Homework 5

Zachary DeStefano, 15247592

CS 273A: Winter 2015

Due: March 10, 2015

Problem 1

Part a

The data does not look very clustered. Here is the code to load the data and make an initial plot.

```
%%
load('data/iris.txt');
X = iris(:,1:2);
Y = iris(:,5);

%%

%Part A
plot(X(:,1),X(:,2),'ro')
```

Part b

I tried a few different initializations. I tried the 3 different ones available in the kmeans function as well as my own initialization points. For $k = 5$, I arranged 5 points in an X-shape in the x_1, x_2 space. For $k = 20$, I did two different initial arrangements: a $4x5$ grid and a $5x4$ grid in the x_1, x_2 space. For $k = 5$ the lowest score came from my initialization. For $k = 20$ the lowest score also came from my initialization, specifically the $5x4$ initialization, although the $4x5$ grid had nearly the same score.

Here is the best 5-means clustering I was able to achieve

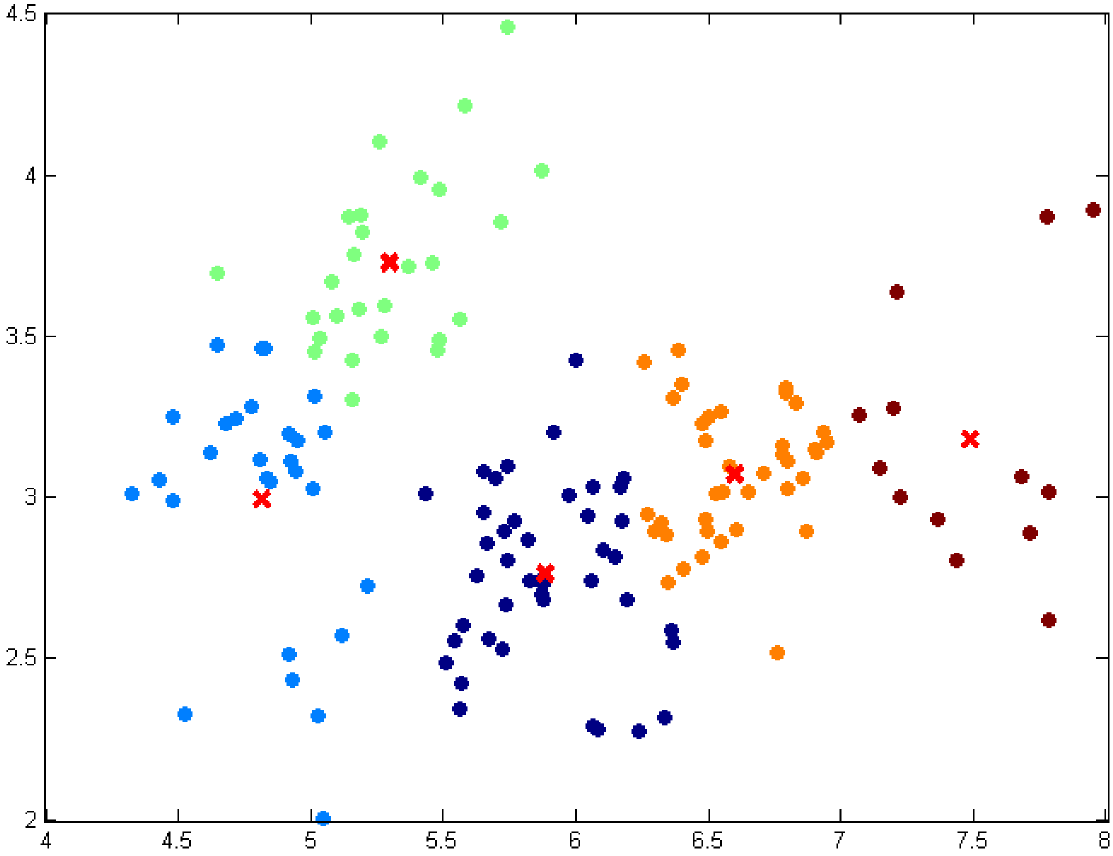

Figure 1: $k = 5$ clustering with x marks for cluster centers

Here is the best 20-means clustering I was able to achieve.

