Name: Jingyi Yuan
UNI: jy2736
Class: EECS E6892

Homework 04

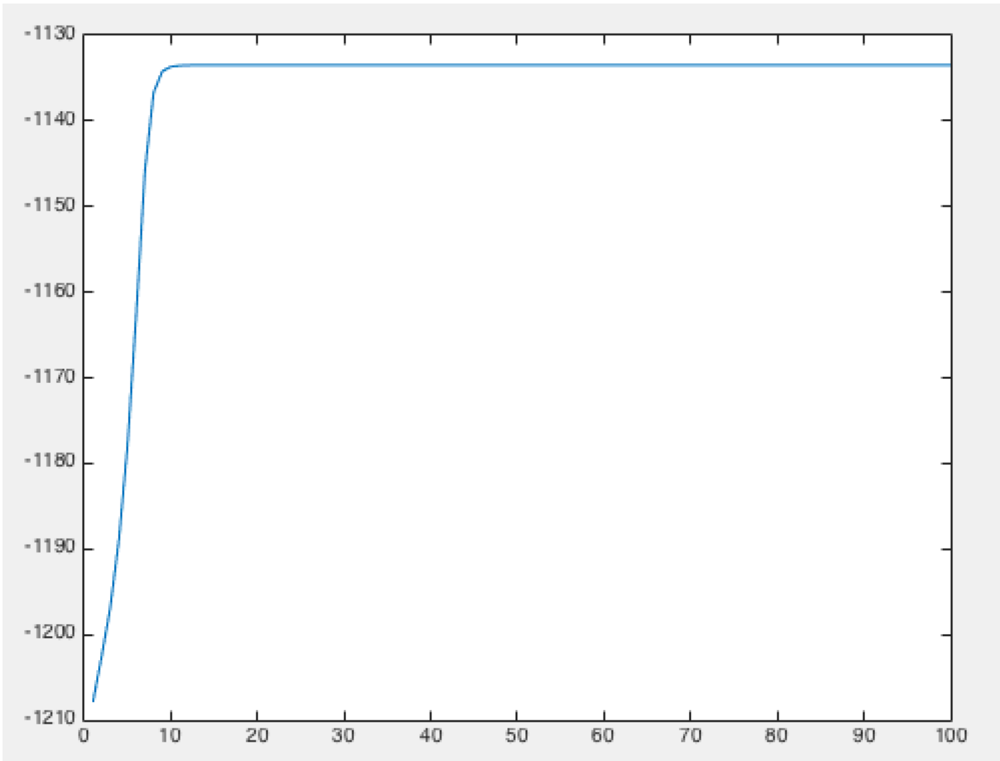
Problem 1

a)

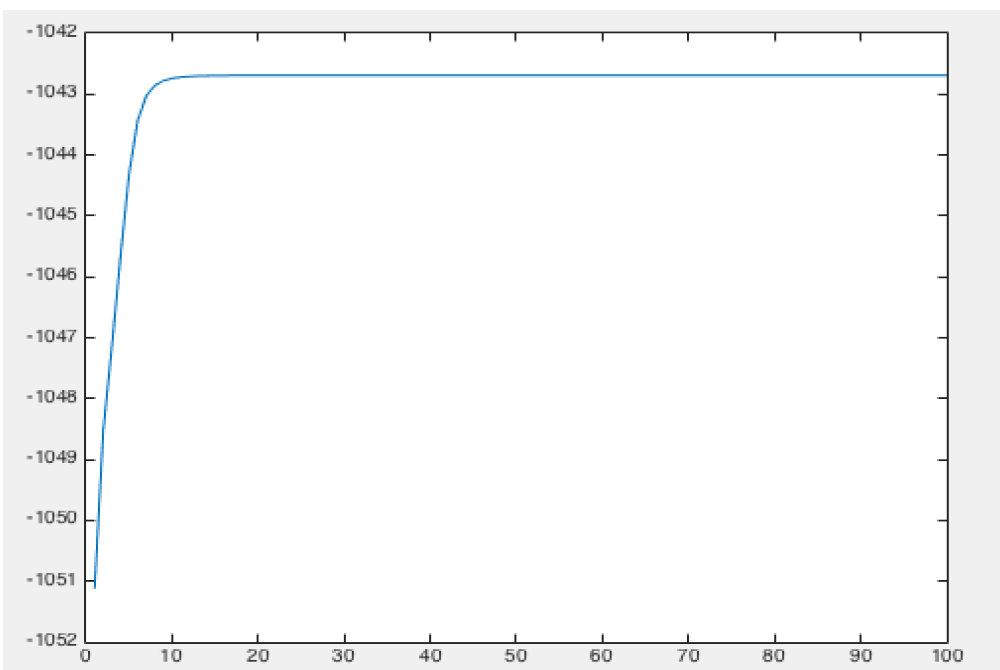
```
clear;
close all

load data.mat
[d, N] = size(X);
T = 100;
K = 2;
fi_mul_x = zeros(d, 1);
fi = zeros(N, K);
nn = zeros(K, 1);
sumfi = zeros(K, 1);
pi_m = zeros(K, 1);
sigma_m = zeros(d, d, K);
ft = zeros(T, 1);

[labels, mu] = kmeans(X', K);% mu:K*d
mu_m = mu';
Y = X';

for j = 1:K
    sigma_m(:,:,j) = cov(Y(labels == j,:));
    pi_m(j) = sum(labels == j)/N;
end

for t = 1:T
    for j = 1:K
        for i = 1:N
            nor = mvnpdf(X(:,i), mu_m(:,j), sigma_m(:,:,j));
            fi(i,j) = pi_m(j) .* nor;%/sum(pi_m .* nor);%num
        end
    end
    ft(t) = sum(log(sum(fi,2)));

    fi = fi./repmat(sum(fi,2),1,K);

    for j = 1:K
        sumfi = sum(fi, 1);
        nn(j) = sumfi(j);
        summ = zeros(d ,1);
        for i = 1:N
            fi_mul_x = fi(i,j).* X(:,i);%d*1
            summ = fi_mul_x + summ;
        end
        mu_m(:,j) = summ./nn(j);%d*1

        sumfixx = zeros(d ,d);
        for i = 1:N
            mul = fi(i,j).* ((X(:,i)-mu_m(:,j))*(X(:,i)-mu_m(:,j))');
            sumfixx = sumfixx + mul;
        end
        sigma_m(:,:,j) = sumfixx./nn(j);%d*d
        pi_m(j) = nn(j)/N;
    end
end
```

b)

