AHSANULLAH UNIVERSITY OF SCIENCE AND TECHNOLOGY

Pattern Recognition Lab

Experiment Number 04

"Implementing K-Means Clustering"

Submitted by-
MD. SHAMIM TOWHID
ID: 12.02.04.097
Section: B2
Year: 4th Semester: 2nd
Date: 13-08-2016

Implementing K-Means Clustering

Md. Shamim Towhid
Computer Science and Engineering Department
Ahsanullah University of Science and Technology
Dhaka, Bangladesh
shamim.towhid@gmail.com

***Objectives*—the objective of this experiment is to understand one of the very popular clustering algorithm known as K-Means clustering algorithm. This is an unsupervised learning method which means the class label is unknown here. But to measure the performance of the algorithm we need to know the ground truth. Here in this experiment we will use cluster purity as performance measure of the classifier. Here we use the dataset that has 150 data with four dimension each. We will cluster the whole dataset into three cluster here, so in our experiment k=3.**

Keywords—unsupervised learning; clustering;K-means clustering; MATLAB code;

I. INTRODUCTION

K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids shoud 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 point is pending, the first step is completed and an early groupage is done. At this point we need to re-calculate k new centroids as barycenters 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.
The whole process can be showed by the following figure –

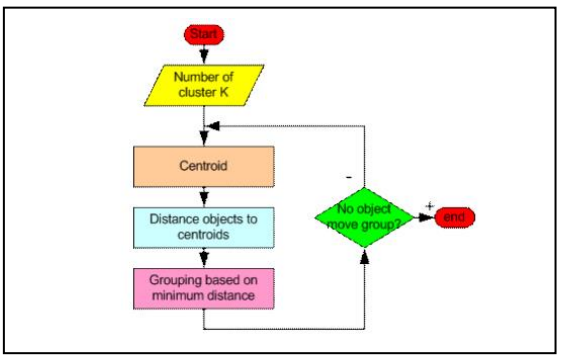

Figure 1 : Fowchart of the algorithm

II. IMPLEMENTATION

A. *Choosing the initial centroids*

Since in this experiment we choose to cluster the whole dataset in three cluster we have to choose three centroids. At first we choose it randomly but the centroid should be as far as possible so we fixed the range. The first centroid will be between 0 and 50 the second one will be between 50 and 100 and the third one will be between 101 and 150.

B. *Calculating the distance matrix*

Here in this experiment we calculate the Euclidian distance between the sample points and centroid in a 3 x 150 matrix. After that this distance matrix will be used to assign groups to each column of the distance matrix. Here we have 150 column so we will get 150 one in the group matrix but in different row. In which row it will be determined by the minimum distance stored in the distance matrix.
After assigning the group now we have to count that how many sample are in each group/cluster. Then we will calculate the new centroids for the next iteration.

C. *Calculating new centroid*

The new centroid will be all the data samples average which corresponding row has a 1 in the group matrix. Thus we will get new set of centroid. After that the whole procedure will continue until 20 times of the time each group has the same number of samples.
In our experiment the result is 50 61 39 most of the time. It will change sometimes cause the initial centroid are chosen randomly. So it is possible to get slight different output but the difference won't be very high.

III. RESULT ANALYSIS

One basic metric to measure how "good" the clustering in comparison to known class labels is called purity.
The mathematical definition of purity is as follows:

$$\text{purity}(\Omega, \mathbb{C}) = \frac{1}{N} \sum_k \max_j |\omega_k \cap c_j|$$

To compute purity , each cluster is assigned to the class which is most frequent in the cluster, and then the accuracy of this assignment is measured by counting the number of correctly assigned documents and dividing by N. where N is the total number of samples.
In our experiment the 150 dataset is actually divided into 50 50 and 50 members. Our result is 50 61 39.
So for the first cluster the cluster purity is 50/50 =1
For the second cluster it is 61/50 = 1.22
For the third cluster it it 39/50 = 0.78
So the total purity is (1+1.22+0.78) = 3

So what does this tell us? A cluster is considered to be "pure" if it has a purity of 1 since that indicates that all of the instances in that cluster are of the same label. This means our original label classification was pretty good and that our Kmeans did a pretty good job. The "best" purity score for an entire data set would be equal to the original K-number of clusters since that would imply that every cluster has an individual purity score of 1.

MATLAB CODE:

```
load irisdataset.txt;
dataset=irisdataset;

%random number generation
a=[1 51 103];
b=[50 98 150];
r1 = round((b(1)-a(1))*rand(1)+a(1));
r2 = round((b(2)-a(2))*rand(1)+a(2));
r3 = round((b(3)-a(3))*rand(1)+a(3));
disp(r1);
disp(r2);
disp(r3);
centroid1 = dataset(r1,:);
centroid2 = dataset(r2,:);
centroid3 = dataset(r3,:);

%distance matrix calculation
for m=1:3
    for n=1:150
        if(m==1)
            v=(dataset(n,:)-
centroid1).^2;

d(m,n)=sqrt(v(1)+v(2)+v(3)+v(4));
        elseif(m==2)
            v=(dataset(n,:)-
centroid2).^2;
d(m,n)=sqrt(v(1)+v(2)+v(3)+v(4));
        elseif(m==3)
            v=(dataset(n,:)-
centroid3).^2;

d(m,n)=sqrt(v(1)+v(2)+v(3)+v(4));
        end
    end
end

%group matrix calculation
for m=1:3
    for n=1:150
        if(d(m,n)==min(d(:,n)))
            g(m,n)=1;
        else g(m,n)=0;
        end
    end
end



a=[0 0 0 0];
newcentroid=[0 0 0 0;0 0 0 0;0 0 0 0];
counter=0;
previous=[0 0 0];
flag=true;
boundarycounter=0;



while flag

%centroid calculation
for m=1:3
   for n=1:150
       if(g(m,n)==1)
           counter=counter+1;
           a=a+dataset(n,:);
       end
   end

   if(counter ~= 0)
       a=a/counter;
   else newcentroid(m,:)=a;
   end
   %check for loop termination
   groupcounter(m)=counter;
   %new centroid
   newcentroid(m,:)=a;

   %re initilize for the next row
calculation
   counter=0;
   a=[0 0 0 0];
```

```
end

%check for loop termination
if(groupcounter==previous)
    boundarycounter=boundarycounter+1;
    if(boundarycounter==20)
        flag=false;
    end
end
previous=groupcounter;

%distance calculation
for m=1:3
    for n=1:150
            v=(dataset(n,:)-
newcentroid(m,:)).^2;

d(m,n)=sqrt(v(1)+v(2)+v(3)+v(4));
    end
end
%group matrix calculation
for m=1:3
    for n=1:150
        if(d(m,n)==min(d(:,n)))
            g(m,n)=1;
        else g(m,n)=0;
        end
    end
end


end

disp(groupcounter);
```

IV. Conclusion

In this experiment we explore the K-means clustering algorithm. This is a very popular and widely used clustering technique. In this algorithm we need to know the ground truth if we want to measure the performance with cluster purity. This is the first unsupervised technique we implement in MATLAB.