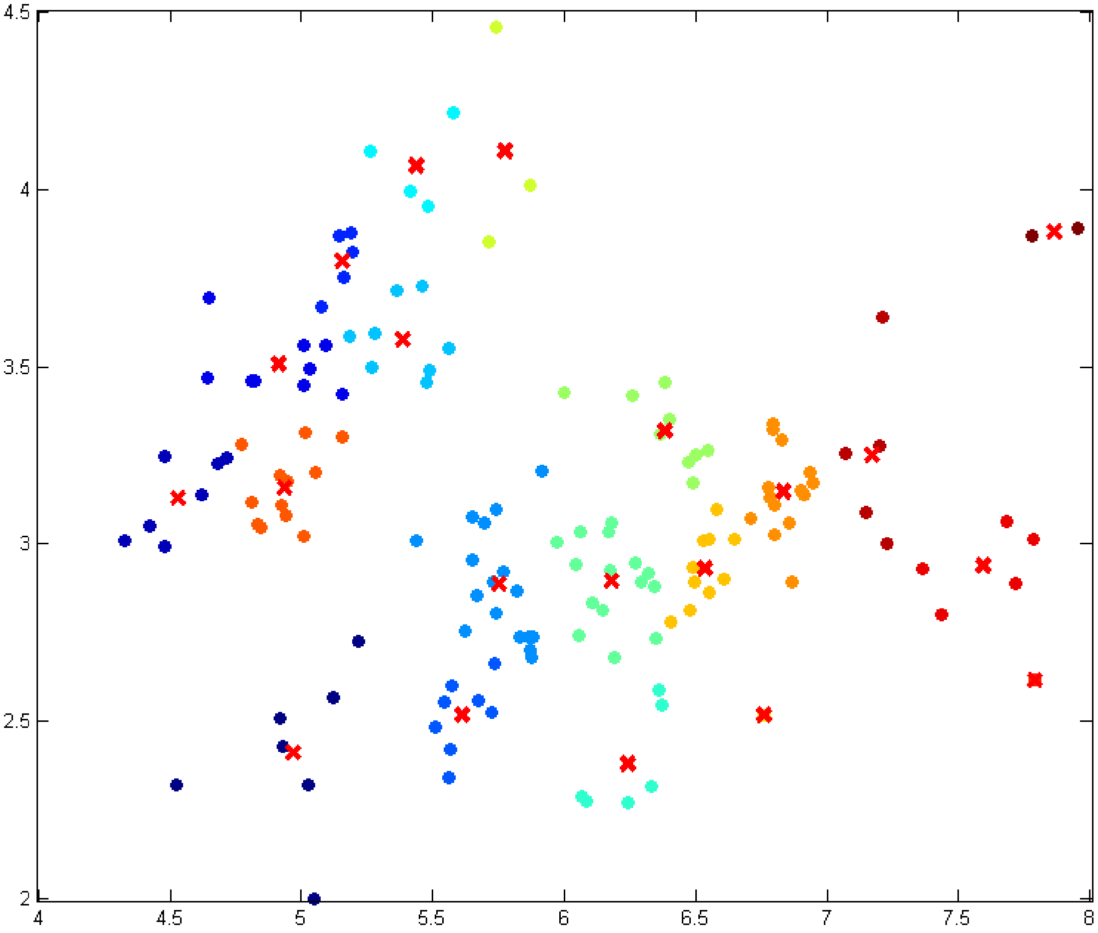

Figure 2: $k = 20$ clustering with x marks for cluster centers

Here is the code for Part B.

```
%%

%Part B

%change this depending on whether we are doing k=5 or k=20
k=5;
%k=20;

minX1 = min(X(:,1));
minX2 = min(X(:,2));
maxX1 = max(X(:,1));
maxX2 = max(X(:,2));

%if k=5, make 5 initial points arranged in X-shape
if(k==5)
    centerX1 = (minX1 + maxX1)/2; centerX2 = (minX2 + maxX2)/2;
    firstQuatX1 = minX1 + (maxX1-minX1)/4;
    thirdQuatX1 = maxX1 - (maxX1-minX1)/4;
    firstQuatX2 = minX2 + (maxX2-minX2)/4;
    thirdQuatX2 = maxX2 - (maxX2-minX2)/4;
    centerPt = [centerX1 centerX2];
    Pt11 = [firstQuatX1 firstQuatX2];
    Pt13 = [firstQuatX1 thirdQuatX2];
    Pt31 = [thirdQuatX1 firstQuatX2];
    Pt33 = [thirdQuatX1 thirdQuatX2];
    initPts5 = [centerPt;Pt11;Pt13;Pt31;Pt33];
end

%if k=20, make 20 points in 4x5 and 5x4 arrangement
if(k==20)
    fifthX1 = (maxX1-minX1)/5;
    sixthX1 = (maxX1-minX1)/6;
    fifthX2 = (maxX2-minX2)/5;
    sixthX2 = (maxX2-minX2)/6;
    initPts20A = zeros(20,2);
    initPts20B = zeros(20,2);
    index = 1;
    for i = 1:5
       for j = 1:4
          curX1A = minX1 + fifthX1*j;
          curX2A = minX2 + sixthX2*i;
          initPts20A(index,:) = [curX1A curX2A];

          curX1B = minX1 + sixthX1*i;
          curX2B = minX2 + fifthX2*j;
          initPts20B(index,:) = [curX1B curX2B];

          index = index+1;
       end
    end
end
```

```
%%
%run k-Means with the different initializations
[z1,c1,score1] = kmeans(X,k,'random');
[z2,c2,score2] = kmeans(X,k,'farthest');
[z3,c3,score3] = kmeans(X,k,'k++');

if(k==5)
   [z4,c4,score4] = kmeans(X,k,initPts5);
end

if(k==20)
    [z5,c5,score5] = kmeans(X,k,initPts20A);
    [z6,c6,score6] = kmeans(X,k,initPts20B);
end

%%

%best score for k=5 was with z4
if(k==5)
   z = z4;
   c = c4;
end
if(k==20)
   z = z6;
   c = c6;
end
figure
plotClassify2D([],X,z)
hold on
plot(c(:,1),c(:,2),'rx','MarkerSize',10,'LineWidth',3);
```

Part c

Here are the plots of agglomerative clustering with 5 clusters.