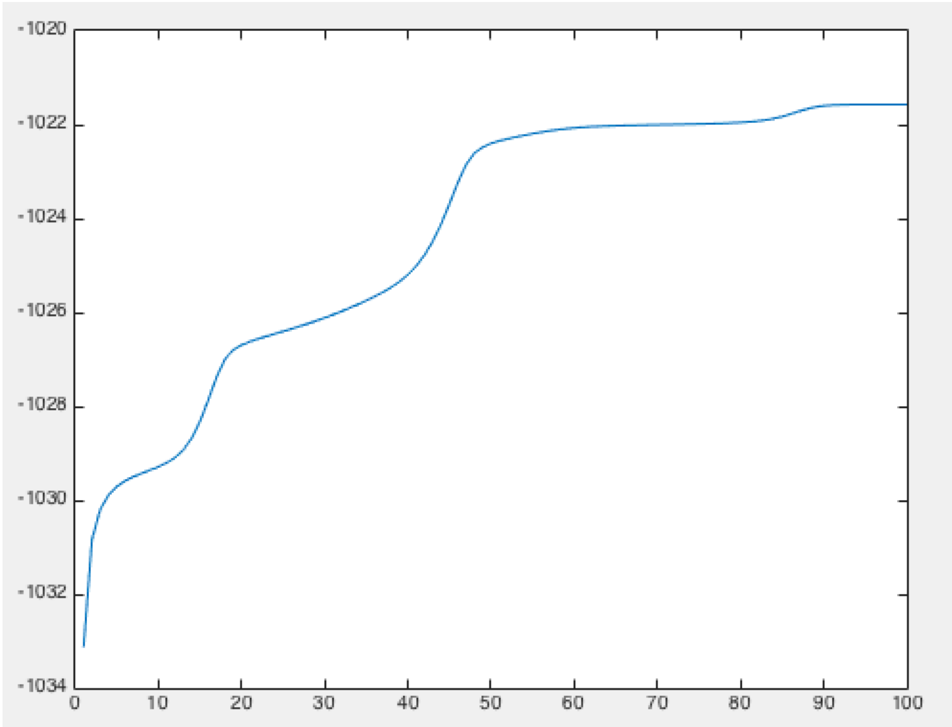
K = 2


-1130
-1140
-1150
-1160
-1170
-1180
-1190
-1200
-1210
0
10
20
30
40
50
60
70
80
90
100

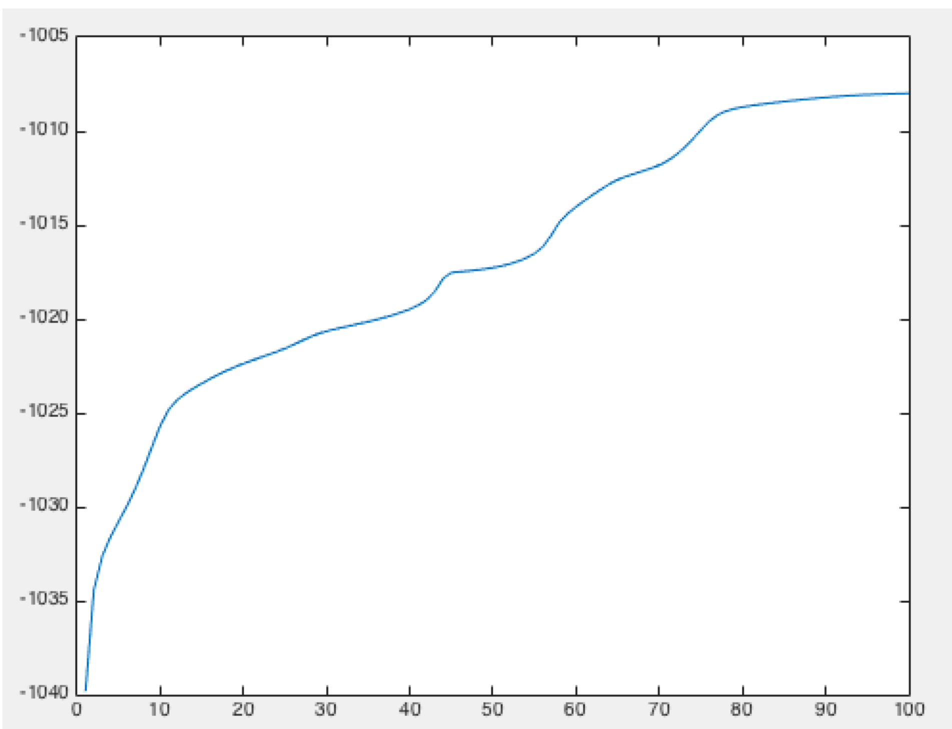
K = 4


-1042
-1043
-1044
-1045
-1046
-1047
-1048
-1049
-1050
-1051
-1052
0
10
20
30
40
50
60
70
80
90
100

