Muhammad Haris Rehman Reg. No. 182797 NUST, SEECS, BEE-8

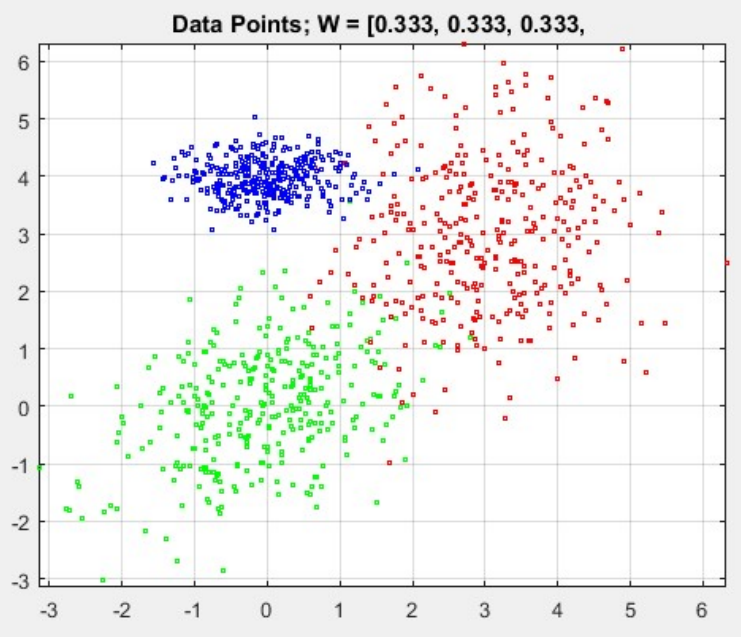
Machine Learning: Assignment # 3

Task 1

Matlab Code:

```
totSamp = 1000; %Total no. of data samples from all mixture components
k_True = 3; % No. of Gaussian Comps
mu1 = [0 0];
Sigma1 = [1 0.4; 0.4 1];
mu2 = [3 3];
Sigma2 = [1 0; 0 2];
mu3 = [0 4];
Sigma3 = [0.4 0; 0 0.1];

W = [1/3, 1/3, 1/3];
if abs(sum(W)-1)>=1e-10
    error('Weight vector must sum to 1');
end
%rng(1);
S1 = mvnrnd(mu1,Sigma1,round(W(1)*totSamp));
S2 = mvnrnd(mu2,Sigma2,round(W(2)*totSamp));
S3 = mvnrnd(mu3,Sigma3,round(W(3)*totSamp));
X = [S1; S2; S3];
figure;
%subplot(121);
plot(S1(:,1),S1(:,2),'go','MarkerSize',2); hold on; grid on;
plot(S2(:,1),S2(:,2),'ro','MarkerSize',2);
plot(S3(:,1),S3(:,2),'bo','MarkerSize',2);
xlim([min(X(:)) max(X(:))]); % Make axes have the same scale
ylim([min(X(:)) max(X(:))]);
%axis([-4 7 -8 8]);
title(sprintf('Data Points; W = [%0.3f, %0.3f, %0.3f, %0.3f, %0.3f]'...
              ,sort(W,'descend')));
```

Output:



Task 2 & 3

Matlab Code:

```
totSamp = 1000; %Total no. of data samples from all mixture components
k_True = 3; % No. of Gaussian Comps
mu1 = [0 0];
Sigma1 = [1 0.4; 0.4 1];
mu2 = [3 3];
Sigma2 = [1 0; 0 2];
mu3 = [0 4];
Sigma3 = [0.4 0; 0 0.1];

W = [1/3, 1/3, 1/3];
if abs(sum(W)-1)>=1e-10
    error('Weight vector must sum to 1');
end
%rng(1);
S1 = mvnrnd(mu1,Sigma1,round(W(1)*totSamp));
S2 = mvnrnd(mu2,Sigma2,round(W(2)*totSamp));
S3 = mvnrnd(mu3,Sigma3,round(W(3)*totSamp));
X = [S1; S2; S3];
figure;
%subplot(121);
plot(S1(:,1),S1(:,2),'go','MarkerSize',2); hold on; grid on;
plot(S2(:,1),S2(:,2),'ro','MarkerSize',2);
plot(S3(:,1),S3(:,2),'bo','MarkerSize',2);
xlim([min(X(:)) max(X(:))]); % Make axes have the same scale
ylim([min(X(:)) max(X(:))]);
%axis([-4 7 -8 8]);
title(sprintf('Data Points; W = [%0.3f, %0.3f, %0.3f, %0.3f, %0.3f]'...
              ,sort(W,'descend')));

%% Fit GMM
K = [3];
BIC = zeros(length(K),1);
for iter = 1:length(K)
    k = K(iter);
    %% Run k-means
    [Priors, Mu, Sigma] = EM_init_kmeans(X', k);
    Mu = Mu'; % Matlab gmm.fit requires that Mu by KxD
    Priors, Mu, Sigma
end
```