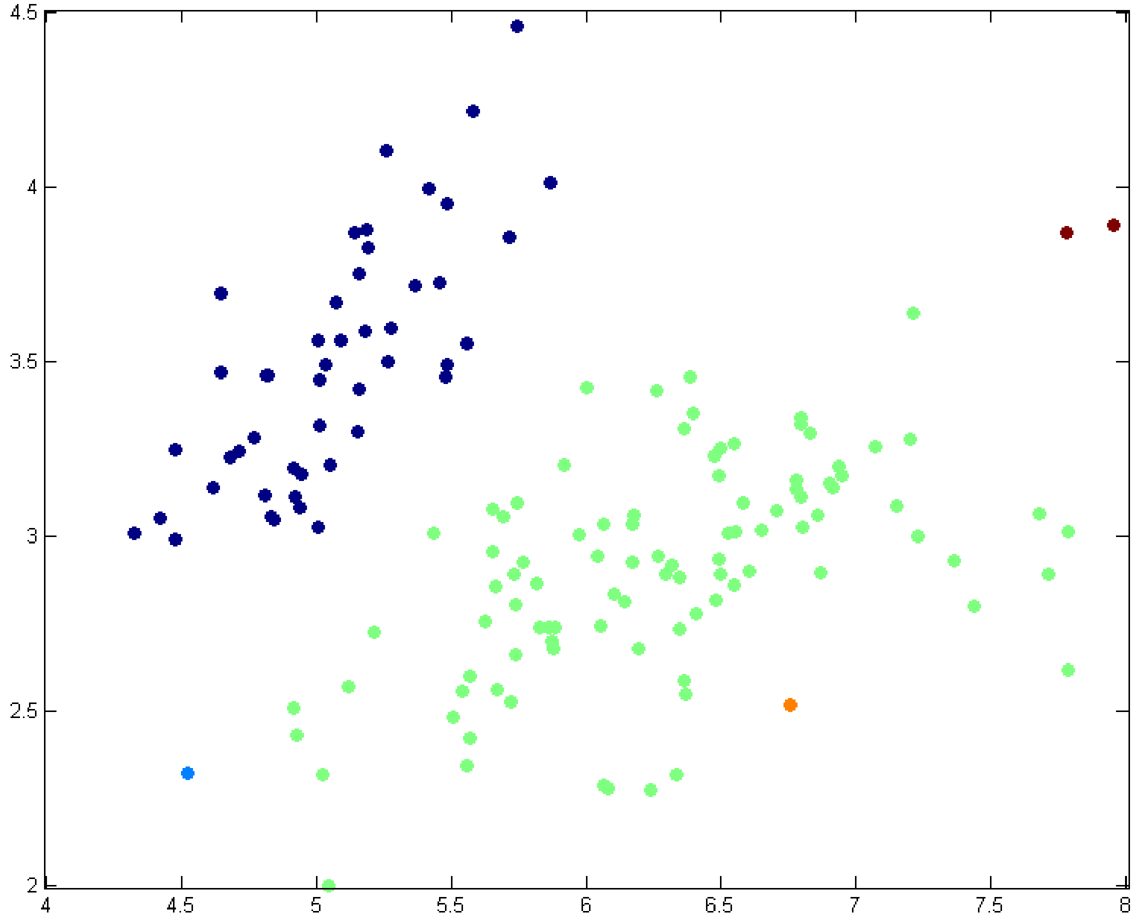

Figure 3: Agglomerative clustering with 5 clusters, single linkage

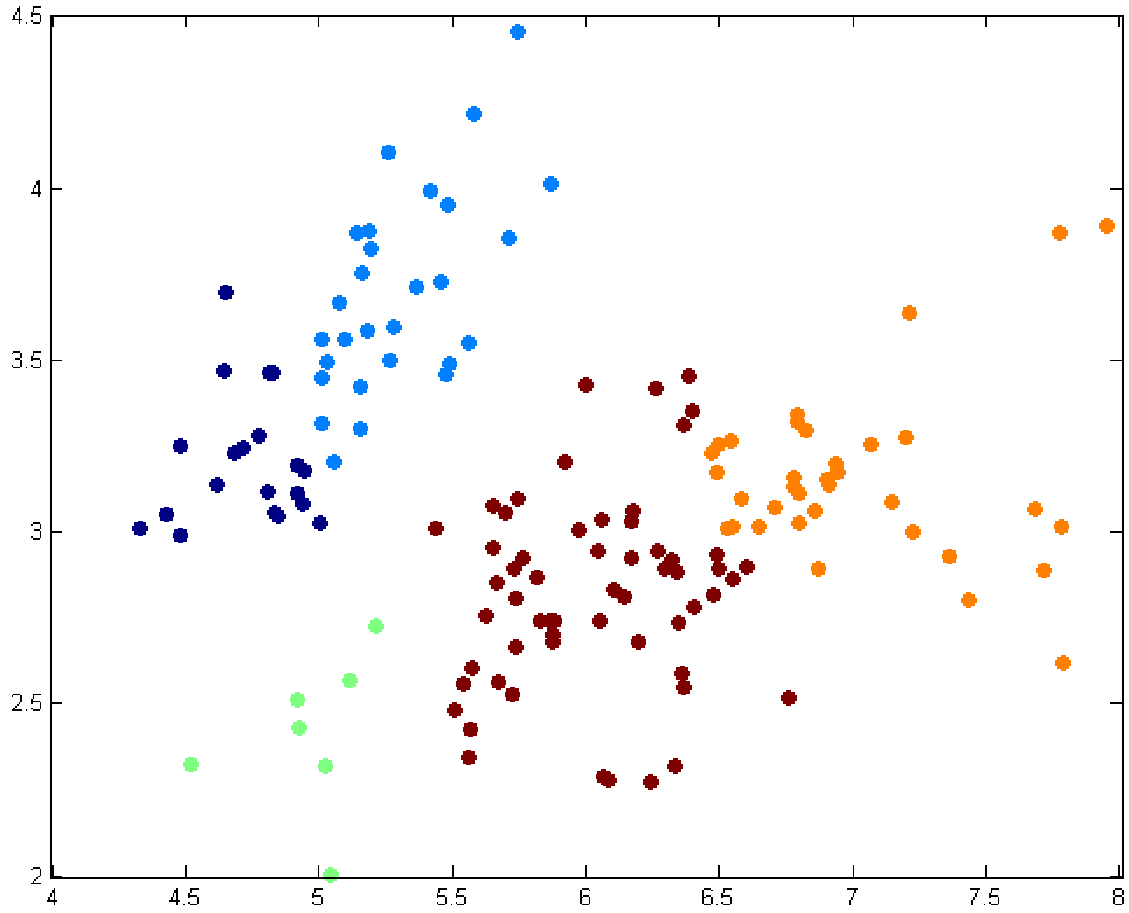

Figure 4: Agglomerative clustering with 5 clusters, complete linkage

Here are the plots of agglomerative clustering with 20 clusters.