K = 8



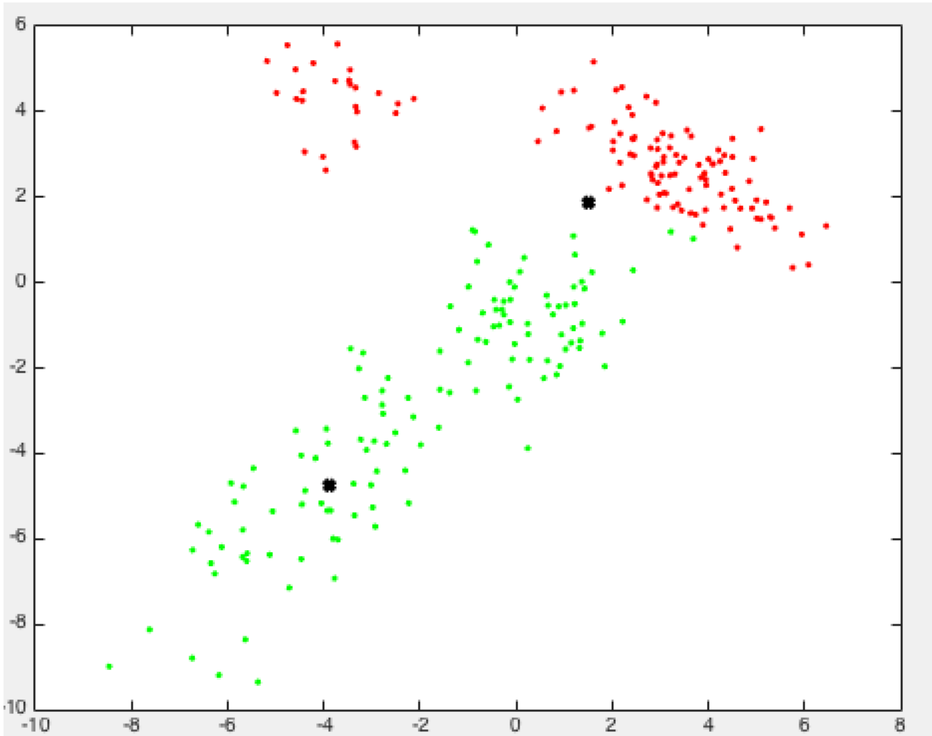
K = 10



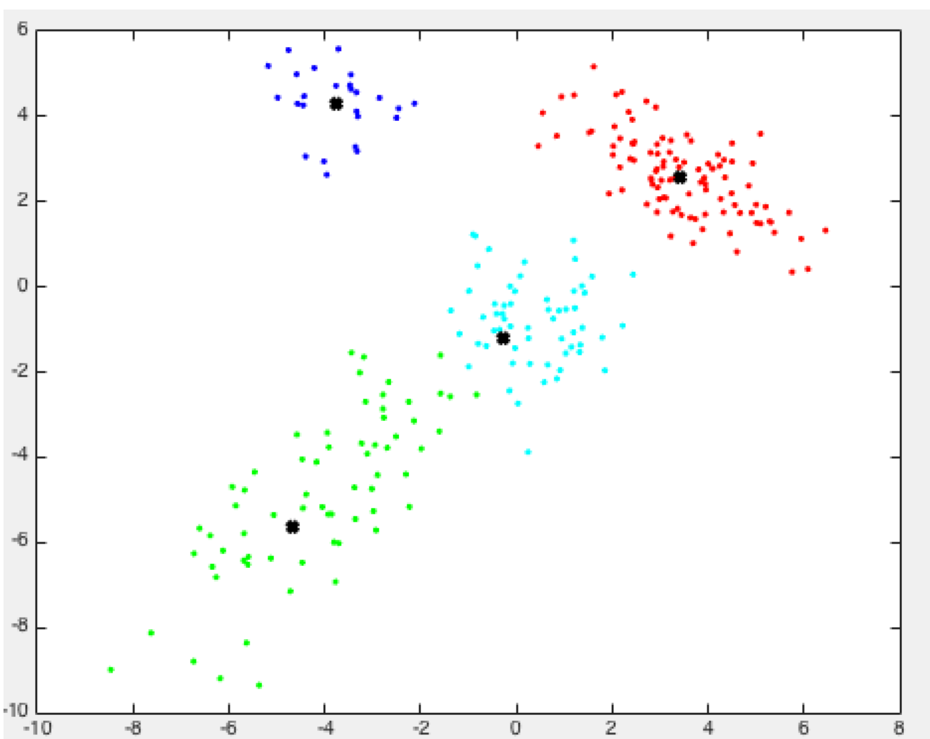
When K becomes larger (as we can see in K = 8 case and K = 10 case), the log likelihood does not converge as well as when K is small. So this might not be the best way to do model selection when K is large.

c)