Output:

```
Priors =
    0.3604    0.2893    0.3504

Mu =
    0.0647    3.9581
    3.1020    3.1046
    0.1352    0.0300

Sigma(:,:,1) =
    0.4775   -0.0329
   -0.0329    0.2040

Sigma(:,:,2) =
    0.8617    0.1485
```

```
    0.1485    1.5757

Sigma(:,:,3) =
    1.0809    0.3817
    0.3817    0.9762
```

Comment:

From the above results, it can be seen that the mean and covariance matrices are not exactly same as from the given data is generated, but they are approximately same.

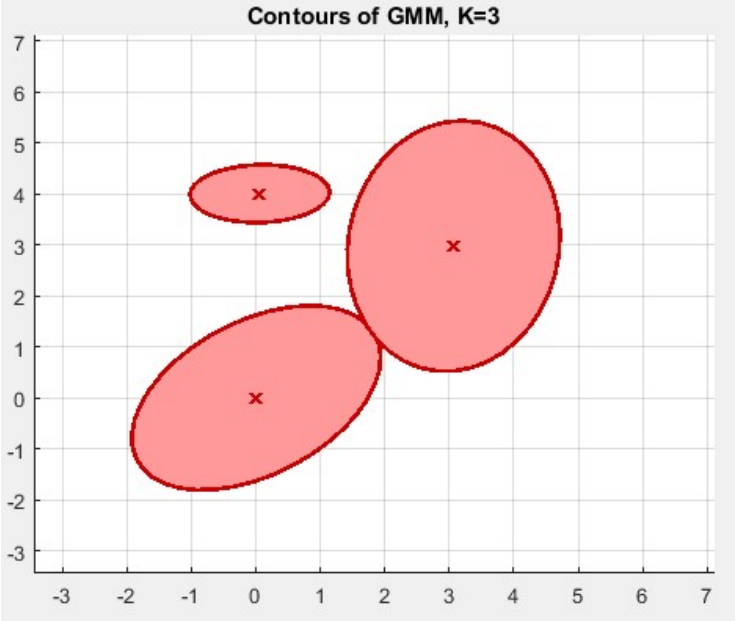
Task 4

Matlab Code:

```
totSamp = 1000; %Total no. of data samples from all mixture components
k_True = 3; % No. of Gaussian Comps
mu1 = [0 0];
Sigma1 = [1 0.4; 0.4 1];
mu2 = [3 3];
Sigma2 = [1 0; 0 2];
mu3 = [0 4];
Sigma3 = [0.4 0; 0 0.1];

W = [1/3, 1/3, 1/3];
if abs(sum(W)-1)>=1e-10
    error('Weight vector must sum to 1');
end
%rng(1);
S1 = mvnrnd(mu1,Sigma1,round(W(1)*totSamp));
S2 = mvnrnd(mu2,Sigma2,round(W(2)*totSamp));
S3 = mvnrnd(mu3,Sigma3,round(W(3)*totSamp));
X = [S1; S2; S3];
figure;
%subplot(121);
plot(S1(:,1),S1(:,2),'go','MarkerSize',2); hold on; grid on;
plot(S2(:,1),S2(:,2),'ro','MarkerSize',2);
plot(S3(:,1),S3(:,2),'bo','MarkerSize',2);
xlim([min(X(:)) max(X(:))]); % Make axes have the same scale
ylim([min(X(:)) max(X(:))]);
%axis([-4 7 -8 8]);
title(sprintf('Data Points; W = [%0.3f, %0.3f, %0.3f, %0.3f, %0.3f]'...
              ,sort(W,'descend')));

%% Fit GMM
K = [3];
BIC = zeros(length(K),1);
for iter = 1:length(K)
    k = K(iter);
    %% Run k-means
    [Priors, Mu, Sigma] = EM_init_kmeans(X', k);
    Mu = Mu'; % Matlab gmm.fit requires that Mu by KxD
    %% Fit GMM. Use k-mean op for initialization
    s = struct('mu',Mu,'Sigma',Sigma,'PComponents',Priors);
    options = statset('Display','final');
    GMModel = gmdistribution.fit(X,k,'Options',options,'Start',s);
    BIC(iter) = GMModel.BIC;

    %% Plot Current GMM
    muModel = GMModel.mu;
    SigModel = GMModel.Sigma;
```

```
    Wmodel = GMModel.ComponentProportion;
    figure; hold on; grid on;
    plotGMM(muModel', SigModel, [.8 0 0], 1);
    title(sprintf('Contours of GMM, K=%d',k));
    xlim([min(X(:)) max(X(:))]); % Make axes have the same scale
    ylim([min(X(:)) max(X(:))]);
end
```