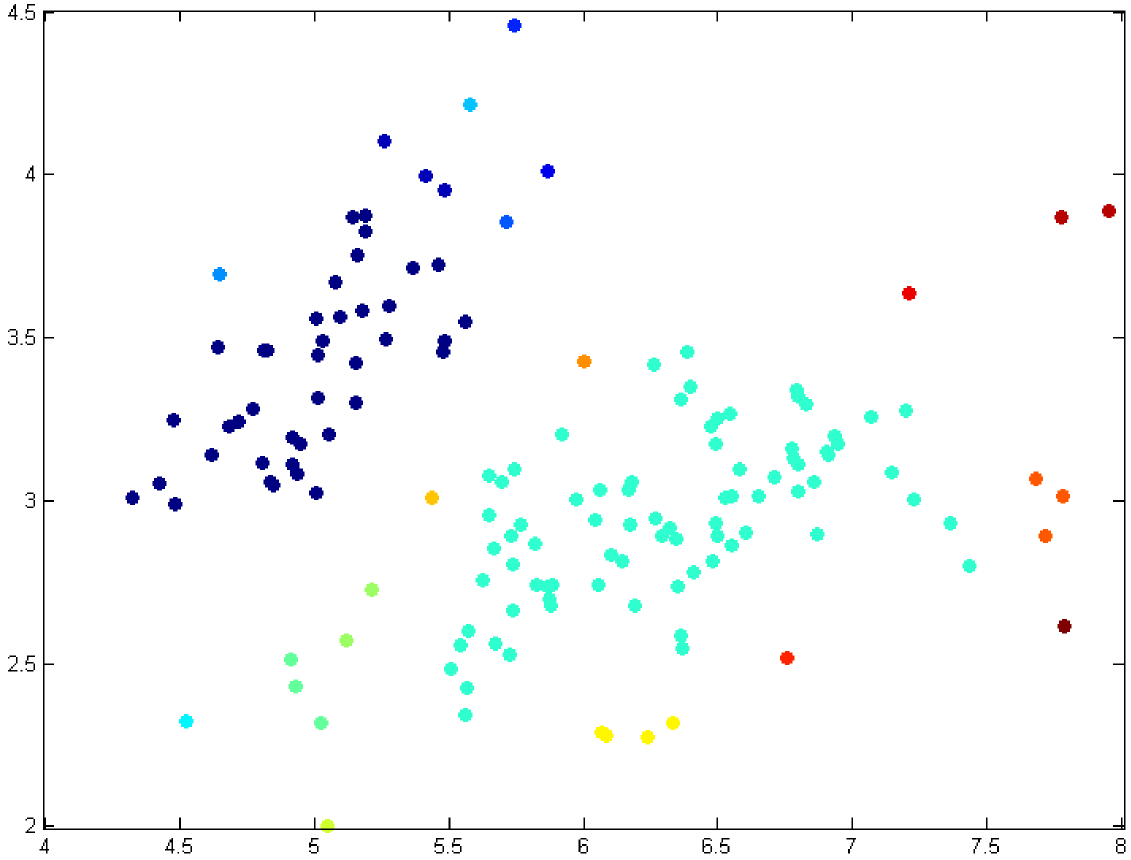

Figure 5: Agglomerative clustering with 20 clusters, single linkage

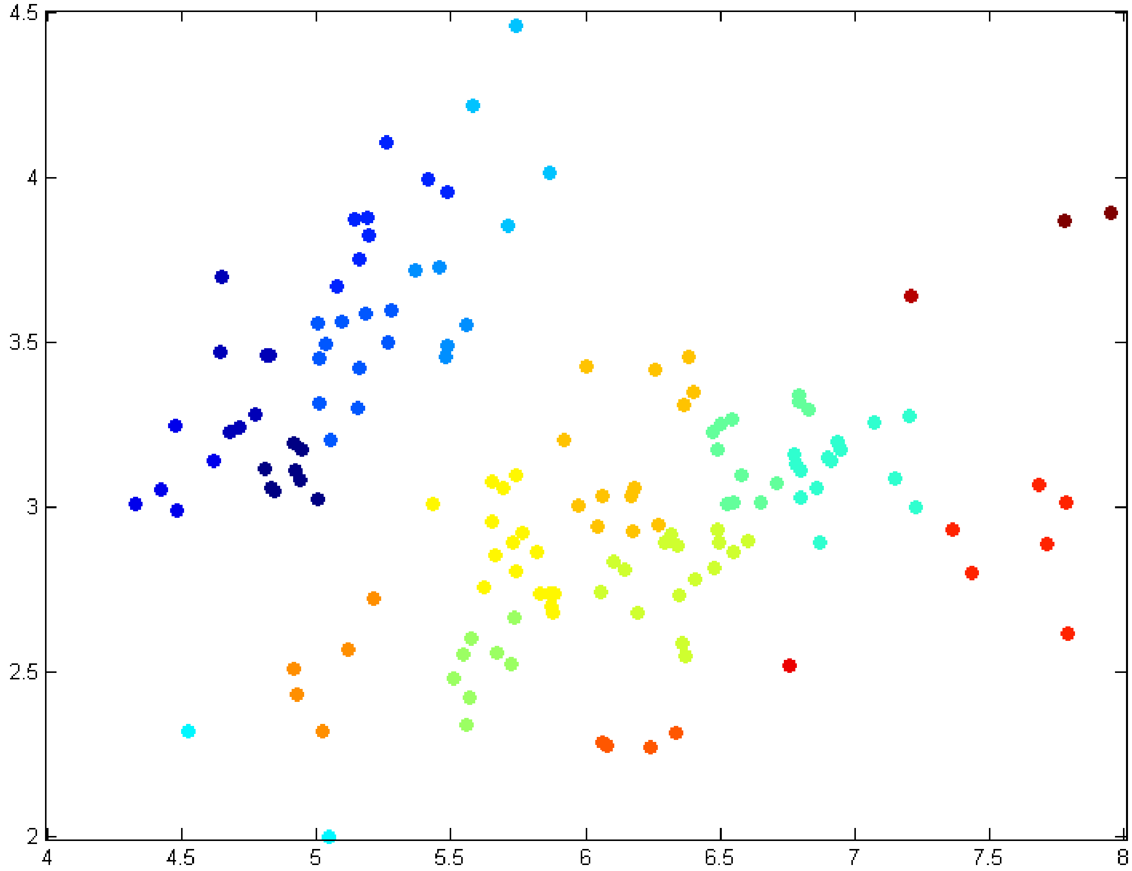

Figure 6: Agglomerative clustering with 20 clusters, complete linkage

As can be observed, single linkage tries to group the points into as few clusters as possible whereas complete linkage spreads out the clustering. The results for complete linkage look quite similar to k-Means.

Here is the code for Part C

```
%%

%hierarchical aggolomorative clustering
k=5; %change this between 5 and 20
z = agglomCluster(X,k,'min');%single linkage
plotClassify2D([],X,z)

z = agglomCluster(X,k,'max');%complete linkage
plotClassify2D([],X,z)
```