K = 2

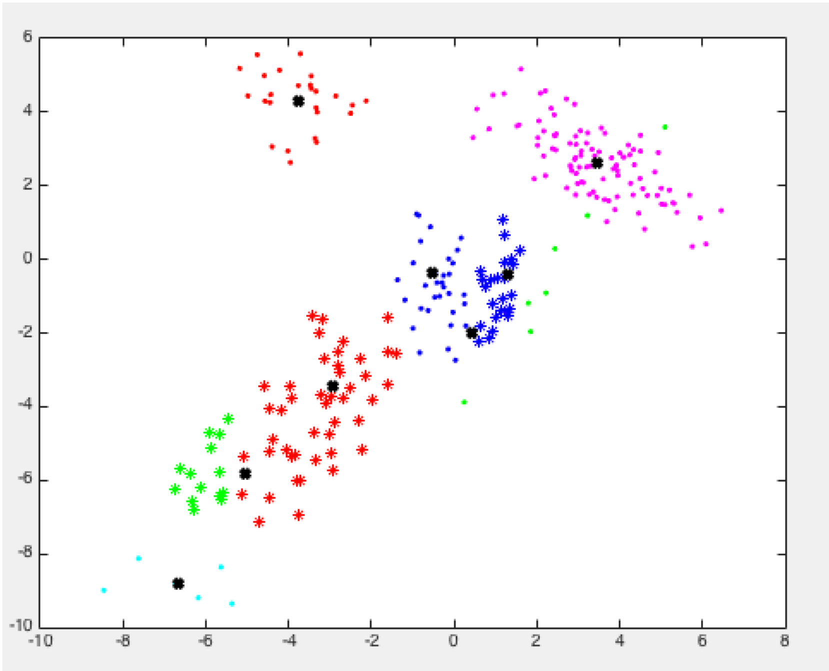


K = 4

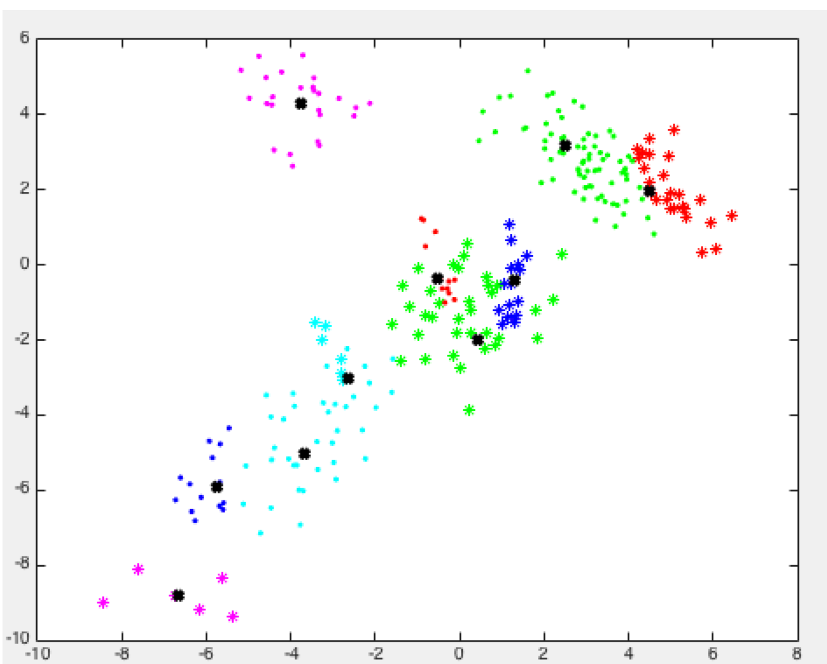


K = 8

("." in red and "* " in red represent different clusters, so do other colors)



K = 10



As K becomes larger, the boundary of each cluster is not as clear as when K is small. For example when K = 8, the green "." cluster is not classified well. This also indicate that the might not be the best way to do classification when K is large.

Problem 2

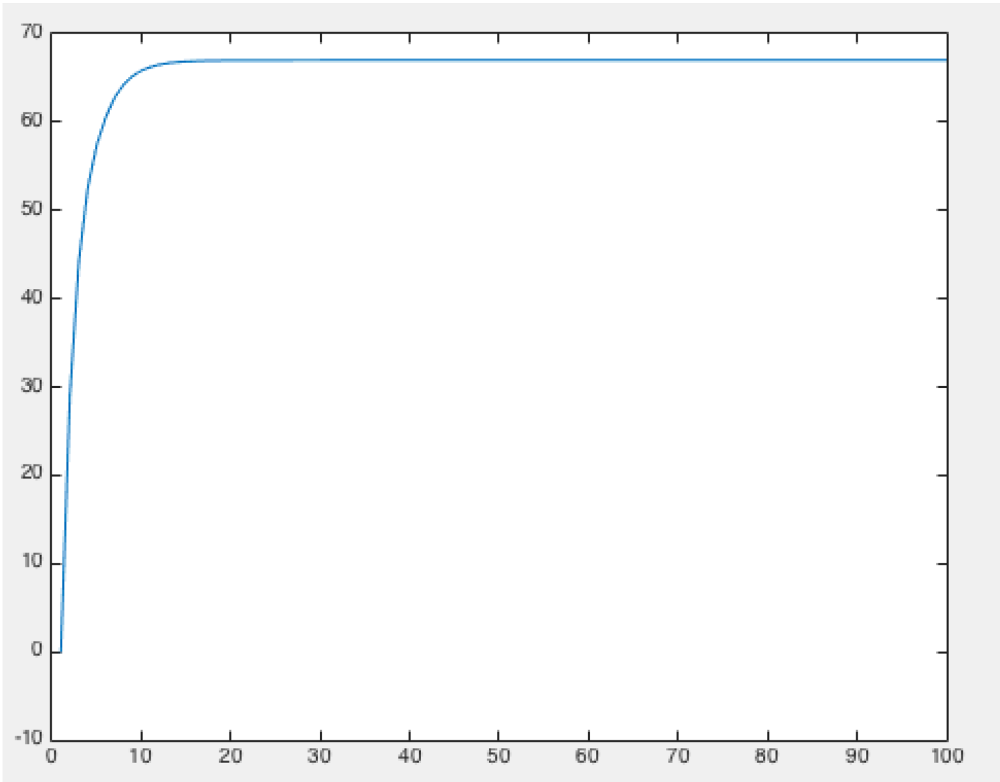
a)

```
clear;
close all

load data.mat
[d, N] = size(X);
K = 10;
T = 100;
t1 = zeros(K,1);
t2 = zeros(K,1);
t3 = zeros(K,1);
t4 = zeros(K,1);
alpha_0 = ones(K,1);
alpha = ones(K,1);
c = 10;
a = d.*ones(K,1);
a_0 = d.*ones(K,1);
A = cov(X(1,:),X(2,:));%d*d
B_0 = d/10.*A;%d*d
B = zeros(d, d, K);%d*d*K
sigma = zeros(d, d, K);
n = zeros(K, 1);
fi = zeros(N, K);
fi_mul_x = zeros(d, 1);

[labels, mu] = kmeans(X', K);% mu:K*d
m = mu';%d*K
%m = zeros(d, K);
Y = X';

for j = 1:K
    B(:,:,j) = B_0;
    sigma(:,:,j) = cov(Y(labels == j,:));
    %sigma(:,:,j) = [0.9, 0.4;0.4, 0.3];
end

L = zeros(T,1);
for t = 1:T
    for j = 1:K
        for i = 1:N
            psia = 0;
            for k = 1:d
                psia = psia + psi((1 - k + a(j))/2);
            end
            t1(j) = psia - log(det(B(:,:,j)));
            t2(j) = (X(:,i) - m(:,j))'*(a(j).*pinv(B(:,:,j)))*(X(:,i) - m(:,j));
            t3(j) = trace(a(j).*pinv(B(:,:,j))*sigma(:,:,j));
            t4(j) = psi(alpha(j)) - psi(sum(alpha));
            fi(i,j) = exp(0.5*t1(j) - 0.5*t2(j) - 0.5*t3(j) + t4(j));
        end
    end
    fi = fi./repmat(sum(fi,2),1,K);
    for j = 1:K
        sumfi = sum(fi, 1);
        n(j) = sumfi(j);
        alpha(j) = alpha_0(j) + n(j);%.*ones(K,1);
        sigma(:,:,j) = pinv(1/c.*eye(d) + n(j)*a(j).*pinv(B(:,:,j)));
        summ = zeros(d ,1);
        for i = 1:N
            fi_mul_x = fi(i,j).* X(:,i);%d*1
```