Output:

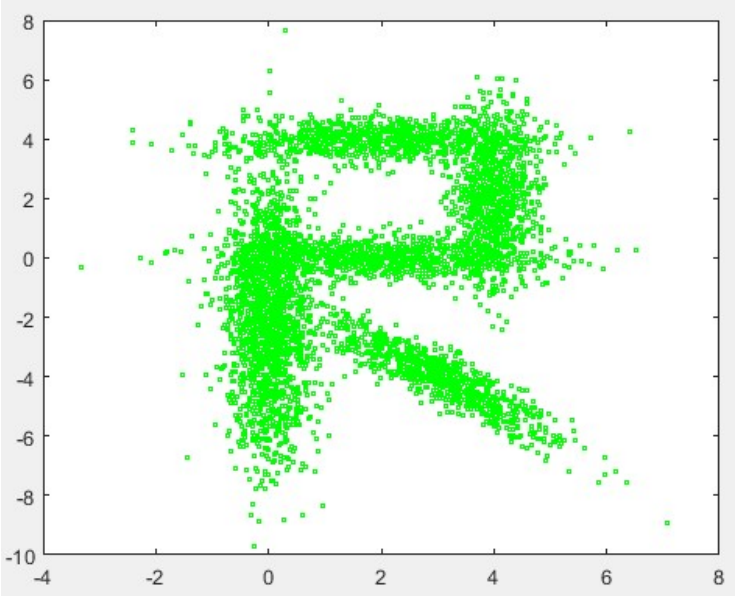


Task 5

Matlab Code:

```
load Lab-07
plot(X(:,1),X(:,2),'go','MarkerSize',2);
```

Output:



Comment:

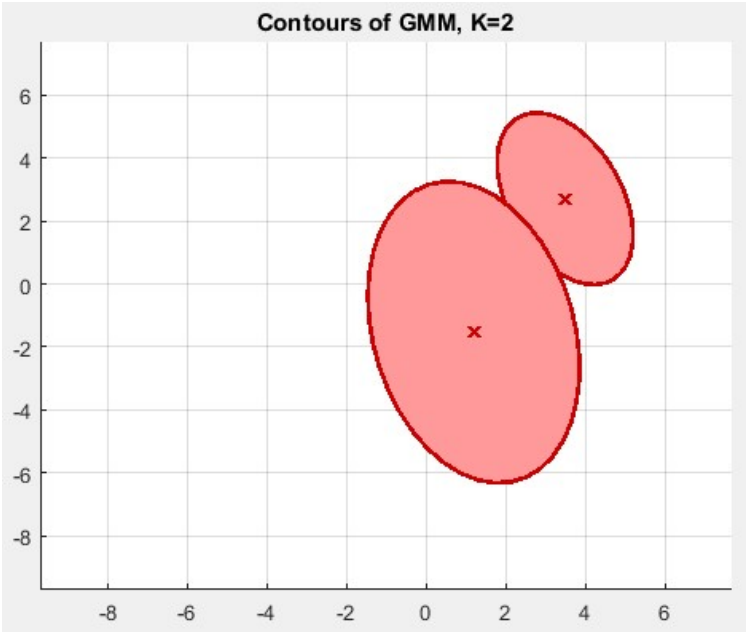
By looking at the plot above, we can see that, there can be 5 Gaussians fitted in the given data mixtures.

