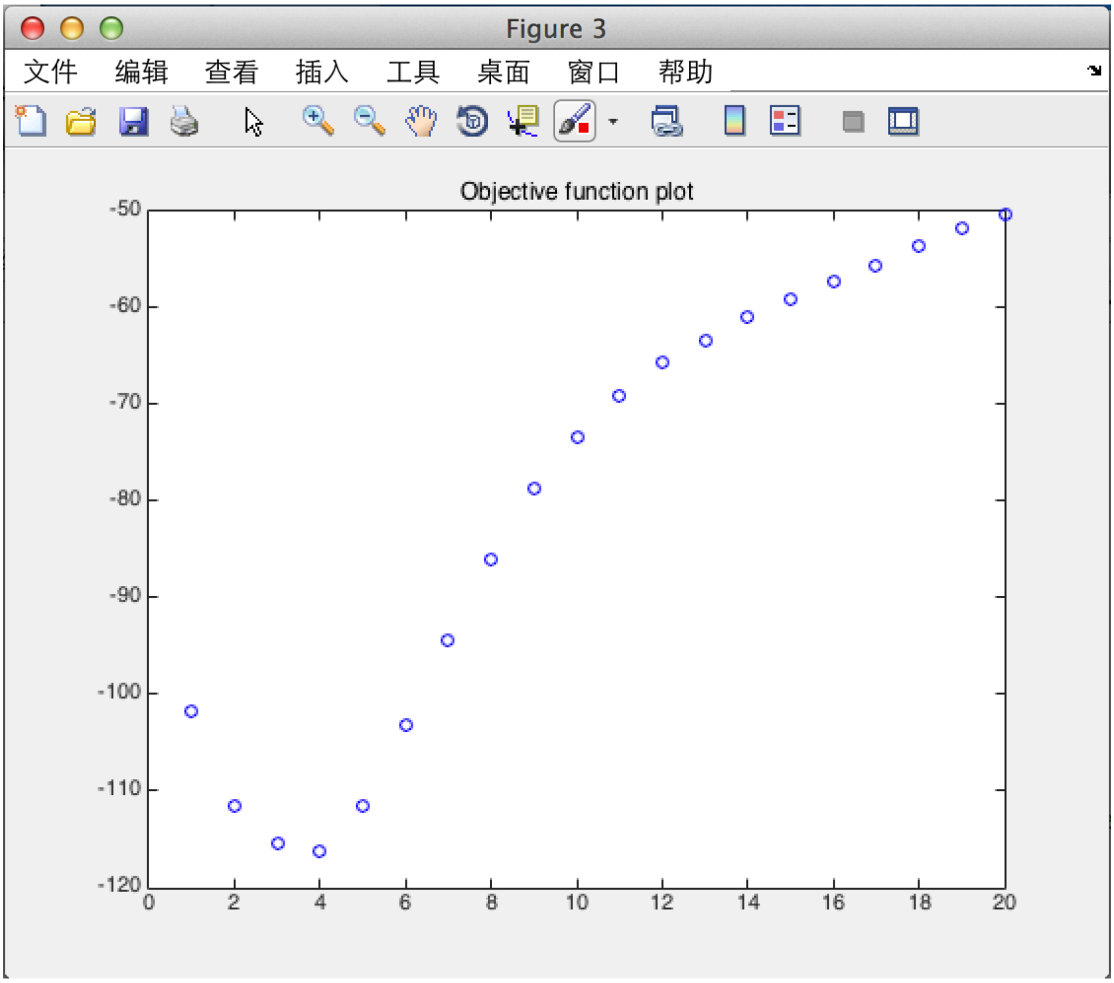
Problem 1

Part 1:

With the
The objective function value with 20 iterations is showed below:
The X-axis is the iteration times, and the Y-axis is the objective function value.