Part d

Here is the plot of 5 clusters found using EM

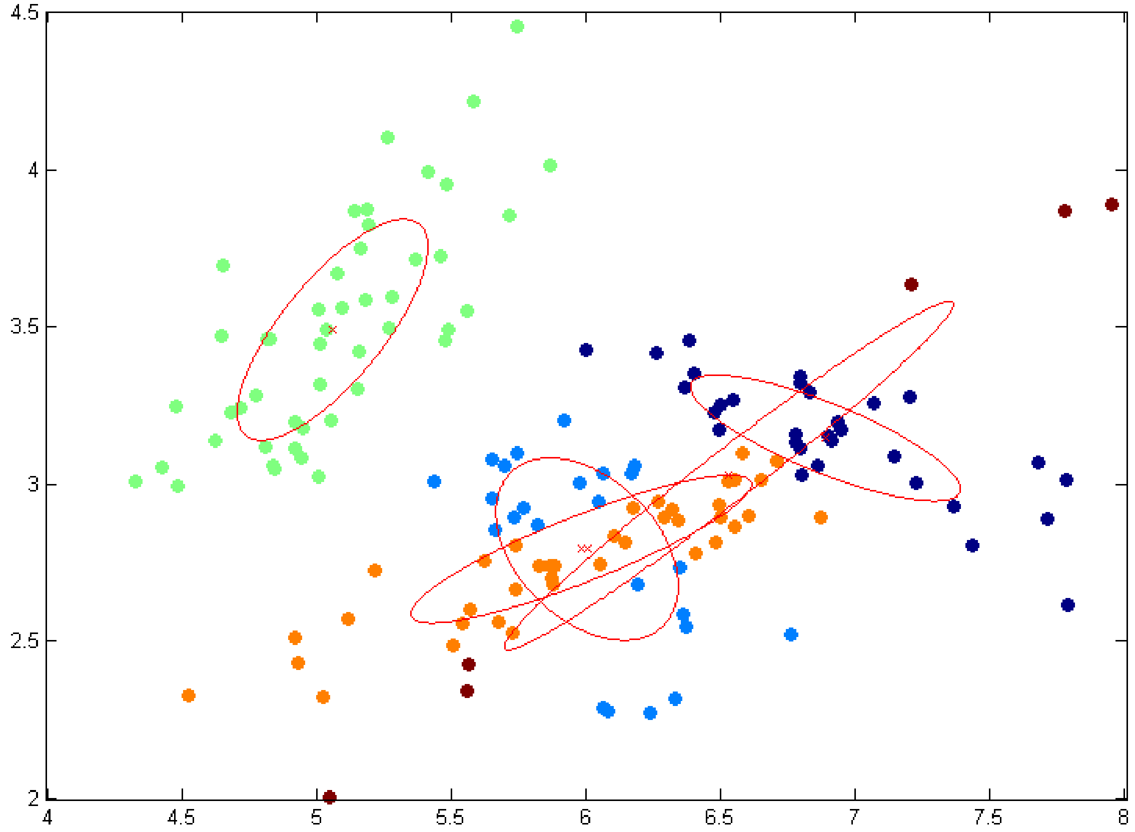

Figure 7: EM run to figure out 5 clusters

Here is the plot of 20 clusters found using EM

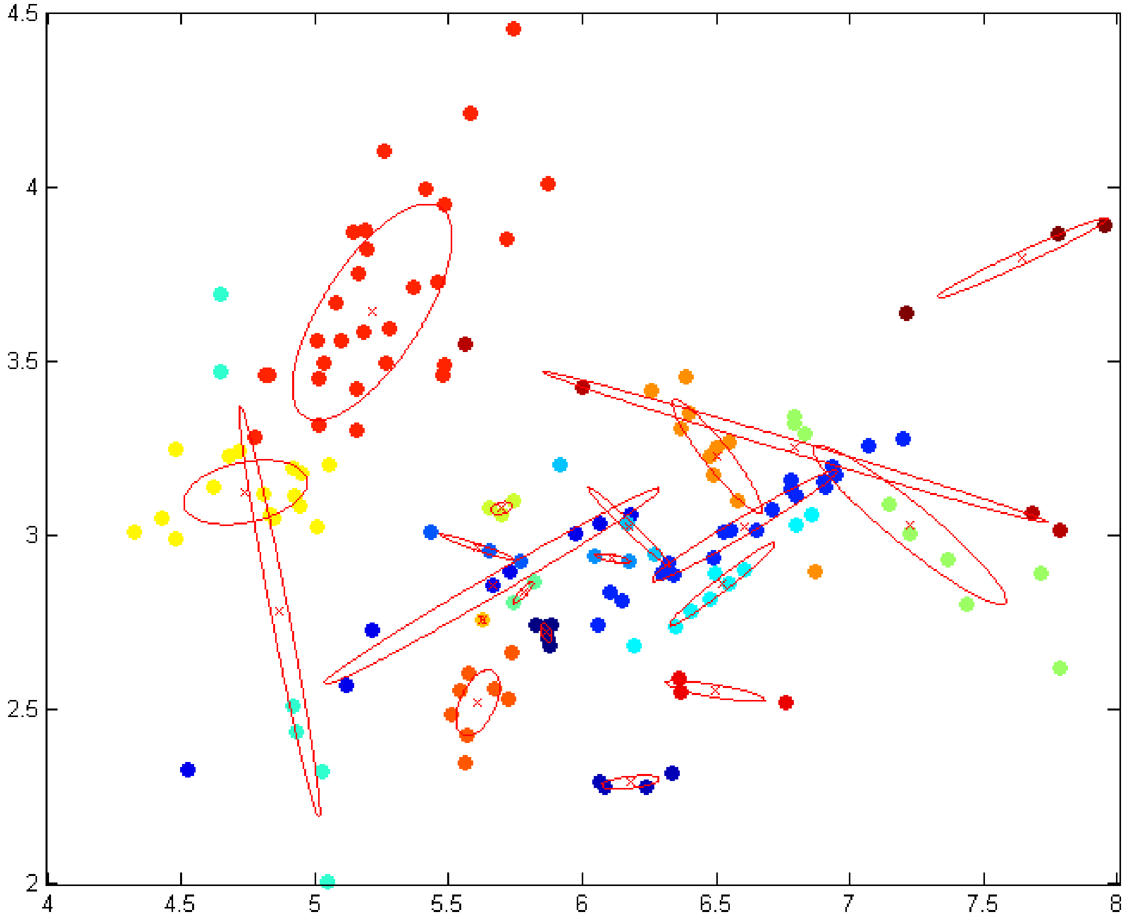

Figure 8: EM run to figure out 20 clusters

I used the same initializations as k-means, including the points I constructed. The score I used was the log likelihood. With this score, the higher the better. The best log likelihood I achieved occurred when I specified points. This is similar to what happened when I did k-Means.

Overall, k-Means clustering seems to be the most reasonable as the clusters look the most logical to me. The EM clustering does have the advantage of being able to discern clusters that end up sharing regions of the Euclidean space.

Here is the code for Part D. It relies on the code from Part A and B.

```
%%
k=5;
%k=20;
[z1,T1,~,score1] = emCluster(X,k,'random');
[z2,T2,~,score2] = emCluster(X,k,'farthest');
[z3,T3,~,score3] = emCluster(X,k,'k++');

if(k==5)
   [z4,T4,score4] = emCluster(X,k,initPts5);
end

if(k==20)
    [z5,T5,score5] = emCluster(X,k,initPts20A);
    [z6,T6,score6] = emCluster(X,k,initPts20B);
end
%%
z=z4;T=T4; %for 5 clusters
%z=z5;T=T5; %when there are 20 clusters
plotClassify2D([],X,z)
hold on
for i=1:k
   plotGauss2D(T.mu(i,:),T.Sig(:,:,i),'r');
end
```

Problem 2

Part a

The final k-Means cost after one run ends up being 2.0369

Part b

The k-Means cost for the next 4 runs were the following (in order of run):
2.4407
2.0955
2.4191
2.0324

Since the last one had the smallest cost, I will use that clustering

Here is the code for Part A and B

```
% Read in vocabulary and data (word counts per document)
[vocab] = textread('data/text/vocab.txt','%s');
[did,wid,cnt] = textread('data/text/docword.txt','%d%d%d','headerlines',3);
X = sparse(did,wid,cnt); % convert to a matlab sparse matrix
D = max(did); % number of docs
W = max(wid); % size of vocab
N = sum(cnt); % total number of words
% It is often helpful to normalize by the document length:
Xn= X./repmat(sum(X,2),[1,W]) ; % divide word counts by doc length

for i=1:size(Xn,1)
   [sorted,order] = sort( Xn(i,:), 2, 'descend');
    fprintf('Doc %d: ',i); fprintf('%s ',vocab{order(1:10)}); fprintf('\n');
end
%%

%Part A and B. scores(1) is the result for Part A.
K=20;
numTrials = 5;
scores = zeros(1,numTrials);
bestScore = Inf;
for j=1:numTrials
    [zCur,cCur,score] = kmeans(Xn,K);
    scores(j) = score;
    if(score < bestScore)
       z = zCur;
       c = cCur;
       bestScore = score;
    end
end
```

Part c

This is the number of articles per cluster (line *i* corresponds to number of documents in cluster *i*):

```
1
1
1
54
16
1
38
5
6
2
2
1
7
1
3
2
13
12
1
35
```

Here is a histogram visual of this data

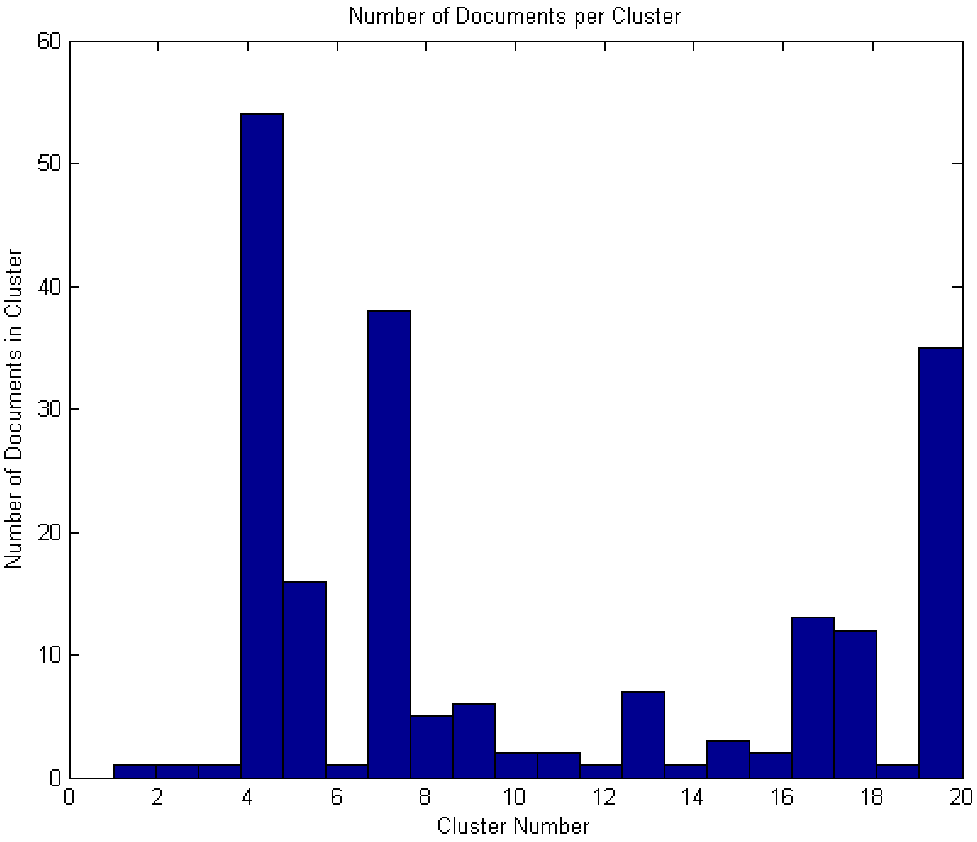

Figure 9: Histogram of Number of Articles Per Cluster

Here are the first most likely words for each cluster:

Cluster 1: laredo border international mexico river american city green mayor rio
Cluster 2: feet square broadway seventh street side times avenue block million
Cluster 3: test end houston 000 0101 0102 100 1900 1900s 1968
Cluster 4: millennium city night fireworks times 2000 midnight yeltsin russian friday
Cluster 5: y2k 2000 problems computer koskinen saturday computers system problem reported
Cluster 6: warrick heisman florida award guy national championship college field game
Cluster 7: american president national america war sports country 000 young white
Cluster 8: century book week school finds photographs 000 sales lives war
Cluster 9: game games goal million play going fortson team elias bruins
Cluster 10: home stay fathers children group child kids working called com
Cluster 11: archbishop bishop cardinal york church close leader late served american
Cluster 12: 2000 computer problem systems city failure officials 000 100 ahead
Cluster 13: bowden vick bowl coach florida game tech virginia football yards
Cluster 14: iqbal police 100 children apartment later month news story thursday
Cluster 15: season giants rusie game team 000 coach games players free
Cluster 16: buses authority diesel natural gas plan mta city york hybrid
Cluster 17: bradley candidates campaign mccain hampshire bush political president republican voters
Cluster 18: square times city economy york millennium 000 2000 mexico party
Cluster 19: lott players union executive player nfl think case challenge director
Cluster 20: team game season league players coach games play win teams

Some of these seem to be interpretable sets. Cluster 17 seems to be politics related articles. Cluster 19 seems to be sports related articles. Other clusters though, such as cluster 3, do not seem to have a coherent theme.

