Course No.: **ELEN-857**
Course Title: **Advanced Pattern Recognition Method**
Department: **Electrical and Computer Engineering**

Project 2: K-means clustering algorithm

Submitted To:
Dr. Robert Y.Li, Professor
Department of Electrical Engineering
Telephone: (336) 285-3716; E-mail: eeli@ncat.edu

Prepared By:
Name: Mrinmoy Sarkar
Banner Id: 950-363-260
E-mail: msarkar@aggies.ncat.edu

Contents:

1. Abstract
2. Technical Description
3. Results
4. Summary
5. Appendix

1. Abstract:

The main purpose of the project is to apply K-means clustering algorithm to Fisher's Iris data. Fisher's Iris data contains a set of measurements related to 3 species of the Iris plant. The three species are Iris Setosa, Iris Versicolor, and Iris Virginica.

2. Technical Description:

The dataset contains 50 plants from each of the 3 species. There are 4 features in the dataset named sepal length, sepal width, petal length and petal width. MATLAB programming language is used to implement the K-means algorithm. Three different K (2,3,4) and two different thresholds T (0.01, 0.1) are used to cluster the 150 data samples. And corresponding confusion matrix is calculated.

3. Results:

 # (K = 2, T = 0.01) cluster A: 53 cluster B: 97

Initial centers: Z_1 = (5.1000 3.5000 1.4000 0.2000),
Z_2 = (7.0000 3.2000 4.7000 1.4000)

Final centers: Z_1 = (5.0057 3.3604 1.5623 0.2887),
Z_2 = (6.3010 2.8866 4.9588 1.6959)

Confusion Matrix:

	Iris Setosa	Iris Versicolor	Iris Virginica
A	50	3	0
B	0	47	50

 # (K = 2, T = 0.10) cluster A: 53 cluster B: 97

Initial centers: Z_1 = (5.1000 3.5000 1.4000 0.2000),
Z_2 = (7.0000 3.2000 4.7000 1.4000)

Final centers: Z_1 = (5.0057 3.3604 1.5623 0.2887),
Z_2 = (6.3010 2.8866 4.9588 1.6959)

Confusion Matrix:

	Iris Setosa	Iris Versicolor	Iris Virginica
A	50	3	0
B	0	47	50

 # (K = 3, T = 0.01) cluster A: 50 cluster B: 62 cluster C: 38

Initial centers: $Z_1 = (5.1000 \quad 3.5000 \quad 1.4000 \quad 0.2000)$,
$Z_2 = (7.0000 \quad 3.2000 \quad 4.7000 \quad 1.4000)$,
$Z_3 = (6.3000 \quad 3.3000 \quad 6.0000 \quad 2.5000)$

Final centers: $Z_1 = (5.0060 \quad 3.4180 \quad 1.4640 \quad 0.2440)$,
$Z_2 = (5.9016 \quad 2.7484 \quad 4.3935 \quad 1.4339)$,
$Z_3 = (6.8500 \quad 3.0737 \quad 5.7421 \quad 2.0711)$

Confusion Matrix:

	Iris Setosa	Iris Versicolor	Iris Virginica
A	50	0	0
B	0	48	14
C	0	2	36

 # (K = 3, T = 0.10) cluster A: 50 cluster B: 62 cluster C: 38

Initial centers: $Z_1 = (5.1000 \quad 3.5000 \quad 1.4000 \quad 0.2000)$,
$Z_2 = (7.0000 \quad 3.2000 \quad 4.7000 \quad 1.4000)$,
$Z_3 = (6.3000 \quad 3.3000 \quad 6.0000 \quad 2.5000)$

Final centers: $Z_1 = (5.0060 \quad 3.4180 \quad 1.4640 \quad 0.2440)$,
$Z_2 = (5.9194 \quad 2.7532 \quad 4.3903 \quad 1.4194)$,
$Z_3 = (6.8211 \quad 3.0658 \quad 5.7474 \quad 2.0947)$

Confusion Matrix:

	Iris Setosa	Iris Versicolor	Iris Virginica
A	50	0	0
B	0	48	14
C	0	2	36

 # (K = 4, T = 0.01) cluster A: 50 cluster B: 40 cluster C: 32 cluster D: 28

Initial centers: $Z_1 = (5.1000 \quad 3.5000 \quad 1.4000 \quad 0.2000)$,
$Z_2 = (7.0000 \quad 3.2000 \quad 4.7000 \quad 1.4000)$,
$Z_3 = (6.3000 \quad 3.3000 \quad 6.0000 \quad 2.5000)$,
$Z_4 = (5.8000 \quad 2.7000 \quad 5.1000 \quad 1.9000)$

Final centers: $Z_1 = (5.0060 \quad 3.4180 \quad 1.4640 \quad 0.2440)$,
$Z_2 = (6.2525 \quad 2.8550 \quad 4.8150 \quad 1.6250)$,
$Z_3 = (6.9125 \quad 3.1000 \quad 5.8469 \quad 2.1312)$,

Z_4 = (5.5321   2.6357   3.9607   1.2286)

Confusion Matrix:

	Iris Setosa	Iris Versicolor	Iris Virginica
A	50	0	0
B	0	23	17
C	0	0	32
D	0	27	1

 # (K = 4, T = 0.10) cluster A: 50 cluster B: 40 cluster C: 32 cluster D: 28

Initial centers: Z_1 = (5.1000   3.5000   1.4000   0.2000),
Z_2 = (7.0000   3.2000   4.7000   1.4000),
Z_3 = (6.3000   3.3000   6.0000   2.5000),
Z_4 = (5.8000   2.7000   5.1000   1.9000)