(1) K=2



(2) K=3

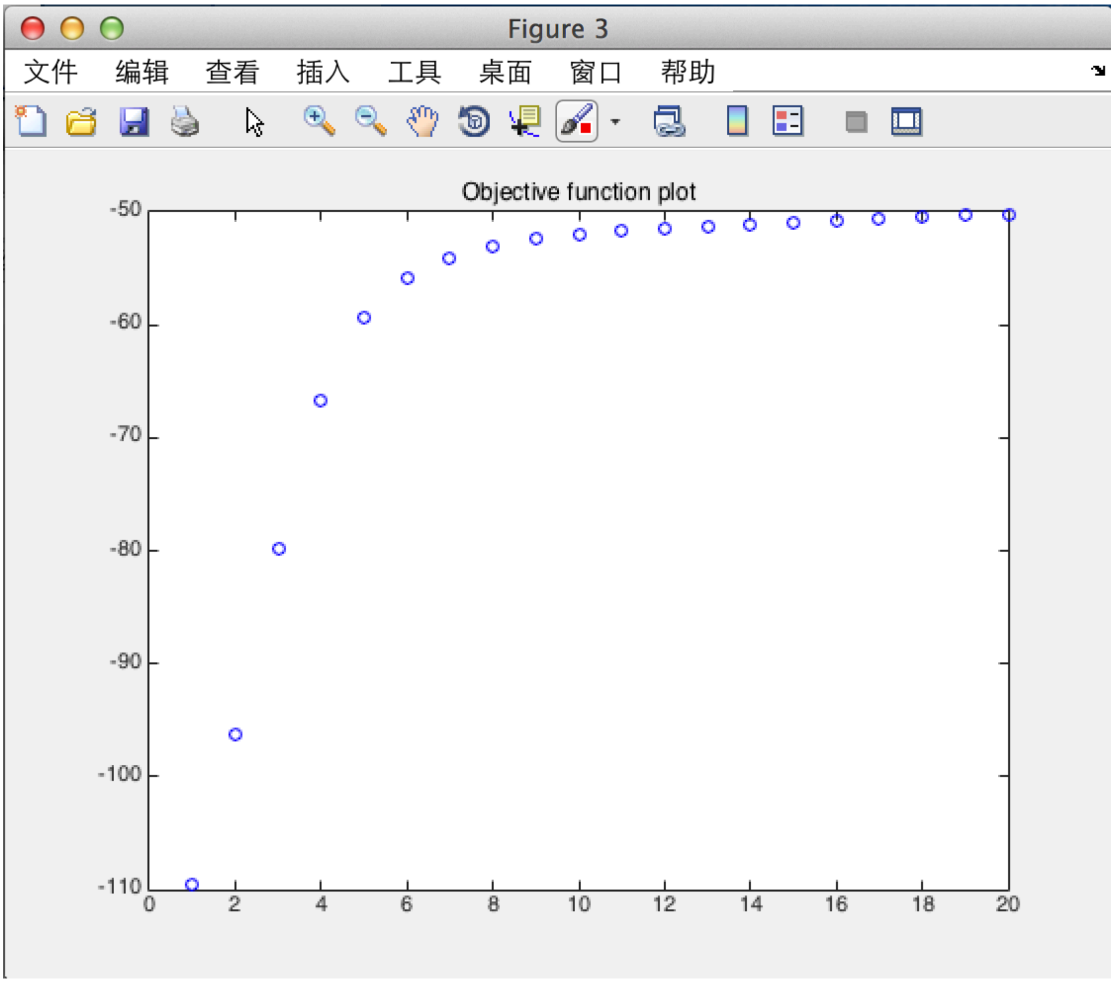


Figure 3
文件
编辑
查看
插入
工具
桌面
窗口
帮助
Objective function plot
-50
-60
-70
-80
-90
-100
-110
0
2
4
6
8
10
12
14
16
18
20

(3) K=4

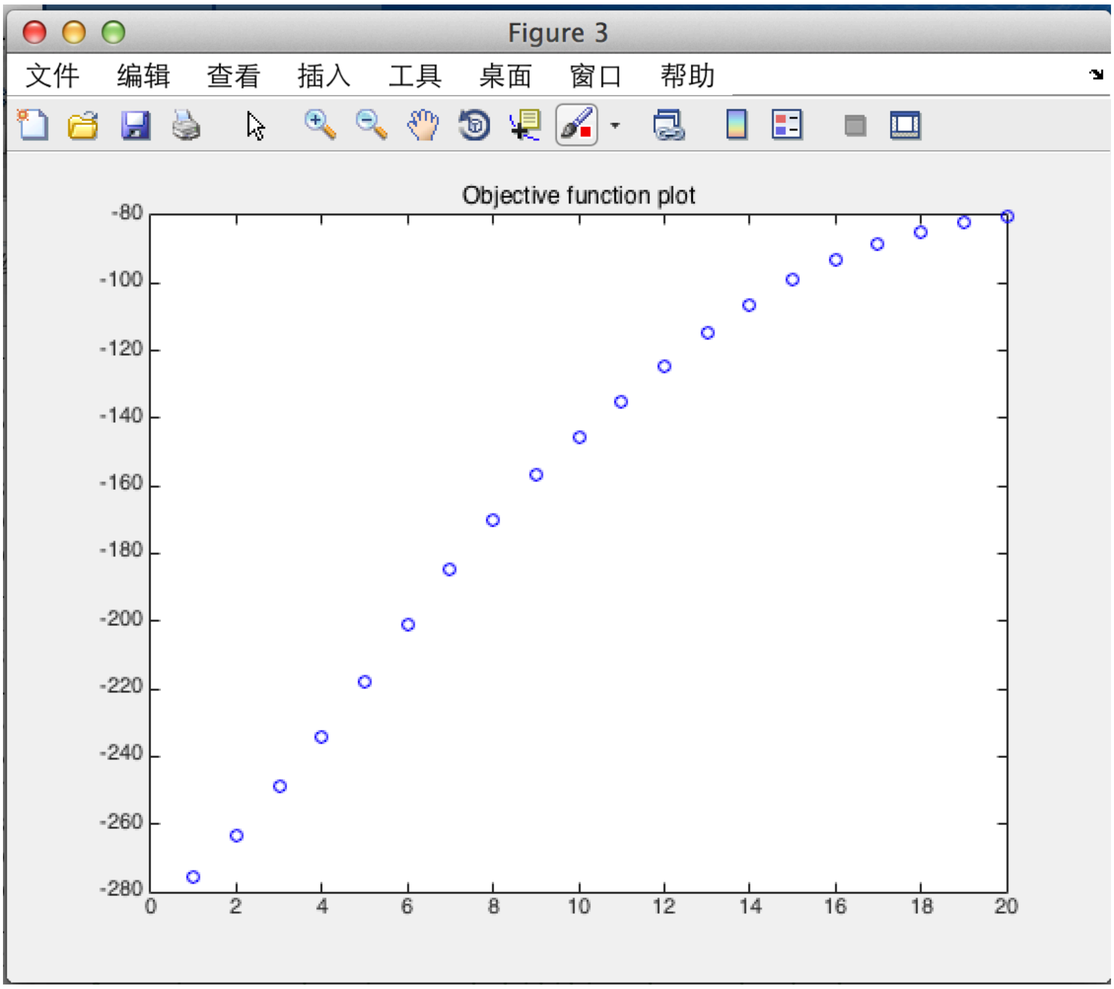
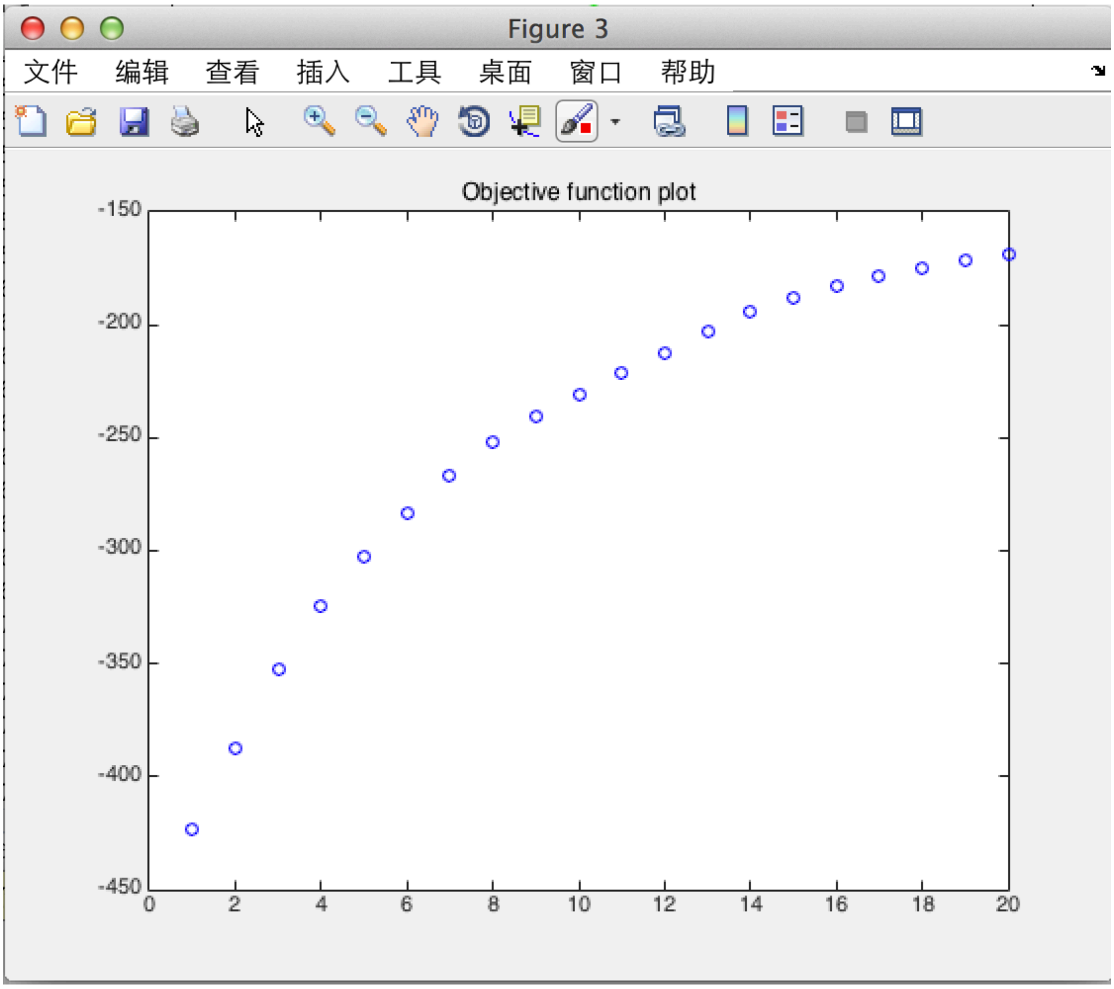


