering\NumericalExample_files\NumericalExample_clip_image102.gif 

* Determine New Distance Matrix & New Group Matrix

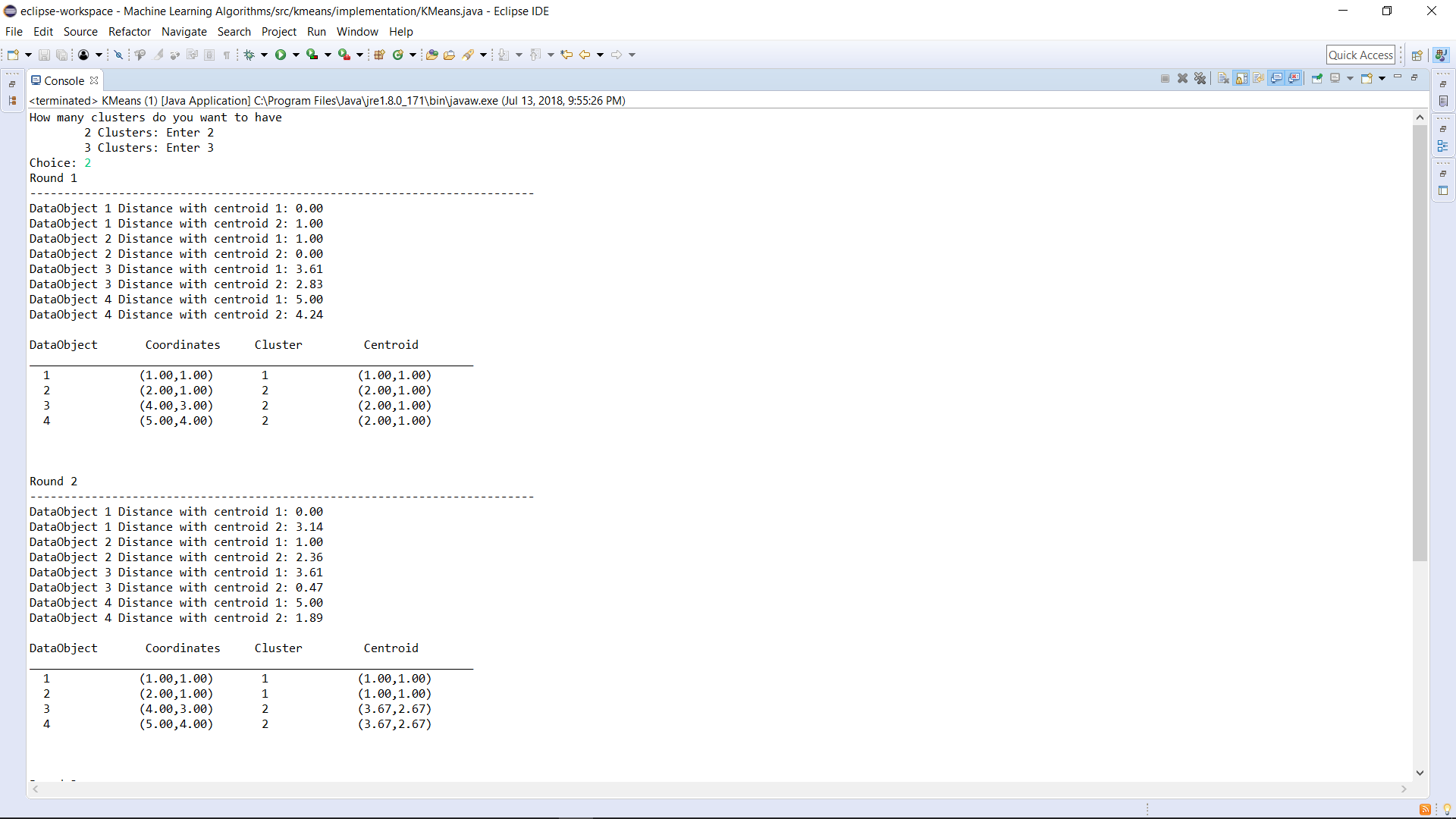
The new distance Matrix given the new centroids is,



Iterate until *stable* (= no object move group) i.e by Checking if an object changed its group status. NO Object has moved ie group status changed, **hence we conclude that A&B =gp 1 while C&D=gp 2.**

Results.(A screenshot of the K means output.)