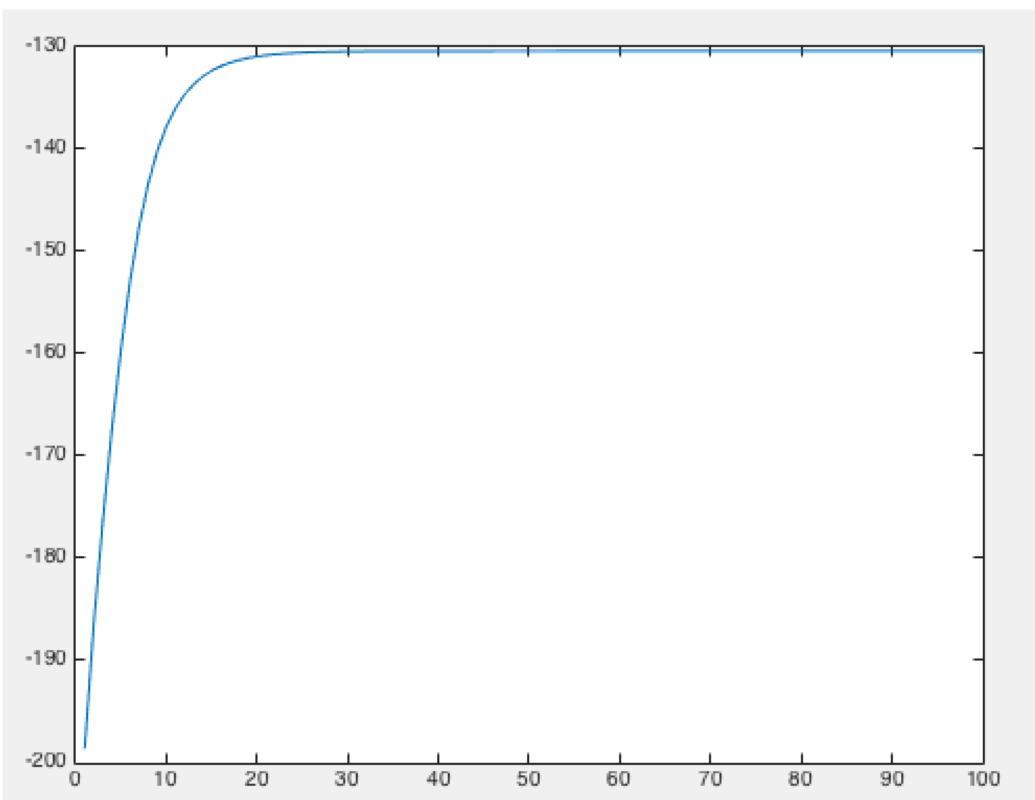
```
            summ = fi_mul_x + summ;
        end
        m(:,j) = sigma(:,:,j) * (a(j)*pinv(B(:,:,j))*summ);
        a(j) = a_0(j) + n(j);

        sumfixx = zeros(d ,d);
        for i = 1:N
            mul = fi(i,j).* ((X(:,i)-m(:,j))*(X(:,i)-m(:,j))'+sigma(:,:,j));
            sumfixx = sumfixx + mul;
        end
        B(:,:,j) = B_0 + sumfixx;
    end

    ga = 1;
    for j = 1:K
        ga = ga + gammaln(alpha(j));
    end
    Elnxcml = 0;
    Elnc = 0;
    Elnpm = 0;
    Elnpt = 0;
    Elnqc = 0;
    Elnqm = 0;
    Elnql = 0;
    Elnqpi = 0;
    for j = 1:K
        for i = 1:N
            Elnxcml = Elnxcml + fi(i,j)*(0.5*t1(j) - 0.5*t2(j) - 0.5*t3(j) + t4(j));
            Elnc = Elnc + fi(i,j)*t4(j);
            Elnqc = Elnqc + fi(i,j)*log(fi(i,j));
        end
        Elnpm = Elnpm - c/2*trace(sigma(:,:,j) + m(:,j)*m(:,j)');
        Elnpt = Elnpt + (a_0(j)-d-1)/2*t1(j) + 0.5*trace(B_0*a(j).*pinv(B(:,:,j)));
        Elnqm = Elnqm + 0.5*log(det(pinv(sigma(:,:,j))));
        Elnql = Elnql - 0.5*log(2)*a(j)*d - log(gamma(0.5*(a(j)+1))) - log(gamma(0.5*a(j))) + 0.5*a(j)*log(det(B(:,:,j))) + 0.5*(a(j)-d+1)*t1(j) - (

        Elnqpi = Elnqpi + (alpha(j) - 1)*t4(j);

    end

    L(t) = Elnxcml + Elnc + Elnpm + Elnpt - Elnqc - Elnqm - Elnql - Elnqpi + log(exp(gammaln(sum(alpha))-ga));
end

plot(L);

fi_t = fi';
[maxnum,I] = max(fi_t);
color = ['r','g','b','c','m','r','g','b','c','m'];
dot = ['.','.','.','.','.','*','*','*','*','*'];

figure;
for i=1:N
    plot(X(1,i),X(2,i),[color(I(i)) dot(I(i))])
    hold on
end
plot(m(1,:),m(2,:),['k','x'],'LineWidth',4)
```

b)

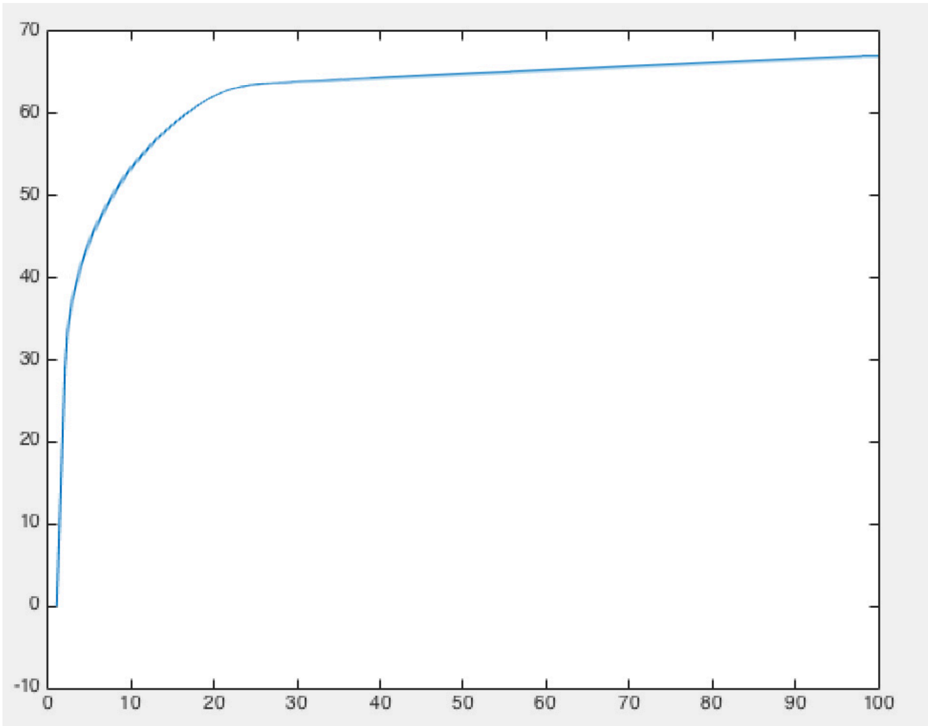
K = 2


70
60
50
40
30
20
10
0
-10
0
10
20
30
40
50
60
70
80
90
100

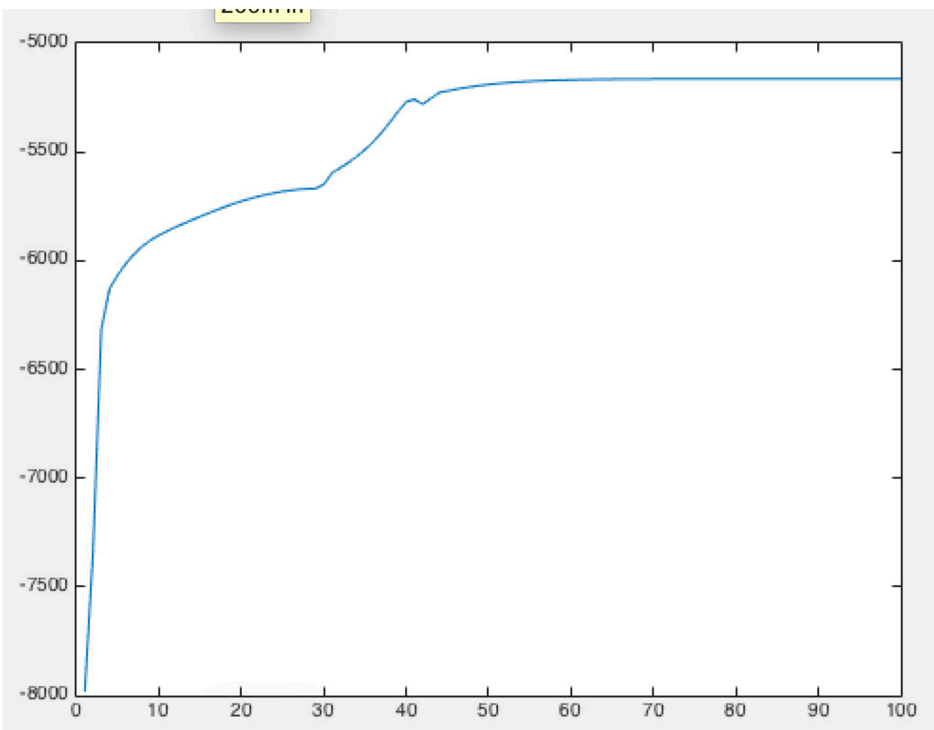
K = 4


-130
-140
-150
-160
-170
-180
-190
-200
0
10
20
30
40
50
60
70
80
90
100

k = 10



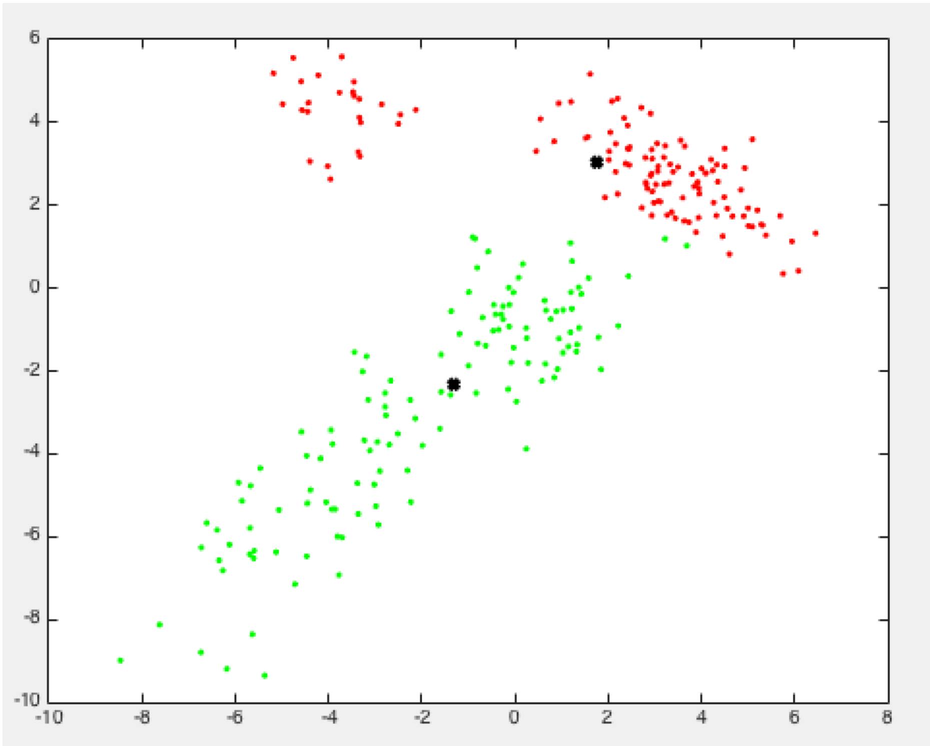
K = 25



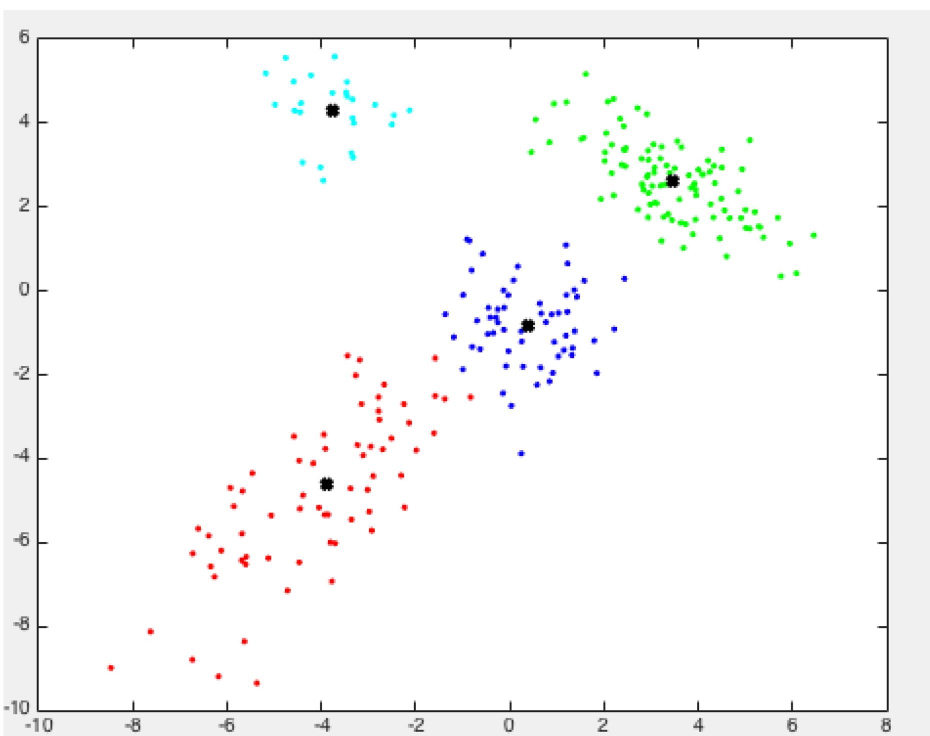
We notice that when K becomes larger, the objective function is not convergent as well as when K is small. When K is 