Figure 3
文件
编辑
查看
插入
工具
桌面
窗口
帮助
Objective function plot
-80
-100
-120
-140
-160
-180
-200
-220
-240
-260
-280
0
2
4
6
8
10
12
14
16
18
20

(4) K=5



With some more experiment, we can see that the graph is not necessarily the same, since the random choice of initial condition is different, and will influence how our algorithm will work.

Part 2:

I plot the original data with the pre-determined pdf, and the original data with predict cluster. And make comparison with them, the graph is showed below:
In each graph, the left one is the original cluster with pre-determined pdf, and the right one is the predict result.

(1) K=3

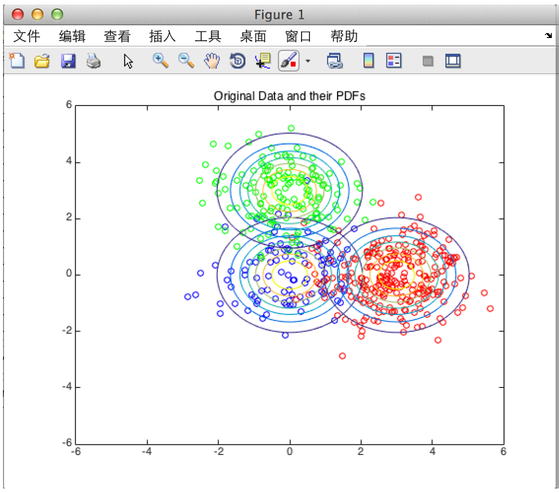


Figure 1
文件 编辑 查看 插入 工具 桌面 窗口 帮助
Original Data and their PDFs
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

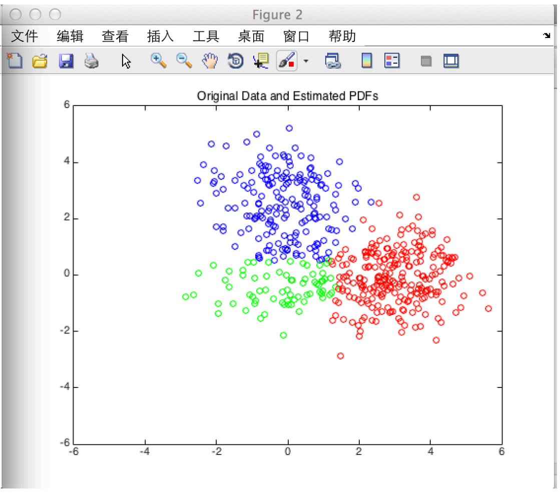


Figure 2
文件 编辑 查看 插入 工具 桌面 窗口 帮助
Original Data and Estimated PDFs
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