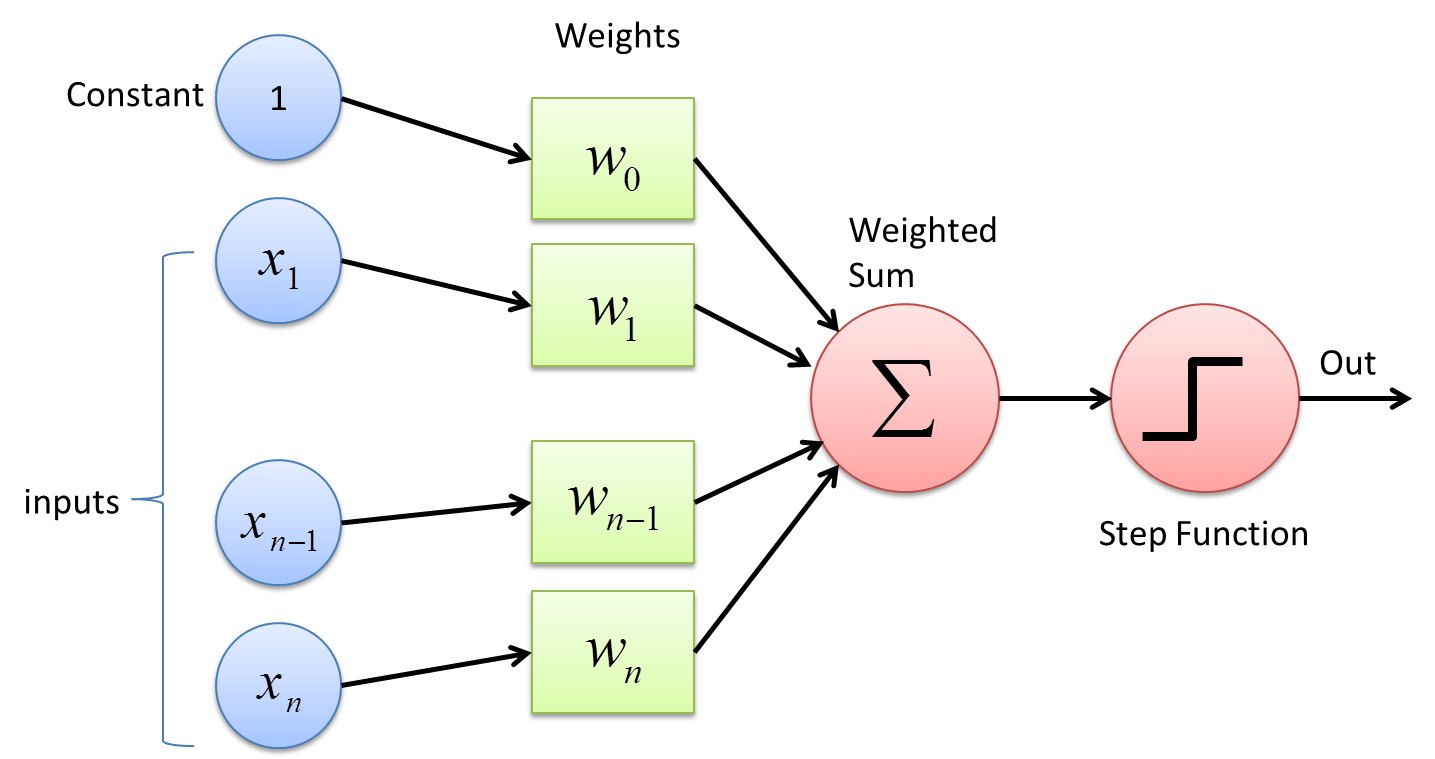
# Perceptron Algorithm.

A perceptron models a neuron i.e. it receives n inputs (corresponding to features) sums those inputs and produces an output.

Features of Perceptron Algorithm.

1. It is an algorithm for supervised learning of binary classifiers (functions that can decide whether an input, represented by a vector of numbers, belongs to some specific class or not). It thus maps its input ***X*** (a real-valued vector) to a function ***f(x)*** (a single binary value).
2. It is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector.

Perceptron Illustration.

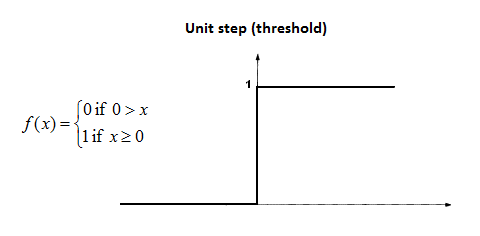


As illustrated above, the perceptron has 4 parts:

1. Input Value represented by ***X1, X2, …., Xn.***
2. Weights and Bias: The weights from the inputs to the neuron i.e. ***W0, W1, …., Wn*** and a bias (constant value 1) that allows us to shift the transfer function curve horizontally along the input areas while leaving the shape unaltered. Thus in graphical application: i) The Bias is the intercept.

ii)The Weights is the slope.

1. Net Sum (Weighted Sum): Calculated as: ***x1.w1*** + ***x2.w2*** + …. + ***xn.wn.***
2. Unit Step Activation Function:



**NOTE:** Perceptron only has two output values as illustrated above i.e. **0 or 1.**

Perceptron Learning Rule Pseudocode.

Suppose we have a set of learning samples consisting of an input vector x and a desired output d(x). For a classification task the d(x) is usually +1 or -1. The perceptron learning rule is very simple and can be stated as follows:

1. Start with random weights for the connections;
2. Select an input vector **X** from the set of training samples;
3. If **y ≠d(x)** (the perceptron gives an incorrect response), modify all connections **wi** according to:

**Δwi = d(x)xi;**

1. Go back to 2 if stopping condition (error value is 0) is not reached.

Code Explanation:

For each training instance (***X,*** Actual), compute Activation Output ***f(w, x).***

The Learning Rule used is:

Error = Output – Actual i.e.

***W(i+1) = Wi + (change in Wi).***

***(change in Wi) =* Learning Rate X (Target Value – Actual Value) X xi.**

Results of Implemented Perceptron Code in Java:

