CSE 6740: Computational Data Analysis
Assignment #2

Due on Thursday, October 17, 2019

Shahrokh Shahi
(GT Account: sshahi3)

Q4 – Programming: Text Clustering

The following is the implementation of mycluster in MATLAB. The names of the variables and indices are chosen to match the assignment description.

```
function [ class ] = mycluster( bow, K )
%
% Your goal of this assignment is implementing your own text clustering algo.
%
% Input:
%     bow: data set. Bag of words representation of text document as
%     described in the assignment.
%
%     K: the number of desired topics/clusters.
%
% Output:
%     class: the assignment of each topic. The
%     assignment should be 1, 2, 3, etc.
%
% For submission, you need to code your own implementation without using
% any existing libraries

% YOUR IMPLEMENTATION SHOULD START HERE!

% hard-coded parameters
MAX_IT = 200;
EPS    = 100 * eps;

% input parameters
[nDocs, nWords] = size(bow);
nClusters = K;

% initializing the mixture coefficient p(c)= \pi_c
pi_c = rand(nClusters, 1);
pi_c = pi_c ./ sum(pi_c);   % normalizing

% initialization \mu
mu = rand(nWords, nClusters);
mu = mu ./ sum(mu);
% mu = mu ./ repmat(sum(mu), nWords, 1);

% initializing \gamma
gamma = zeros(nDocs, nClusters);
gamma_prev = gamma;



% iterations
for iter = 1 : MAX_IT
    % ------------------------------ E-step ------------------------------ %
    % p(Di) = sum(p(Di|c)p(c))
    p_Di = zeros(nDocs, 1);
    p_Di_c = ones(nDocs, nClusters);

    for i = 1 : nDocs
        for c = 1 : nClusters
            for j = 1 : nWords
                p_Di_c(i,c) = p_Di_c(i,c) * mu(j,c) ^ bow(i,j);
            end
            p_Di(i) = p_Di(i) + p_Di_c(i,c) * pi_c(c);
        end

        for c = 1 : nClusters
            gamma(i,c) = pi_c(c) * p_Di_c(i,c) / p_Di(i);
        end
    end
```

```
    % ------------------------------ M-step ------------------------------ %
    % mu = X / Y:
    X = (gamma' * bow)';
    Y  = zeros(1, nClusters);
    for c = 1 : nClusters
        for i = 1 : nDocs
            for l = 1 : nWords
                Y(c) = Y(c) + gamma(i,c) * bow(i,l);
            end
        end
    end
    % updating mu
    mu = X ./ repmat(Y, nWords, 1);

    % updating p(c)
    pi_c = sum(gamma) ./ nDocs;

    % ------------------------------------------------------------------- %
    % checking convergency
    % the convergency check is currently disabled, since it will be
    % converged quickly (usually within <5 iterations. But it can easily
    % be used by uncommenting the following lines:

    % if sum(sum(gamma-gamma_prev)) < EPS
    %    break
    % end
    % gamma_prev = gamma;
    % ------------------------------------------------------------------- %
end
% fprintf('clustering converged at iteration = %3d\n',iter);

% class indices (the index of maximums)
[~, class] = max(gamma,[],2);

end
```

After running homework.m several times, the following accuracy values are obtained:

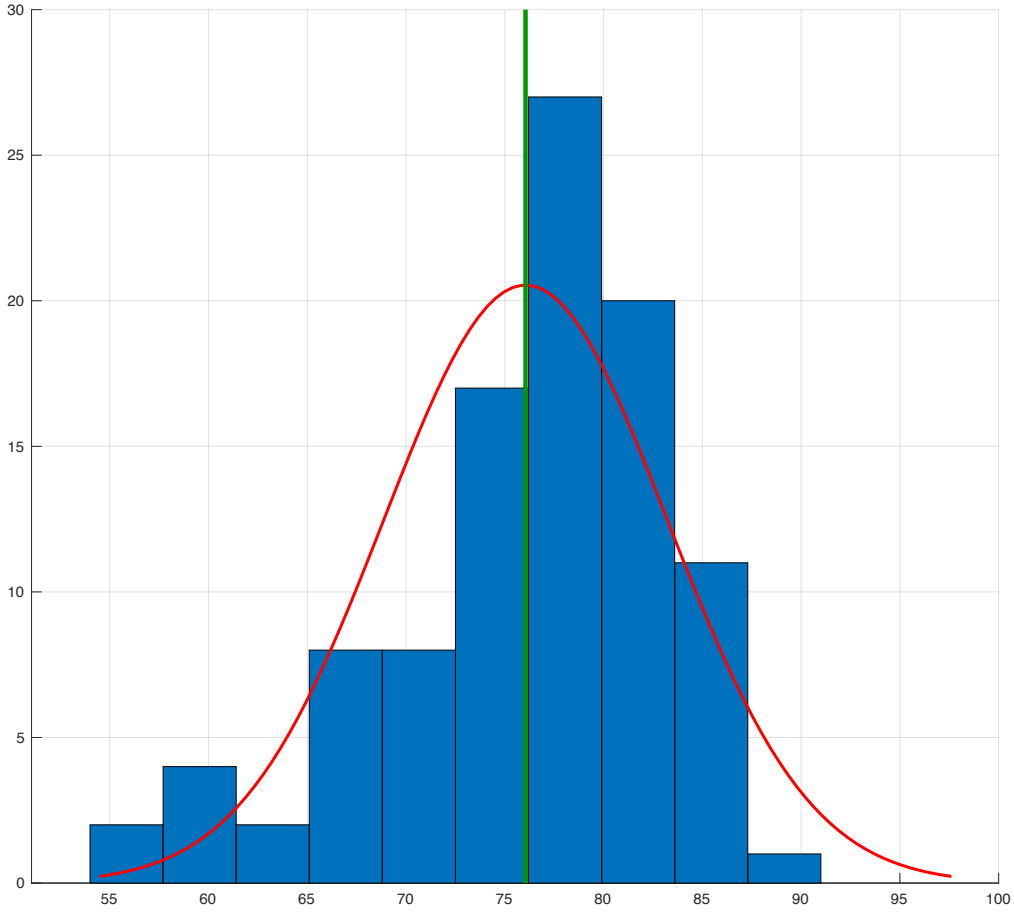
```
>> homework2
acc =
   77.5000

>> homework2
acc =
    79

>> homework2
acc =
   87.2500

>> homework2
acc =
   78.5000

>> homework2
acc =
   73.7500

>> homework2
acc =
   74.2500
```

```
>> homework2
acc =
   84.7500

>> homework2
acc =
   79.2500

>> homework2
acc =
   75.2500

>> homework2
acc =
   88.5000

>> homework2
acc =
   73.7500
```

Although it is not a requirement in the homework, we can run the procedure for several times, say 100, and draw the accuracy results by running the following code. This code is also submitted (testRun.m)

```
%% MY TEST SUIT
%  Developed by: Shahrokh Shahi (sshahi3)
%  I wrote this simple code to check the outputs of my clustering function

%% Initialization
clc
clear
close all

%% Hard-coded Values & Loading Data
MAX_IT = 100;

load('data');
T = X(:,1:100);
label = X(:,101);

%% Run Loop
acc = zeros(MAX_IT, 1);

for iter = 1 : MAX_IT
    index     = mycluster(T,4);
    acc(iter) = AccMeasure(label,index);
end

%% Plot Outputs
figure(1);
clf;
hold on;
grid on;
norm=histfit(acc,floor(MAX_IT/10),'normal');
[mean, var] = normfit(acc)
line([mean, mean], ylim, 'Color', [0, .6, 0], 'LineWidth', 3);
```

The accuracy values are plotted in the following figure which has a normal shape (as expected). We can also fit a normal distribution on the obtained values:



The mean of the accuracy for 100 runs is $\overline{acc} = 76.047$ with $\sigma = 7.187$