(2) K＝5

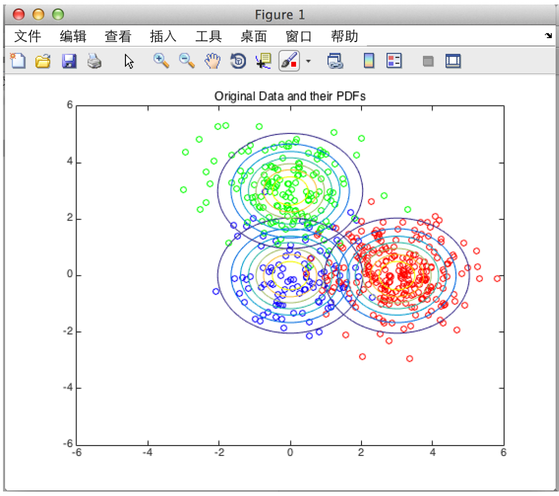


Figure 1
文件 编辑 查看 插入 工具 桌面 窗口 帮助
Original Data and their PDFs
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

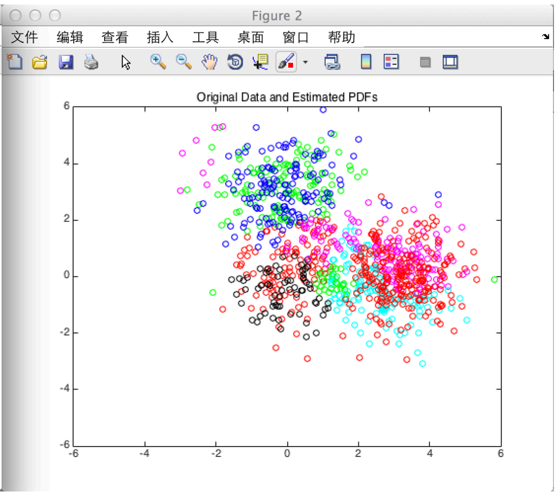


Figure 2
文件 编辑 查看 插入 工具 桌面 窗口 帮助
Original Data and Estimated PDFs
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

And with another try, I got some different result.

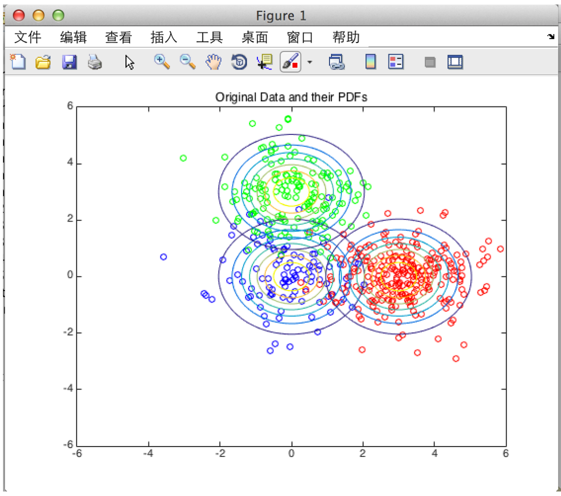


Figure 1
文件 编辑 查看 插入 工具 桌面 窗口 帮助
Original Data and their PDFs
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

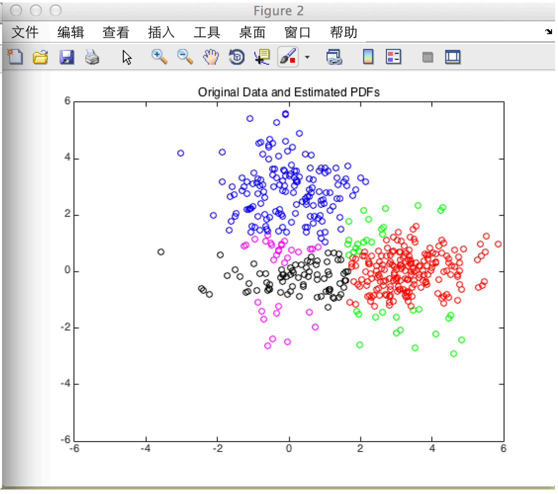


Figure 2
文件 编辑 查看 插入 工具 桌面 窗口 帮助
Original Data and Estimated PDFs
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6