25, the objective function becomes kind of out of shape. But this performs better then EM.

c)

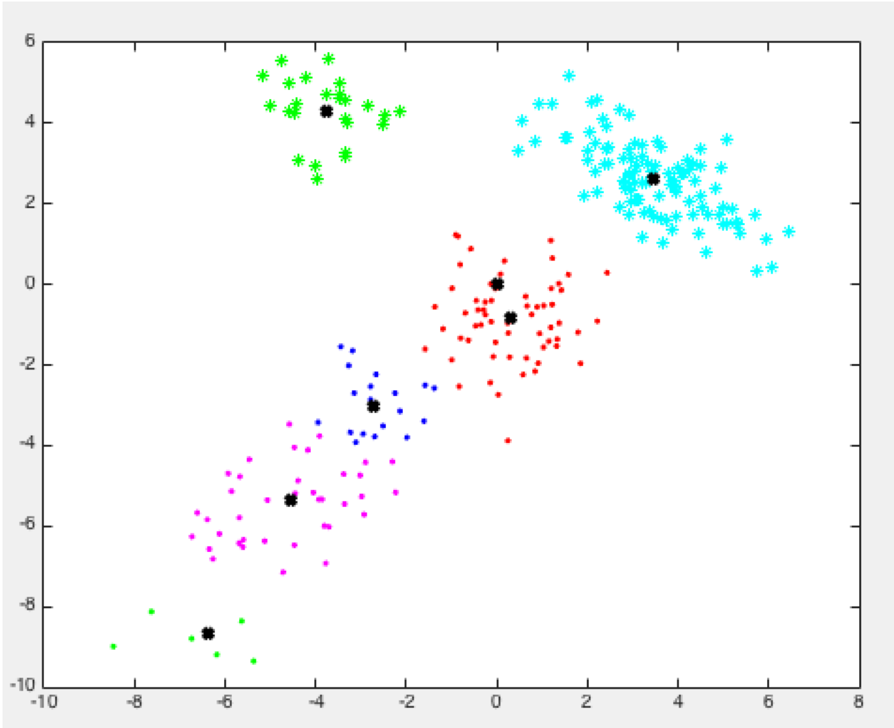
K = 2



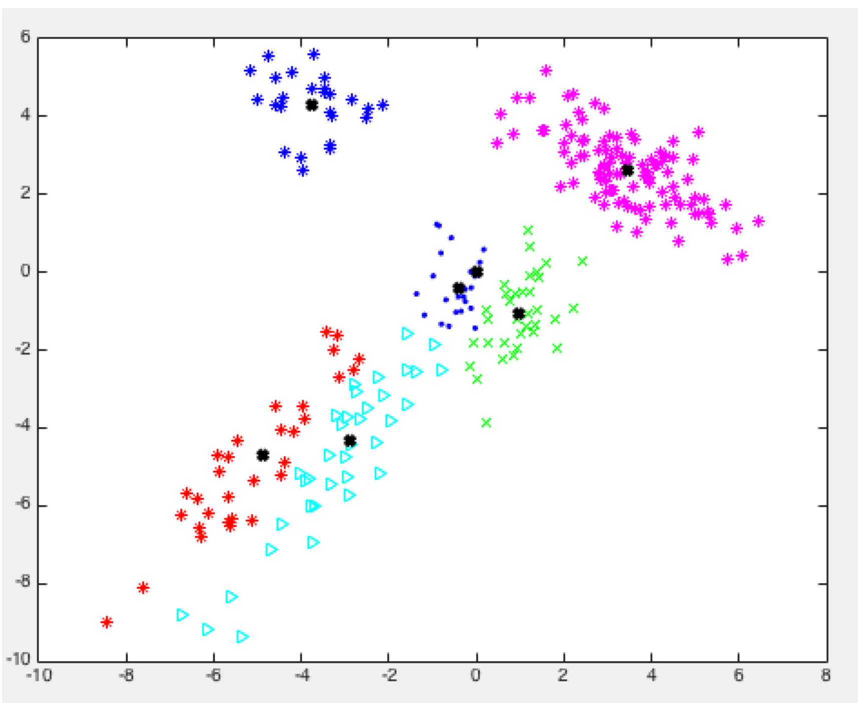
K = 4



K = 10



K = 25



As K becomes larger, VI can only have about 6 clusters. Although the result may be a little different for each cluster, the boundary is pretty clear. VI performs better then EM and the number of the clusters is convergent as K becomes larger.