Task 6

Matlab Code:

```
load Lab-07
plot(X(:,1),X(:,2),'go','MarkerSize',2);

%% Fit GMM
K = [2];
BIC = zeros(length(K),1);
for iter = 1:length(K)
    k = K(iter);
    %% Run k-means
    [Priors, Mu, Sigma] = EM_init_kmeans(X', k);
    Mu = Mu'; % Matlab gmm.fit requires that Mu by KxD
    %% Fit GMM. Use k-mean op for initialization
    s = struct('mu',Mu,'Sigma',Sigma,'PComponents',Priors);
    options = statset('Display','final');
    GMModel = gmdistribution.fit(X,k,'Options',options,'Start',s);
    BIC(iter) = GMModel.BIC;

    %% Plot Current GMM
    muModel = GMModel.mu;
    SigModel = GMModel.Sigma;
    Wmodel = GMModel.ComponentProportion;
    figure; hold on; grid on;
    plotGMM(muModel', SigModel, [.8 0 0], 1);
    title(sprintf('Contours of GMM, K=%d',k));
    xlim([min(X(:)) max(X(:))]); % Make axes have the same scale
    ylim([min(X(:)) max(X(:))]);
end
```

Output:



Task 7

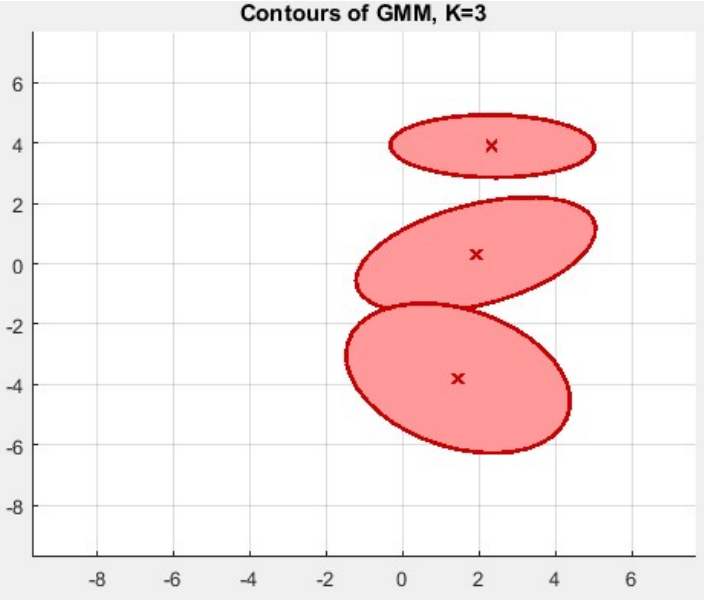
Matlab Code:

```
load Lab-07
plot(X(:,1),X(:,2),'go','MarkerSize',2);
```

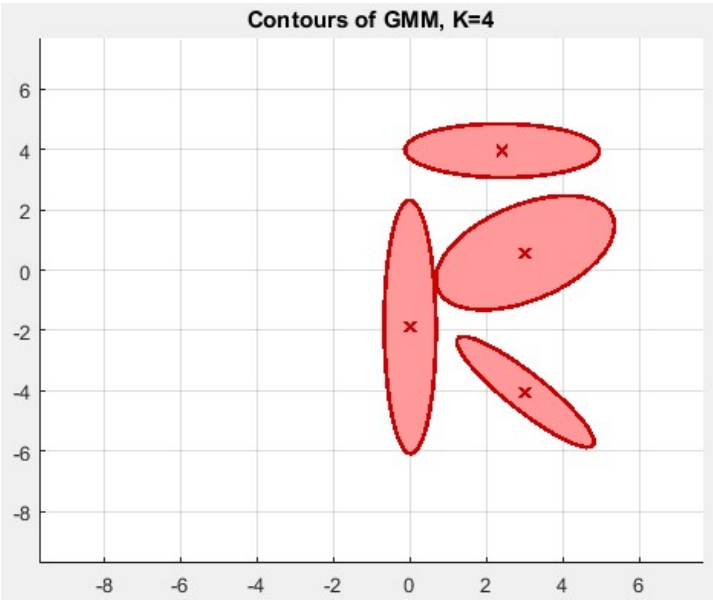
```
%% Fit GMM
K = [3, 4, 5, 6, 7, 8, 9];
BIC = zeros(length(K),1);
for iter = 1:length(K)
    k = K(iter);
    %% Run k-means
    [Priors, Mu, Sigma] = EM_init_kmeans(X', k);
    Mu = Mu'; % Matlab gmm.fit requires that Mu by KxD
    %% Fit GMM. Use k-mean op for initialization
    s = struct('mu',Mu,'Sigma',Sigma,'PComponents',Priors);
    options = statset('Display','final');
    GMModel = gmdistribution.fit(X,k,'Options',options,'Start',s);
    BIC(iter) = GMModel.BIC;

    %% Plot Current GMM
    muModel = GMModel.mu;
    SigModel = GMModel.Sigma;
    Wmodel = GMModel.ComponentProportion;
    figure; hold on; grid on;
    plotGMM(muModel', SigModel, [.8 0 0], 1);
    title(sprintf('Contours of GMM, K=%d',k));
    xlim([min(X(:)) max(X(:))]); % Make axes have the same scale
    ylim([min(X(:)) max(X(:))]);
end
```