Here is the code to complete Part C. It relies on the code from the previous part:

```
%%

%Part C

%gets the number of documents per cluster
[numDocsPerCluster,Clusters] = hist(z,unique(z));

%prints out the clusters
for i=1:K
   [~,orderI] = sort( c(i,:), 'descend');
   fprintf('Cluster %d: ',i); fprintf('%s ',vocab{orderI(1:10)}); fprintf('\n');
end
%%
%prints out the histogram of num docs per cluster
hist(z,20)
xlabel('Cluster Number');
ylabel('Number of Documents in Cluster');
title('Number of Documents per Cluster');
```

Part d

After printing the documents related to document 1, these seem to all be related to sports.

After printing the documents related to document 15, they seem to all be related to the Y2K phenomenon.

After printing the documents related to document 30, they do not seem to have a discernable theme.

Here is the code for Part D

```
%%

%Part D
%gets the assignments for docs 1,15,30
docNums = [1 15 30];
assignments = zeros(1,3);
docsWithSameCluster = cell(1,3);
for i = 1:3
   assignments(i) = z(docNums(i));
   docsWithSameCluster{i} = find(z == assignments(i));
end

for cluster = 1:3
    currentDocs = docsWithSameCluster{cluster};
   for doc = 1:min(12,length(currentDocs));
       curDocNum = currentDocs(doc);
       fname = sprintf('data/text/example1/20000101.%04d.txt',curDocNum);
        txt = textread(fname,'%s',10,'whitespace','\r\n');
        fprintf('%s\n',txt{:});
        fprintf('\n');
   end
   fprintf('\n\n\n');
end
```

Part e

The best score when 40 clusters were done was 1.6950. This is a better score than what we achieved when $k = 20$. The cluster related to documents 1 and 15 had the same theme as what we got with 20 clusters. With document 30, there was still not a discernable theme to the documents that were in the same cluster as it. When I first ran the k-Means clustering with $k = 40$, there were 100 documents in the same cluster as document 1 and they did not have a theme. I re-ran k-Means with a different initialization, the k++ method, and I achieved the current result.

Here is the code for Part E

```
%Part E
K=40;
numTrials = 4;
scores = zeros(1,numTrials);
bestScore = Inf;
for j=1:numTrials
    [zCur,cCur,score] = kmeans(Xn,K,'k++');
    scores(j) = score;
    if(score < bestScore)
       z = zCur;
       c = cCur;
       bestScore = score;
    end
end

%gets the assignments for docs 1,15,30
docNums = [1 15 30];
assignments = zeros(1,3);
docsWithSameCluster = cell(1,3);
for i = 1:3
   assignments(i) = z(docNums(i));
   docsWithSameCluster{i} = find(z == assignments(i));
end

for cluster = 1:3
    currentDocs = docsWithSameCluster{cluster};
   for doc = 1:min(12,length(currentDocs));
       curDocNum = currentDocs(doc);
       fname = sprintf('data/text/example1/20000101.%04d.txt',curDocNum);
        txt = textread(fname,'%s',10,'whitespace','\r\n');
        fprintf('%s\n',txt{:});
        fprintf('\n');
   end
   fprintf('\n\n\n');
end
```

Problem 4

Part a and b

Here is the code to complete Part a and b

```
%%
X = load('data/faces.txt'); % load face dataset
i=5;
img = reshape(X(i,:),[24 24]); % convert vectorized datum to 24x24 image patch
imagesc(img); axis square; colormap gray; % display an image patch; you may have to squint

%%
%Part A
mu = mean(X);
muMat = repmat(mu,size(X,1),1);
X_0 = X-muMat;

%%
%Part B
[U,S,V] = svd(X_0);
W = U*S;
```

Part c

Here is the plot of mean-squared error as a function of K

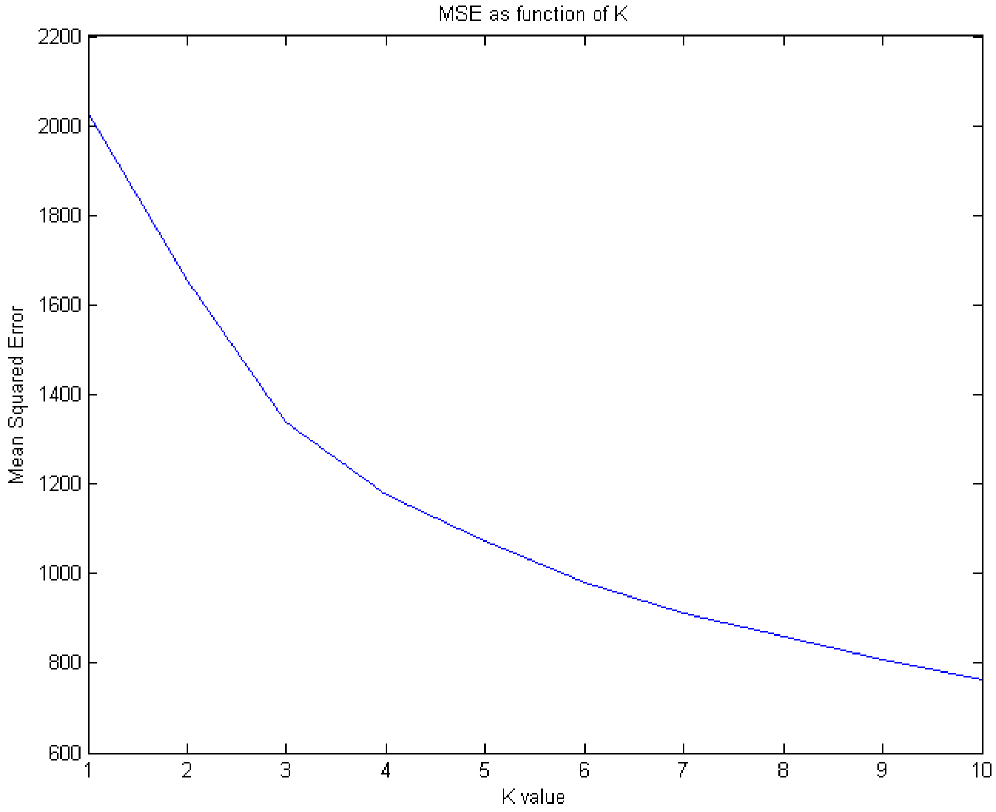

Figure 10: MSE of reconstruction as function of K

Here is the code to complete Part C. It uses the code from Part A and B

```
%%
%Part C
mseSVD = zeros(1,10);
for K=1:10
   X_0hat = W(:,1:K)*(V(:,1:K)');
   mseSVD(K) = mean( mean( (X_0hat-X_0).^2 ) );
end
plot(mseSVD);
xlabel('K value');
ylabel('Mean Squared Error');
title('MSE as function of K');
```