problem 3

```
clear;
close all

load data.mat
[d, N] = size(X);
Y = X';
labels = ones(N,1);

m_0 = mean(Y)';%2*1
c_0 = 0.1;
a_0 = d;
A_0 = cov(X(1,:),X(2,:));
B_0 = c_0*d.*A_0;
alpha = 1;

K = 1;

T = 1;
for t = 1:T
s = zeros(N,1);%size of cluster j
sumx = zeros(d,1);
m = zeros(d,1);
c = zeros(N,1);
a = zeros(N,1);
B = zeros(d, d, N);
sigma = zeros(d, d, N);
fi = zeros(N,1);
fi_new = zeros(N,1);
mu = zeros(d,1);
for j = 1:K
    n = zeros(N,1);
    for ii = 1:N%calculate n(i) & s(j)
```

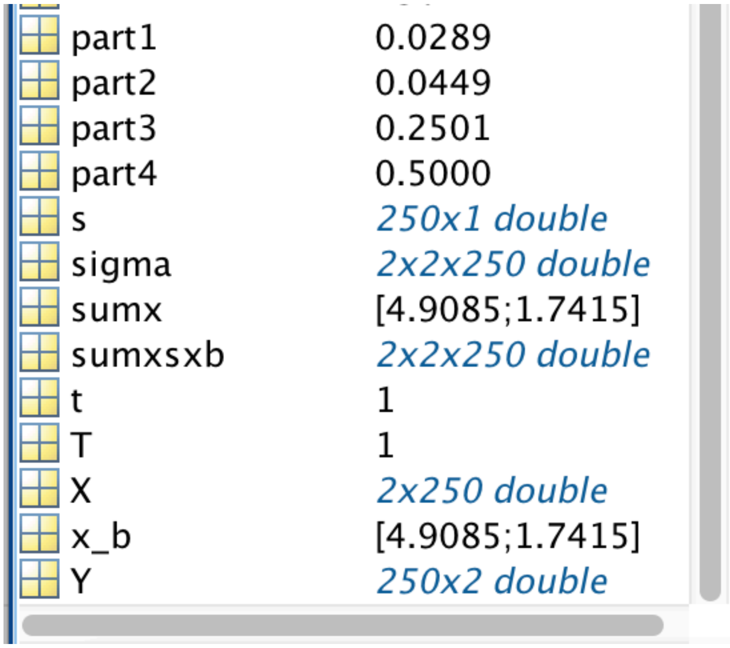
```
            if(labels(ii) == j)
                s(j) = s(j) + 1;%how many numbers in cluster j
            end
            n(j) = s(j) - 1;
            if (labels(ii) == j)
                sumx = sumx + X(:,ii);
            end
        end%calculate n(i)

        x_b = sumx./s(j);%d*1
        m(:,j) = c_0/(c_0 + s(j)).*m_0 + 1/(c_0 + s(j)).*sumx;
        c(j) = s(j) + c_0;
        a(j) = s(j) + a_0;
        sumxsxb = zeros(d, d, N);

        for ii = 1:N
            if (labels(ii) == j)
                sumxsxb(:,:,j) = sumxsxb(:,:,j)  + (X(:,ii)-x_b)*(X(:,ii)-x_b)';
            end
        end
        B(:,:,j) = B_0 + sumxsxb(:,:,j)  + s(j)/(a(j)*s(j)+1).*(x_b - m(:,j))*(x_b - m(:,j))';

        sigma(:,:,j) = wishrnd(B(:,:,j),a(j));
        mu(:,j) = mvnrnd(m(:,j), pinv(c(j).*sigma(:,:,j)));
    end


    for i = 1:3
    for j = 1:K
        %calculate phi for current cluster
        fi(j) = mvnpdf(X(:,i), mu(:,j), pinv(sigma(:,:,j))) .* (n(j)/(alpha + N -1));
```

```
        %calculate phi for a new cluster
        part1 = c_0/(pi*(1+c_0))^(d/2);
        part2 = det(B_0 + (c_0/(1+c_0).*(X(:,i)-m_0)*(X(:,i)-m_0)'))^(-0.5*(a_0+1));
        part3 = det(B_0)^(-0.5*a_0);
        part4 = exp(gammaln((a_0+1)/2) + gammaln(a_0/2) - gammaln(a_0/2) - gammaln((a_0-1)/2));
        fi_new(j) = alpha/(alpha + N - 1)*(part1 * part2/part3 * part4);

        fi_oldd = fi(j)/(fi(j) + fi_new(j));
        fi_neww = fi_new(j)/(fi(j) + fi_new(j));
    end
        class = discretesample([fi_oldd, fi_neww], 1);
        if (class == 2)
            K = K + 1;
            %[labels, mu] = kmeans(X', K);%random sample
            labels(i) = K;%put i in the new cluster

            x_b = X(:,i);%d*1
            sumx = X(:,i);%d*1
            m(:,K) = c_0/(c_0 + 1).*m_0 + 1/(c_0 + 1).*sumx;
            c(K) = 1 + c_0;
            a(K) = 1 + a_0;
            sumxsxb(:,:,K) = (X(:,i)-x_b)*(X(:,i)-x_b)';

            B(:,:,K) = B_0 + sumxsxb(:,:,K)  + 1/(a(K)+1).*(x_b - m(:,K))*(x_b - m(:,K))';

            sigma(:,:,K) = wishrnd(B(:,:,K),a(K));
            mu(:,K) = mvnrnd(m(:,K), pinv(c(K).*sigma(:,:,K)));
        end
        %end
    end
    end
```

The workspace is like follows:

Workspace

Name	Value
a	250x1 double
a_0	2
A_0	[12.3038,7.5328;...
alpha	1
B	2x2x250 double
B_0	[2.4608,1.5066;1...
c	250x1 double
c_0	0.1000
class	2
d	2
fi	250x1 double
fi_new	250x1 double
fi_neww	1
fi_oldd	0
i	3
ii	250
j	3
K	4
labels	250x1 double
m	[1.1946e-16,3.7...
m_0	[1.1946e-16;-3....
mu	[-6.9640e-05,5....
n	250x1 double
N	250



We can guess that K is convergent with iterations.