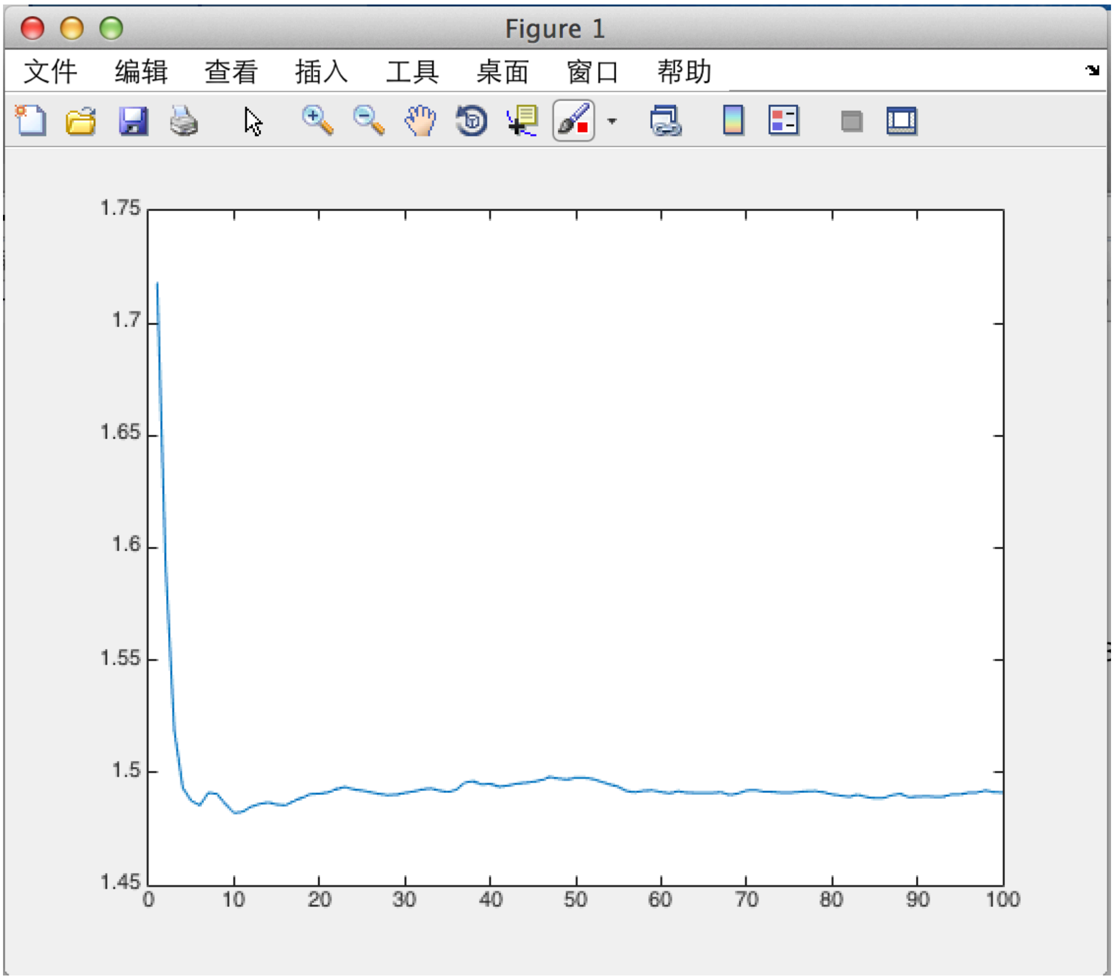
With the result above, we can see that K=3 can easily cluster the data clearly, yet K=5 will be a lot cover each other. Also, the result is not stable, there may be several kinds of data can be seen.

Problem 2

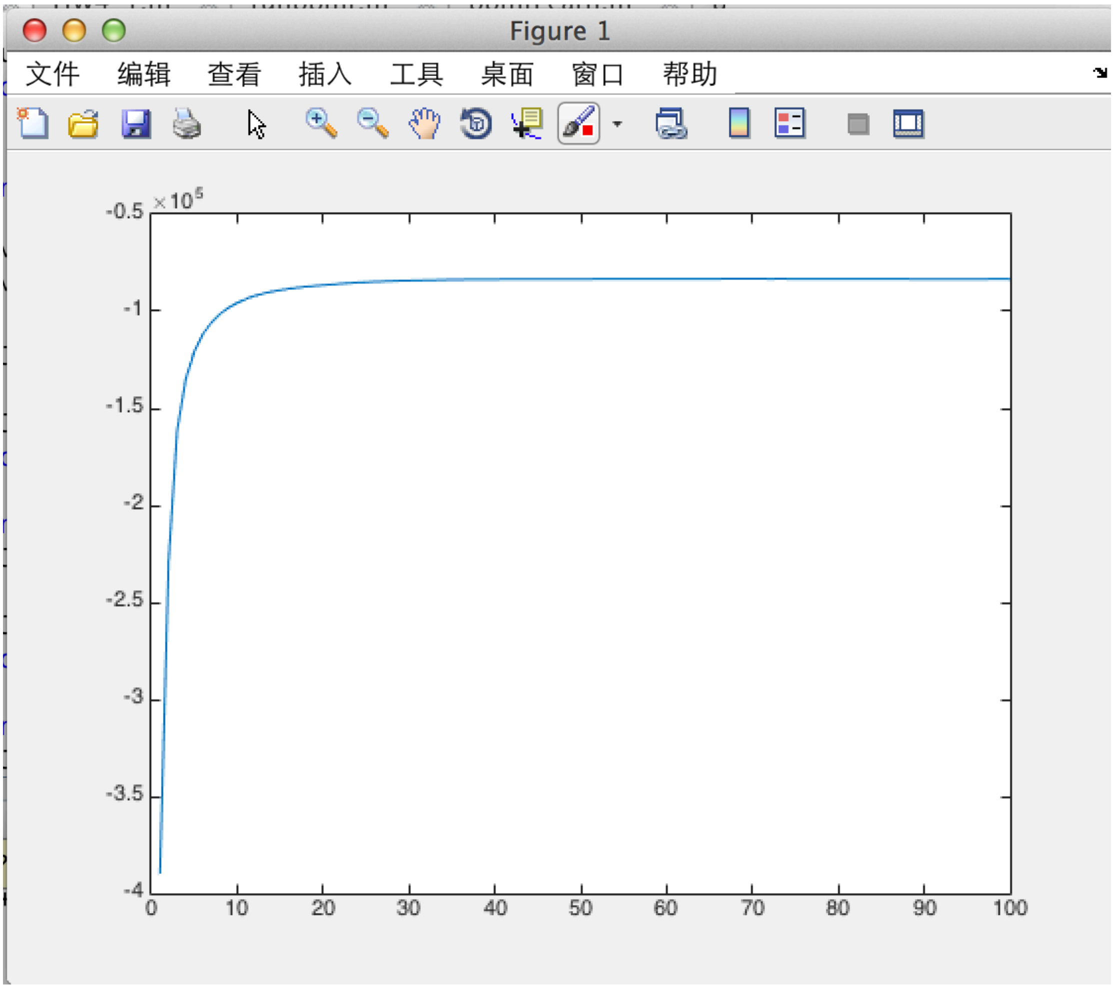
Part 1:

Since wei got 100 iterations, we set iteration times as X-axis value, Y-axis value is according to each problem.