Final centers: Z_1 = (5.0060   3.4180   1.4640   0.2440),
Z_2 = (6.2541   2.8865   4.8486   1.6459),
Z_3 = (6.9125   3.1000   5.8469   2.1312),
Z_4 = (5.6000   2.6194   4.0032   1.2419)

Confusion Matrix:

	Iris Setosa	Iris Versicolor	Iris Virginica
A	50	0	0
B	0	23	17
C	0	0	32
D	0	27	1

4. Summary:

- For different values of K and T, we see that **Iris Setosa** always belongs to cluster **A**.
- As there exists overlapping data among classes, so K-means algorithm cannot separate all the data into three different clusters even though K is set to 3.
- If T is very low, the K-means algorithm takes more iteration to converse.
- Whatever the value of K, K-means algorithm always converses.

- For both values of T, K-means algorithm outputs the same result.
- The initial centers play an important role in correct clustering for k-means algorithms.

6. Appendix

MATLAB Code:

```
%% file name project2.m
% author: Mrinmoy Sarkar
% email: msarkar@aggies.ncat.edu
% date: 10/6/2017


clear;
close all;

% load data to a veriable
data = importdata('iris.txt');

% no. of class is 3 named Iris-setosa, Iris-versicolor and Iris-verginica
% there are 4 attributes named sepal-length, sepal-width, petal-length,
% petal-width
% there are 50 plants for each species

irisSetosa = zeros(50,4);
irisVersicolor = zeros(50,4);
irisVerginica = zeros(50,4);

n = size(data,1);

indxSeto = 1;
indxVers = 1;
indxVerg = 1;

for i=2:n
    x = strsplit(cell2mat(data(i)));
    if strcmp(x(5), 'Iris-setosa')
        for j=1:4
            irisSetosa(indxSeto,j) = str2double(cell2mat(x(j)));
        end
        indxSeto = indxSeto + 1;
    elseif strcmp(x(5), 'Iris-versicolor')
        for j=1:4
            irisVersicolor(indxVers,j) = str2double(cell2mat(x(j)));
        end
        indxVers = indxVers + 1;
    elseif strcmp(x(5), 'Iris-virginica')
        for j=1:4
            irisVerginica(indxVerg,j) = str2double(cell2mat(x(j)));
        end
        indxVerg = indxVerg + 1;
    end
end
```

```
X_true = {irisSetosa, irisVersicolor, irisVerginica};
X = [irisSetosa; irisVersicolor; irisVerginica];


%% K-means algorithms
noOfTrueClasses = 3;
trueA = array2table(X(1:50,:));
trueB = array2table(X(51:100,:));
trueC = array2table(X(101:150,:));
trueClasses = {trueA, trueB, trueC};

X = X';
Z_init = [5.1 3.5 1.4 0.2;...
          7.0 3.2 4.7 1.4;...
          6.3 3.3 6.0 2.5;...
          5.8 2.7 5.1 1.9]';

K = [2 3 4];
T = [0.01 0.1];
for i=1:length(K)
    for j=1:length(T)
        [z,classes] = kmeanAlgorithm(X,K(i),Z_init(:,1:K(i)),T(j));
        disp('Initial cluster centers:');
        disp((Z_init(:,1:K(i)))');
        disp('Final cluster centers:');
        disp(z');
        fprintf('#(K = %d, T = %0.2f) ',K(i),T(j));
        for cl = 1:K(i)
            fprintf('cluster %d: %d ', cl , size(classes{cl},2))
        end
        fprintf('\n')
        confusionMat = zeros(noOfTrueClasses, K(i));
        for m = 1:noOfTrueClasses
            for n = 1:K(i)
                predictedData = (classes{n})';
                count = 0;
                for p=1:size(predictedData,1)
                    g =
intersect(trueClasses{m},array2table(predictedData(p,:)));
                    if ~isempty(g)
                        count = count + 1;
                    end
                end
                confusionMat(m,n) = count;
            end
        end
        % print confusion matrix
        fprintf('Confusion Matrix:\n');
        tc = 'ABC';
        for c = 1:1:size(confusionMat,2)
                fprintf('  | cluster %d ',c);
        end
        dasLine ={'\n----------------------------\n',...
                  '\n------------------------------------------\n',...
                  '\n-------------------------------------------------------
\n'};
```

```
        fprintf(dasLine{i});
        for r = 1:size(confusionMat,1)
            fprintf('%c ',tc(r));
            for c = 1:1:size(confusionMat,2)
                fprintf('|      %2d     ',confusionMat(r,c));
            end
            fprintf(dasLine{i})
        end
    end
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [z,classes] = kmeanAlgorithm(x,k,z,T)
classes = cell(1,k);
for i=1:k
    classes{1,i}=[];
end
iterationNo = 1;
while 1
    %fprintf('Iteration Number : %d\n', iterationNo);
    for i=1:size(x,2)
        temp = ones(size(z)).*x(:,i);
        [m mi] = min(sum((z-temp).^2));
        classes{1,mi} = [classes{1,mi} x(:,i)];
    end
    zNew = zeros(size(z));
    for i=1:k
        temp = classes{1,i};
        zNew(:,i) = (1/size(temp,2))*sum(temp,2);
    end
    if sum(sum(abs(z-zNew)> T)) == 0
        break;
    else
        z=zNew;
    end
    for i=1:k
        classes{1,i}=[];
    end
    iterationNo = iterationNo + 1;
end
fprintf('Iteration Number : %d\n', iterationNo);
end
```

<u>%%</u>