Part d

Here is what the first few principal directions looks like:

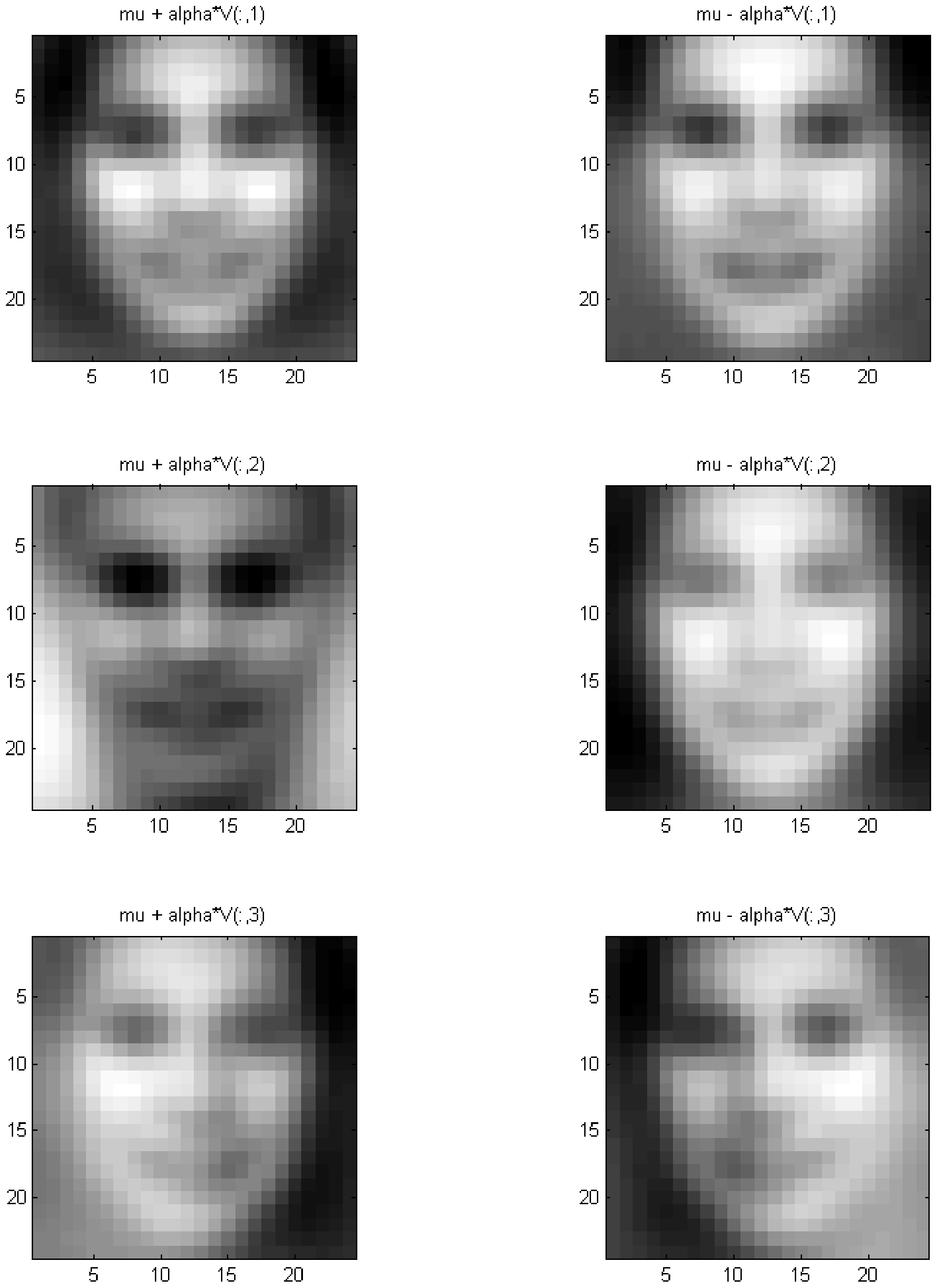

Figure 11: Face image for j=1,2,3

Here is the code I used to get the images in part D. It uses the Part A and B code.

```
%%
%Part D
figure
for j=1:3
    alpha = 2*median(abs(W(:,j)));
    img1 = reshape(mu + alpha*V(:,j)', [24 24]);
    img2 = reshape(mu - alpha*V(:,j)', [24 24]);

    subplot(3,2,2*j-1);
    imagesc(img1); axis square; colormap gray;
    title(strcat('mu + alpha*V(:,',num2str(j),')'));

    subplot(3,2,2*j);
    imagesc(img2); axis square; colormap gray;
    title(strcat('mu - alpha*V(:,',num2str(j),')'));
end
```

Part e

Here are some of the faces shown in their coordinates in the first two directions