(1)The RMSE graph is listed below:



(2) The objective function plot is listed below:



Part 2:

As the requirement said, I set up a random function, to find 3 certain movies, then for each one of them, find the 5 nearest movies.

One sample result is showed below:

3 movies randomly chosen and their nearest 5 neighbors are listed below :

```
      1

     'Cool Hand Luke (1967)'

 :
     'Substance of Fire, The (1996)'

     'Nina Takes a Lover (1994)'

     'Blink (1994)'
```

```
      'Across the Sea of Time (1995)'

      'Pagemaster, The (1994)'

        2

      'Gandhi (1982)'

  :
      'Substance of Fire, The (1996)'

      'Nina Takes a Lover (1994)'

      'Blink (1994)'

      'Across the Sea of Time (1995)'

      'Pagemaster, The (1994)'

        3

      'World of Apu, The (Apur Sansar) (1959)'

  :
      'Substance of Fire, The (1996)'

      'Nina Takes a Lover (1994)'

      'Blink (1994)'

      'Across the Sea of Time (1995)'

      'Pagemaster, The (1994)'
```

Part 3:

I apply K-means algorithm for Ui, and K=30. Vector length is 20.
After the algorithm is done, randomly select 5 centroids from 30, then calculate the dot product of all movies according to certain centroid. Then pick 10 largest ones.

The result of program is listed below:

```
5 centroid randomly chosen  ，  their indexes are: (among 30 cluster)
     5
    12
     6
    22
    21

Their closest 10 movies are listed below :
Centroid
     1

  10 movies are :
    'Leaving Las Vegas (1995)'

    'L.A. Confidential (1997)'

    'Star Wars (1977)'

    'Close Shave, A (1995)'

    'Swingers (1996)'

    'Basquiat (1996)'

    'Wrong Trousers, The (1993)'

    'Shallow Grave (1994)'

    'Full Monty, The (1997)'

    'Empire Strikes Back, The (1980)'

Centroid
     2

  10 movies are :
    'Spitfire Grill, The (1996)'

    'Amistad (1997)'

    'Mr. Smith Goes to Washington (1939)'

    'Bridge on the River Kwai, The (1957)'
```

```
        'Schindler's List (1993)'

        'Patton (1970)'

        'Magnificent Seven, The (1954)'

        'Searching for Bobby Fischer (1993)'

        'Singin' in the Rain (1952)'

        'Titanic (1997)'

Centroid
     3

  10 movies are :
        'Clockwork Orange, A (1971)'

        'Boogie Nights (1997)'

        'Wag the Dog (1997)'

        'Godfather, The (1972)'

        'Big Lebowski, The (1998)'

        'Bonnie and Clyde (1967)'

        'Forrest Gump (1994)'

        'Bridge on the River Kwai, The (1957)'

        'Apocalypse Now (1979)'

        'As Good As It Gets (1997)'

Centroid
     4

  10 movies are :
        'Titanic (1997)'

        'Secrets & Lies (1996)'
        'Fargo (1996)'

        'Shall We Dance? (1996)'

        'Chasing Amy (1997)'

        'Face/Off (1997)'

        'Strictly Ballroom (1992)'

        'Usual Suspects, The (1995)'

        'Wag the Dog (1997)'

        'Full Monty, The (1997)'

Centroid
    5

  10 movies are :
        'Tomorrow Never Dies (1997)'

        'Citizen Kane (1941)'

        'L.A. Confidential (1997)'

        'Leaving Las Vegas (1995)'

        'Contact (1997)'

        'Brazil (1985)'

        'Deer Hunter, The (1978)'

        'Ulee's Gold (1997)'

        'Mother (1996)'

        'Sling Blade (1996)'
```

Source Code Attachment

For problem 2, the part 2 and part 3 code must have the result in part 1 in working region. Or we can just put code 1,2,3 one by one and we can get all results.

Problem 1,

```
clear;
%1 Generate 500 points

mu1 = [0 0];      % Mean
sigma1 = [ 1 .0;   % Covariance matrix
        .0  1];
m1 = 100;         % Number of data points

mu2 = [3 0];
sigma2 = [1 0;
        0 1];
m2 = 250;

mu3 = [0 3];
sigma3 = [1 0;
        0 1];
m3 = 150;

R1 = chol(sigma1);
X1 = randn(m1, 2) * R1;
X1 = X1 + repmat(mu1, size(X1, 1), 1);

R2 = chol(sigma2);
X2 = randn(m2, 2) * R2;
X2 = X2 + repmat(mu2, size(X2, 1), 1);

R3 = chol(sigma3);
X3 = randn(m3, 2) * R3;
X3 = X3 + repmat(mu3, size(X3, 1), 1);

X = [X1; X2; X3];
```

```matlab
%2 Plot original data

figure(1);

hold off;
plot(X1(:, 1), X1(:, 2), 'bo');
hold on;
plot(X2(:, 1), X2(:, 2), 'ro');
hold on;
plot(X3(:, 1), X3(:, 2), 'go');

set(gcf,'color','white')

gridSize = 100;
u = linspace(-6, 6, gridSize);
[A B] = meshgrid(u, u);
gridX = [A(:), B(:)];

z1 = gaussianND(gridX, mu1, sigma1);
z2 = gaussianND(gridX, mu2, sigma2);
z3 = gaussianND(gridX, mu3, sigma3);

Z1 = reshape(z1, gridSize, gridSize);
Z2 = reshape(z2, gridSize, gridSize);
Z3 = reshape(z3, gridSize, gridSize);

[C, h] = contour(u, u, Z1);
[C, h] = contour(u, u, Z2);
[C, h] = contour(u, u, Z3);

axis([-6 6 -6 6])
title('Original Data and their PDFs');

%3 Initiation of the dataset

m = size(X, 1);

k = 2;
n = 2;  % vector lengths.

% k random initial means
indeces = randperm(m);
mu = X(indeces(1:k), :);
```

```
sigma = [];

% take overal covariance of the dataset as initial variance for each cluster.
for (j = 1 : k)
    sigma{j} = cov(X);
end

% assign equal prior probabilities to each cluster.
phi = ones(1, k) * (1 / k);

% 4: Expectation Maximization

W = zeros(m, k);

% iteration=20
objectPlot= zeros(20,2);

for iter = 1:20

    fprintf('Iteration %d\n ', iter);

    pdf = zeros(m, k);

    % For each cluster...
    for j = 1 : k

        % Evaluate the Gaussian for all data points for cluster 'j'.
        pdf(:, j) = gaussianND(X, mu(j, :), sigma{j});
    end

    % multiply each pdf value by the prior probability for cluster.
    pdf_w = bsxfun(@times, pdf, phi);

    W = bsxfun(@rdivide, pdf_w, sum(pdf_w, 2));

    % do maximization

    % For each of the clusters...
    for j = 1 : k

        % Calculate the prior probability for cluster 'j'.
        phi(j) = mean(W(:, j), 1);
        mu(j, :) = weightedAverage(W(:, j), X);
        sigma_k = zeros(n, n);
        Xm = bsxfun(@minus, X, mu(j, :));

        % Calculate the contribution of each training example to the
covariance matrix.
        for i = 1 : m
            sigma_k = sigma_k + (W(i, j) .* (Xm(i, :)' * Xm(i, :)));
        end

        % Divide by the sum of weights.
        sigma{j} = sigma_k ./ sum(W(:, j));
    end

%    objectValue=0;
%    for i=1:500
%        tmpValue=0;
%        for j=1:k
%            tmpValue=tmpValue+W(i,j)*phi(j);
%        end
%        log(tmpValue)
%        objectValue=objectValue+log(tmpValue);
%    end

    objectValue=0;
    for i=1:500
        objectValue=objectValue+log(max(max(W(i,:))));
    end

    objectPlot(iter,1)= objectValue;
    objectPlot(iter,2)= iter;
%    figure(iter+2);
%    plot(objectPlot(:,2), objectPlot(:,1), 'bo');

    %fprintf(sigma{j});

end

figure(2);
color=['bo','ro','go','yo','po',];
for i=1:500
    [x y]=find(W(i,:)==max(max(W(i,:))));
    cluster=y;
    if (cluster==1)
```

```
        plot(X(i, 1), X(i, 2), 'bo');
    end
    if (cluster==2)
        plot(X(i, 1), X(i, 2), 'ro');
    end
    if (cluster==3)
        plot(X(i, 1), X(i, 2), 'go');
    end
    if (cluster==4)
        plot(X(i, 1), X(i, 2), 'ko');
    end
    if (cluster==5)
        plot(X(i, 1), X(i, 2), 'mo');
    end
    hold on;
end
axis([-6 6 -6 6])
title('Original Data and Estimated PDFs');
set(gcf,'color','white') % white background

figure(3);
plot(objectPlot(:,2), objectPlot(:,1), 'bo');
title('Objective function plot');
```

A support .m file should also be included:

gaussianND.m :

```
function [ pdf ] = gaussianND(X, mu, Sigma)

n = size(X, 2);

meanDiff = bsxfun(@minus, X, mu);

pdf = 1 / sqrt((2*pi)^n * det(Sigma)) * exp(-1/2 * sum((meanDiff *
inv(Sigma) .* meanDiff), 2));

end
```

Problem 2

(1) matlab code for part 1

```
clear;
Xtest=importdata('ratings_test.txt');
Xtrain=importdata('ratings.txt');
Xname=importdata('movies.txt');
Xpredict=zeros(5000,3);
RMSEplot=zeros(100,1);
LJLplot=zeros(100,1);

lamda=10;
sigma=0.5;
d=20;

N1=943;
N2=1682;
M=zeros(N1,N2);
U=zeros(N1,d);
V=zeros(d,N2);
I=eye(d);

for i=1:95000
   M(Xtrain(i,1),Xtrain(i,2))=Xtrain(i,3);
end

iteration=100;

%initializtion of Vj
V=normrnd(3,1,d,N2);%²úÉú¾ùÖµÎªa¡¢·½²îÎªb´óÐ¡ÎªcXdµÄËæ»ú¾ØÕó

for iter=1:iteration
   for i=1:N1
      sum1=zeros(d,d);
      sum2=zeros(d,1);
      for j=1:N2
         if (M(i,j)~=0)
            sum1=sum1+V(:,j)*transpose(V(:,j));
            sum2=sum2+M(i,j)*V(:,j);
         end
      end
      U(i,:)=pinv(lamda*sigma*sigma*I+sum1)*sum2;

   end
   for j=1:N2
```

```
        sum1=zeros(d,d);
        sum2=zeros(d,1);
        for i=1:N1
            if (M(i,j)~=0)
                sum1=sum1+transpose(U(i,:))*U(i,:);
                sum2=sum2+transpose(M(i,j)*U(i,:));
            end
        end
        V(:,j)=transpose(pinv(lamda*sigma*sigma*I+sum1)*sum2);

    end

    tmpSum=0;
    for i=1:5000
        Xpredict(i,3)=round(U(Xtrain(i,1),:)*V(:,Xtrain(i,2)));

tmpSum=tmpSum+(Xpredict(i,3)-Xtest(i,3))*(Xpredict(i,3)-Xtest(i,3));
    end

    RMSE=sqrt(tmpSum/5000);
    RMSEplot(iter,1)=RMSE;

    LJLsum1=tmpSum/(2*sigma*sigma);

    LJLsum2=0;
    for i=1:N1
        LJLsum2=LJLsum2+norm(U(i,:))^2;
    end
    LJLsum2=LJLsum2*lamda/2;

    LJLsum3=0;
    for j=1:N2
        LJLsum3=LJLsum3+norm(V(:,j))^2;
    end
    LJLsum3=LJLsum3*lamda/2;
    LJLsum=LJLsum1+LJLsum2+LJLsum3;
    LJLplot(iter,1)=-LJLsum;

end

% for i=1:5000
%
Xpredict(Xtrain(i,1),Xtrain(i,2))=round(U(Xtrain(i,1),:)*V(:,Xtrain(i
,2)));
```

```
% end

% for i=1:5000
%    sum=sum+(Xpredict(i,3)-Xtest(i,3))*(Xpredict(i,3)-Xtest(i,3));
% end

figure(1)
plot(RMSEplot);

figure(2)
plot(LJLplot);
```