Output:

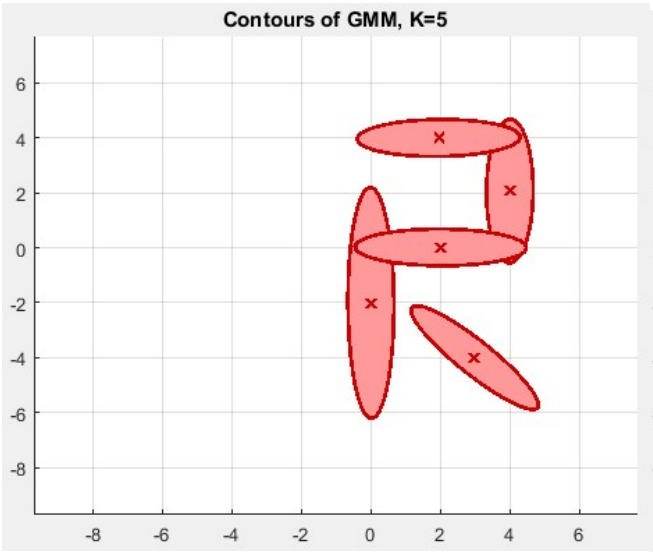
K = 3



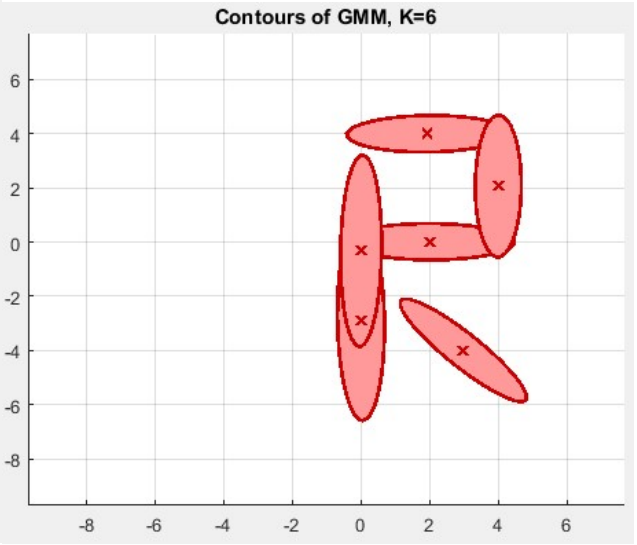
K = 4



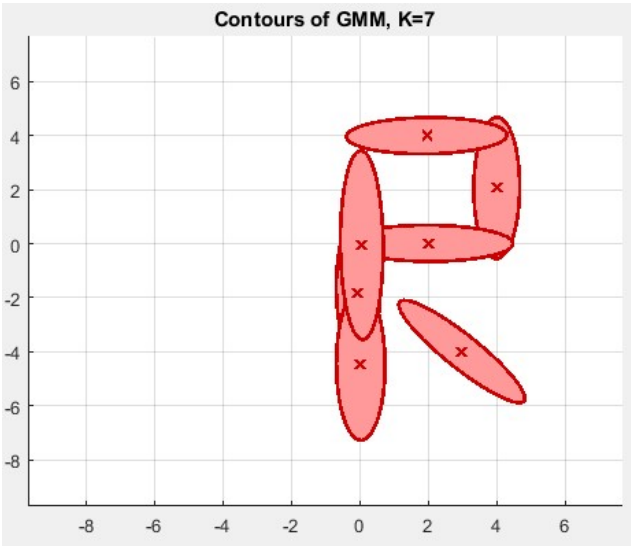
K = 5


Contours of GMM, K=5

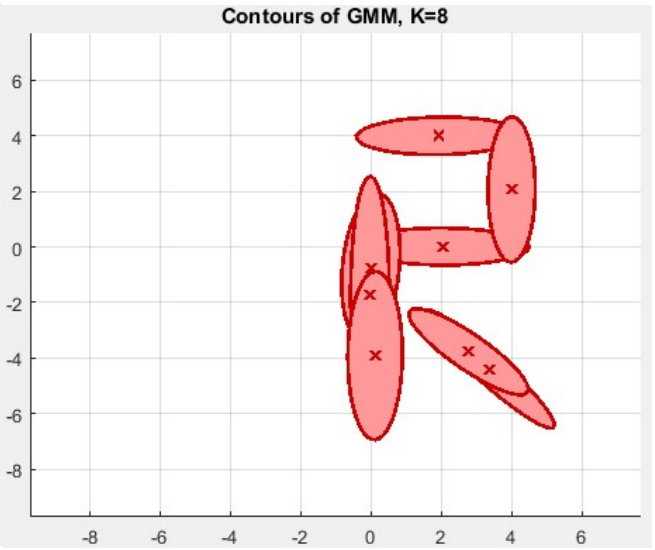
K = 6


Contours of GMM, K=6

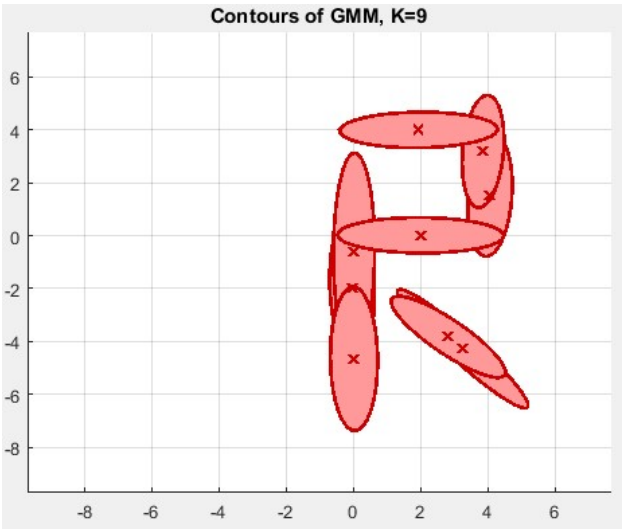
K = 8


Contours of GMM, K=7

K = 9


Contours of GMM, K=8

K = 9

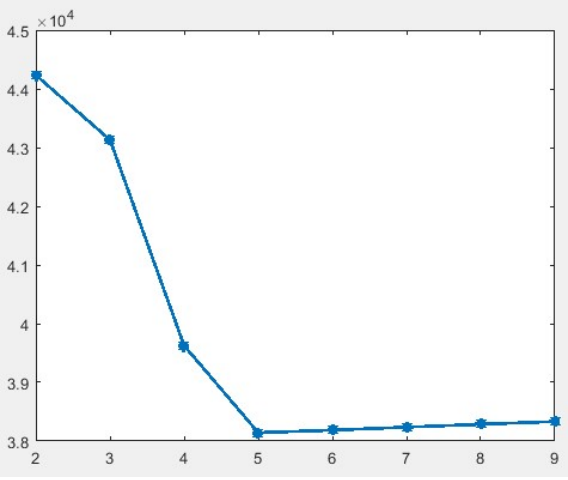

Contours of GMM, K=9

Task 8

Matlab Code:

```
clc; clear all; close all;
load Lab-07
plot(X(:,1),X(:,2),'go','MarkerSize',2);

%% Fit GMM
K = [2,3,4,5,6,7,8,9];
BIC = zeros(length(K),1);
for iter = 1:length(K)
    k = K(iter);
    %% Run k-means
    [Priors, Mu, Sigma] = EM_init_kmeans(X', k);
    Mu = Mu'; % Matlab gmm.fit requires that Mu by KxD

    %% Fit GMM. Use k-mean op for initialization
    s = struct('mu',Mu,'Sigma',Sigma,'PComponents',Priors);
    options = statset('Display','final');
    GMMModel = gmdistribution.fit(X,k,'Options',options,'Start',s);
    BIC(iter) = GMMModel.BIC;

    %% Plot Current GMM
    muModel = GMMModel.mu;
    SigModel = GMMModel.Sigma;
    Wmodel = GMMModel.ComponentProportion;
    figure; hold on; grid on;
    plotGMM(muModel', SigModel, [.8 0 0], 1);
    title(sprintf('Contours of GMM, K=%d',k));
    xlim([min(X(:)) max(X(:))]); % Make axes have the same scale
    ylim([min(X(:)) max(X(:))]);
end

figure;
plot(K,BIC,'-*','LineWidth',2);
```

Output:



Conclusion:

In this assignment, we learnt how to generate and fit the two dimensional Gaussian when from the means and covariance matrices in MATLAB. Secondly, we also learnt, how to fit the Gaussian over the data and finding out how many Gaussians will be fit for the given data in MATLAB.