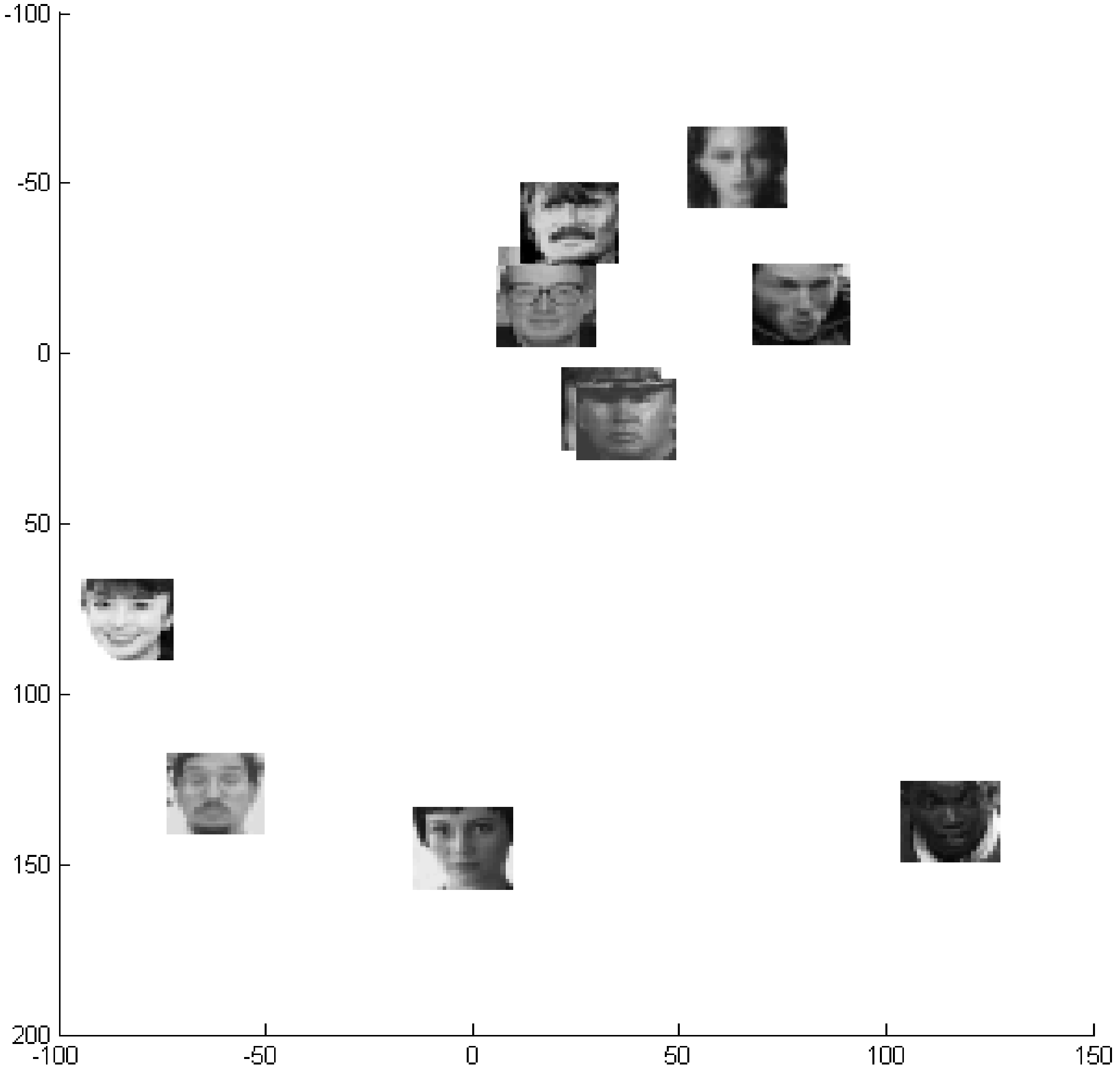

Figure 12: Faces 20-30 displayed with their first two coordinates

Here is the code I used to get the plot for part E. It uses the code from Part A and B.

```
%%

%Part E
idx = 20:30; % pick some data at random or otherwise
figure; hold on; axis ij; colormap(gray);
range = max(W(idx,1:2)) - min(W(idx,1:2)); % find range of coordinates to be plotted
scale = [200 200]./range; % want 24x24 to be visible but not large on new scale
for i=idx, imagesc(W(i,1)*scale(1),W(i,2)*scale(2), reshape(X(i,:),24,24)); end;
```

Part f

Here is the reconstruction of face 10

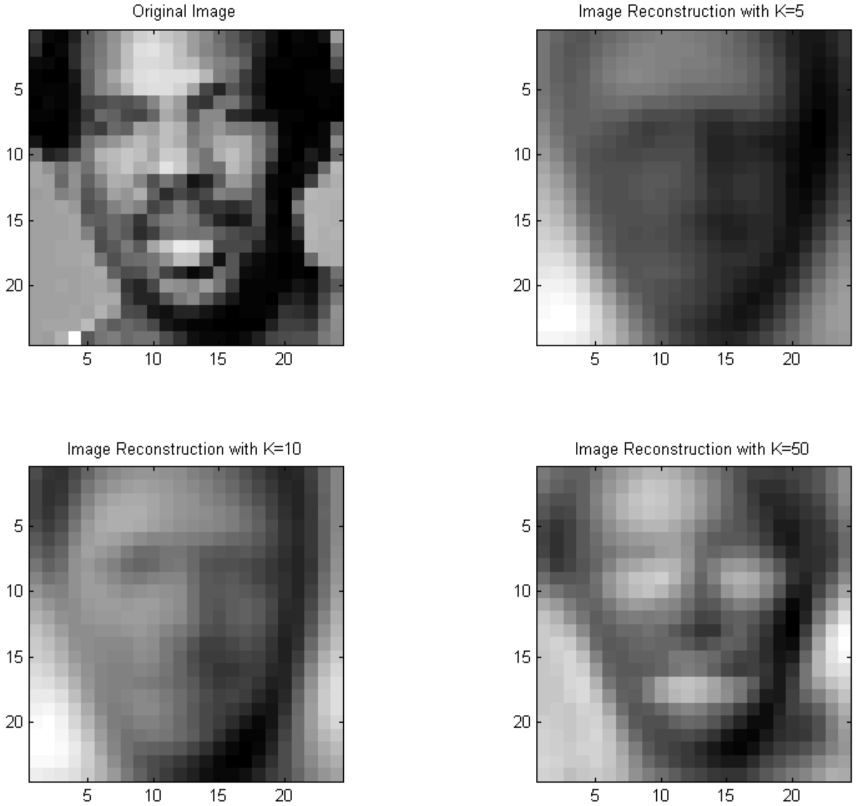
Original Image
Image Reconstruction with K=5
Image Reconstruction with K=10
Image Reconstruction with K=50
5
10
15
20

Figure 13: Face 10 reconstructed from K=5,10,50

Here is the reconstruction of face 15

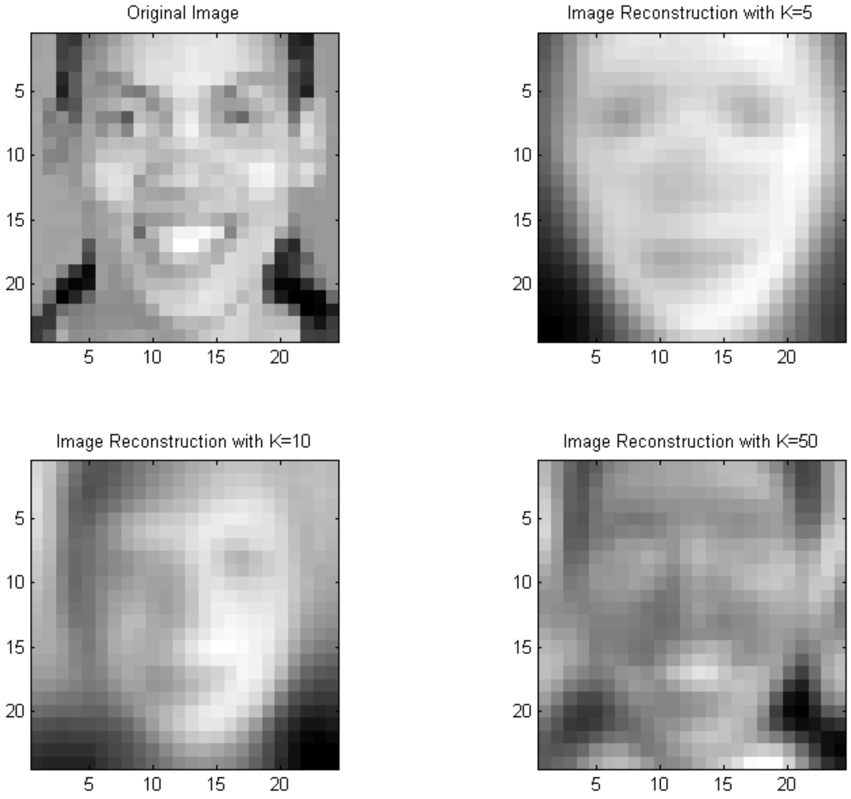
Original Image
Image Reconstruction with K=5
Image Reconstruction with K=10
Image Reconstruction with K=50
5
10
15
20
5
10
15
20

Figure 14: Face 15 reconstructed from K=5,10,50

Here is the code to complete Part F

```
%%

%Part F
%faceNum = 15;
faceNum = 10;
Kvals = [5 10 50];

%display original image
figure
subplot(2,2,1);
imagesc(reshape(X(faceNum,:),[24 24])); axis square; colormap gray;
title('Original Image');

for i = 1:3
    K=Kvals(i);
    X_0hat = W(:,1:K)*(V(:,1:K)');
    imgRecon = reshape(X_0hat(faceNum,:),[24 24]);

    %display image reconstruction
    subplot(2,2,i+1);
    imagesc(imgRecon); axis square; colormap gray;
    title(strcat('Image Reconstruction with K=',num2str(K)));
end
```