(2) matlab code for part 2

```
N2=1682;
randomMovie=round(N2*rand(1,3));

distanceMatrix=zeros(N2,3);
distanceMatrixSort=zeros(N2,3);
resultIndex=zeros(5,3);

for j=1:N2
    for i=1:3
        sum=0;
        for k=1:N2
            sum=sum+norm(V(:,j)-V(:,k))^2;
        end
        distanceMatrix(j,i)=sum;
    end
end

for j=1:3
    distanceMatrixSort(:,j)=sort(distanceMatrix(:,j));
    for i=1:5
        [x,y]=find(distanceMatrix(:,j)==distanceMatrixSort(i+1,j));
        resultIndex(i,j)=x;
    end
end

disp('3 movies randomly chosen and their nearest 5 neighbors are listed
below : ')

for i=1:3
```

```
    disp(i);
    disp(Xname(randomMovie(1,i),1));
    disp(' : ');
    for j=1:5
        disp(Xname(resultIndex(j,i),1));
    end
end
```

(3) matlab code for part 3

```
X = U;

%3 Initiation of the dataset

m = size(X, 1);

k = 30;
n = 20;  % vector lengths.

% k random initial means
indeces = randperm(m);
mu = X(indeces(1:k), :);

sigma = [];

% take overal covariance of the dataset as initial variance for each
cluster.
for j = 1 : k
    sigma{j} = cov(X);
end

% assign equal prior probabilities to each cluster.
phi = ones(1, k) * (1 / k);

% 4: Expectation Maximization

W = zeros(m, k);

% iteration=20

for iter = 1:20

    fprintf('Iteration %d\n ', iter);
    pdf = zeros(m, k);

    % For each cluster...
    for j = 1 : k

        % Evaluate the Gaussian for all data points for cluster 'j'.
        pdf(:, j) = gaussianND(X, mu(j, :), sigma{j});
    end

    % multiply each pdf value by the prior probability for cluster.
    pdf_w = bsxfun(@times, pdf, phi);

    W = bsxfun(@rdivide, pdf_w, sum(pdf_w, 2));

    % do maximization

    % For each of the clusters...
    for j = 1 : k

        % Calculate the prior probability for cluster 'j'.
        phi(j) = mean(W(:, j), 1);
        mu(j, :) = weightedAverage(W(:, j), X);
        sigma_k = zeros(n, n);
        Xm = bsxfun(@minus, X, mu(j, :));

        % Calculate the contribution of each training example to the
covariance matrix.
        for i = 1 : m
            sigma_k = sigma_k + (W(i, j) .* (Xm(i, :)' * Xm(i, :)));
        end

        % Divide by the sum of weights.
        sigma{j} = sigma_k ./ sum(W(:, j));
    end

%     objectValue=0;
%     for i=1:500
%         tmpValue=0;
%         for j=1:k
%             tmpValue=tmpValue+W(i,j)*phi(j);
%         end
%         log(tmpValue)
%         objectValue=objectValue+log(tmpValue);
%    end

   objectValue=0;
   for i=1:500
      objectValue=objectValue+log(max(max(W(i,:))));
   end

   objectPlot(iter,1)= objectValue;
   objectPlot(iter,2)= iter;
%    figure(iter+2);
%    plot(objectPlot(:,2), objectPlot(:,1), 'bo');

   %fprintf(sigma{j});

end

randomCentroid=round(k*rand(1,5));
selectedCentroids=zeros(5,20);
multiplication=zeros(N2,5);
multiplicationSort=zeros(N2,5);
movieIndex=zeros(10,5);


%randomCentroid(:)

for i=1:5
   selectedCentroids(i,:)=mu(randomCentroid(1,i),:);
end

for i=1:5
   for j=1:N2
      multiplication(j,i)=U(i,:)*V(:,j);
   end
end


for j=1:5
   multiplicationSort(:,j)=sort(multiplication(:,j));
   for i=1:10
      [x,y]=find(multiplication(:,j)==multiplicationSort(N2-i+1,j));
      movieIndex(i,j)=x;
   end
end
```

```
disp('5 centroid randomly chosen £¬their indexes are: (among 30
cluster)');

disp(randomCentroid(:));

disp('Their closest 10 movies are listed below : ');

for i=1:5
    disp('Centroid ');
    disp(i);
%    disp('the centroid coordinate is showed below: ')
%    disp(mu(randomCentroid(1,i),:)*100000);
    disp(' 10 movies are : ');
    for j=1:10
        disp(Xname(movieIndex(j,i),1));
    end
end
```

(4) A support .m file should also be included:

gaussianND.m :

```
function [ pdf ] = gaussianND(X, mu, Sigma)

n = size(X, 2);

meanDiff = bsxfun(@minus, X, mu);

pdf = 1 / sqrt((2*pi)^n * det(Sigma)) * exp(-1/2 * sum((meanDiff *
inv(Sigma) .* meanDiff), 2));

end
```