ECE175 HW6 Report

Di Guan (A91041815)

Problem 1.

Consider the quadratic function

$$f(\mathrm{x}) = \mathrm{x}^T A\mathrm{x}.$$

(a).

show that the function can be written as

$$f(\mathrm{y}) = \sum_{i=1}^{n} \alpha_i y_i^2$$

where y is a rotation of x, what is the transformation that maps x into y, and what are the coefficients α_i?

Solution:

since $f(\mathrm{x})$ is the quadratic form of matrix A, then A is symmetric matrix. Thus, A can be expressed as orthogonally similar to diagonal matrix, which is $A = PDP^T$ where P is the set of orthonormal eigenvectors of A and D is a diagonal matrix with all the eigenvalues λ_i of A on the diagonal entries.

Thus $f(\mathrm{x}) = \mathrm{x}^T A\mathrm{x} = \mathrm{x}^T PDP^T\mathrm{x}$, denote $y = P^T\mathrm{x}$, the the quadratic form becomes $f(\mathrm{y}) = y^T Dy = \sum_{i=1}^n y_i D_{ii} y_i = \sum_{i=1}^n y_i \lambda_i y_i = \sum_{i=1}^n \lambda_i y_i^2$.

Therefore, $y = P^T\mathrm{x}$ where the columns of P is the orthonormal eigenvectors of A and α_i is all the eigenvalues of A, λ_i

(b).

consider the case in hw2, namely,

$$A = \begin{bmatrix} 5 & b \\ b & 5 \end{bmatrix}$$

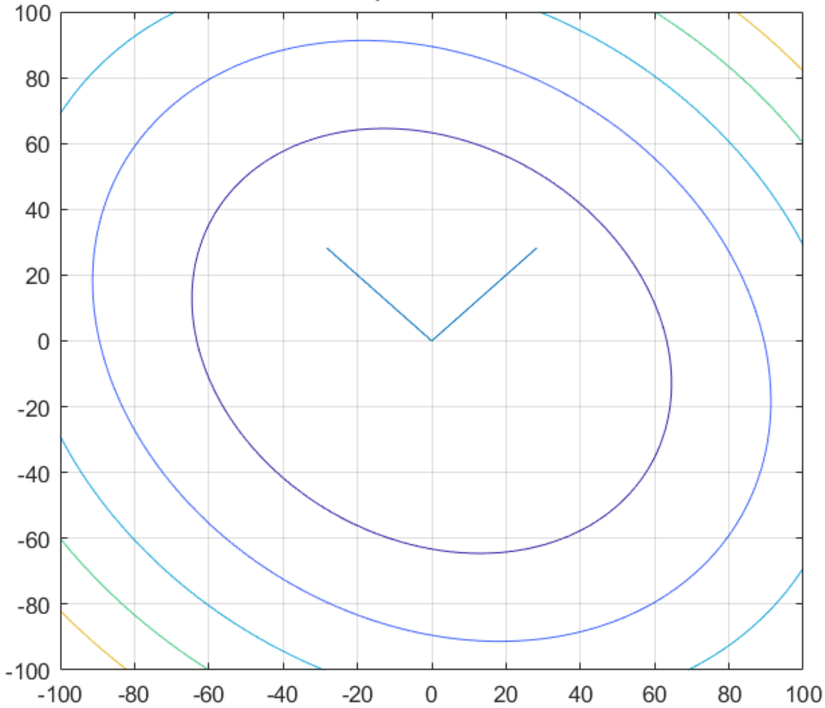
where x is in the range -100 <= x_1 <= 100, -100<= x_2<=100, and $b \in (1, 5, 10)$

what are the vectors that point in the direction of the major axes of the ellipse corresponding to the iso-contours of $f(\mathrm{x})$? make a contour plot of the function and superimpose a plot of the two vectors, for the three values of b.

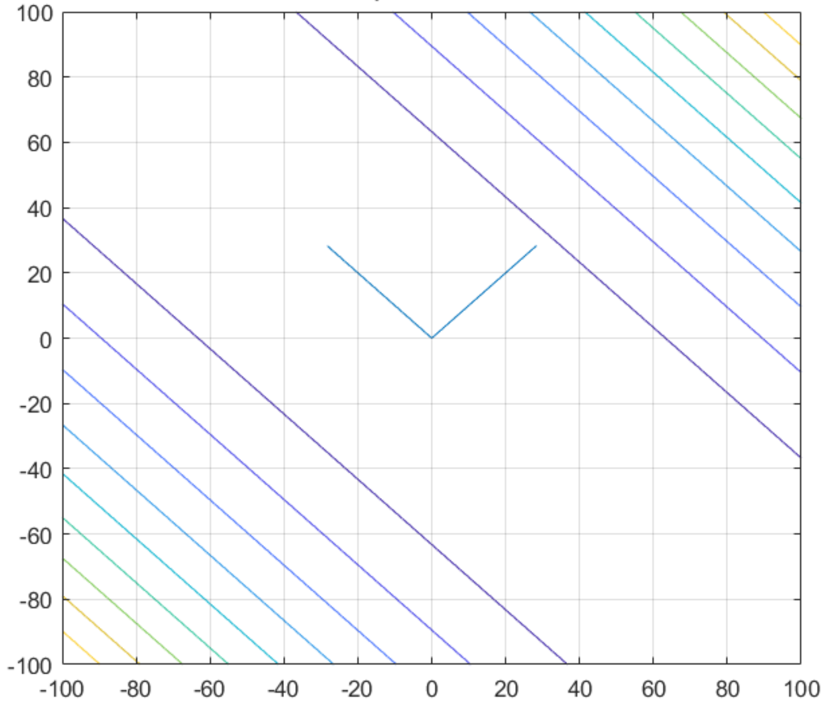
Solution:

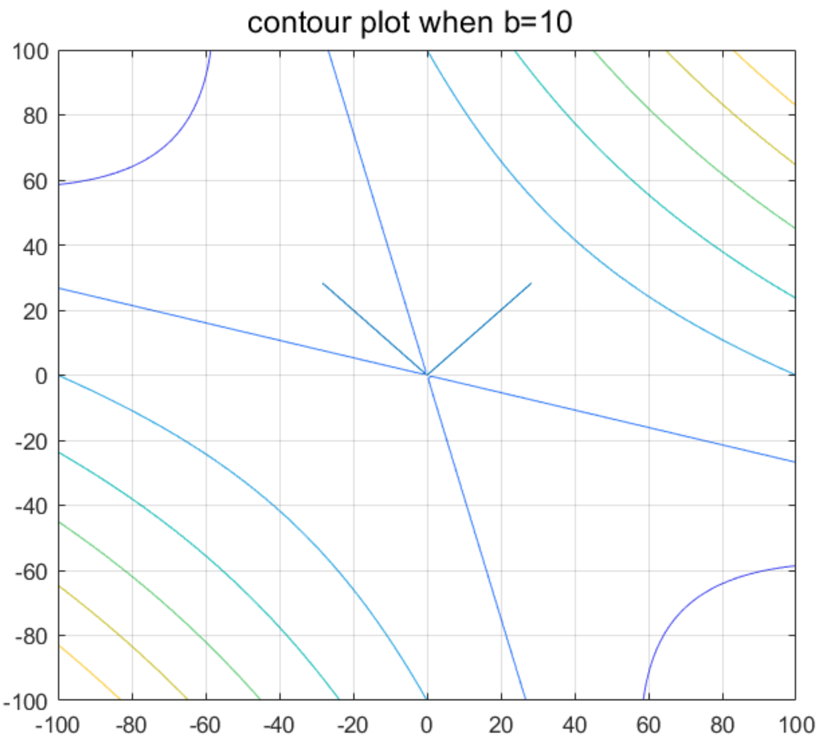
the vectors that point in the direction of the major axes of the ellipse corresponding to the iso-contours of f(x) is the eigenvectors of the matrix A

contour plot when b=1



contour plot when b=5





(c).

for the same matrix, plot a slice through the function $f(\mathrm{y})$ for $y_1 = 0$ and $y_2 = 0$.

Solution:

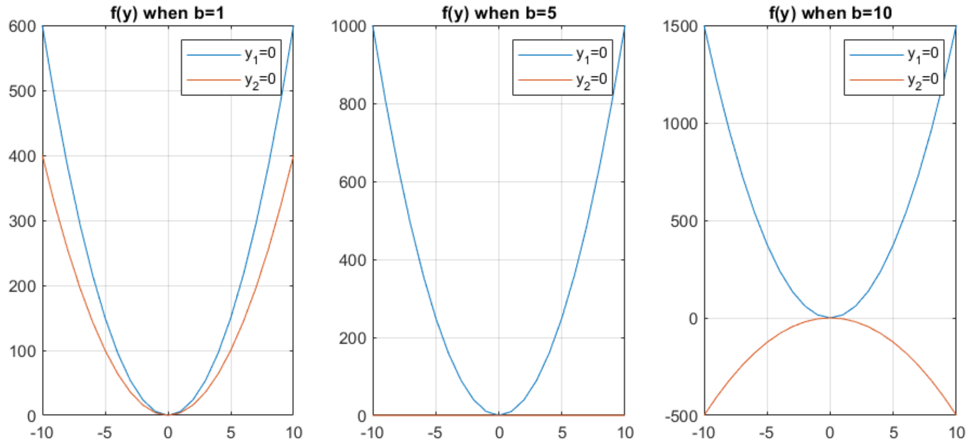
- For $y_1 = 0$, we have

$$f(y) = y^T D y = [0 \quad y_2] \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} 0 \\ y_2 \end{bmatrix} = \lambda_2 y_2^2$$

- For $y_2 = 0$, we have

$$f(y) = y^T D y = [y_1 \quad 0] \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} y_1 \\ 0 \end{bmatrix} = \lambda_1 y_1^2$$

- when $b = 1$, $\lambda_1 = 4$ and $\lambda_2 = 6$
  - $f(0, y_2) = \lambda_2 y_2^2 = 6y_2^2$
  - $f(y_1, 0) = \lambda_1 y_1^2 = 4y_1^2$
- when $b = 5$, $\lambda_1 = 0$ and $\lambda_2 = 10$
  - $f(0, y_2) = \lambda_2 y_2^2 = 10y_2^2$
  - $f(y_1, 0) = \lambda_1 y_1^2 = 0$
- when $b = 10$, $\lambda_1 = -5$ and $\lambda_2 = 15$
  - $f(0, y_2) = \lambda_2 y_2^2 = 15y_2^2$
  - $f(y_1, 0) = \lambda_1 y_1^2 = -5y_1^2$



(d).

explain how the eigenvalues of A affect the curvature of f(x) and, consequently, in which cases they make f(x) a "bowl", a "saddle", or "one-dimensional"

Solution:

The eigenvalues affects the curvature of f(x) and it determines the curvature of f(x) along the direction of the eigenvectors.

For the quadratic form of $f(x) = \mathrm{x}^T A\mathrm{x}$ when $b = 1$, A has two positive eigenvalues of $\lambda_1 = 4, \lambda_2 = 6,$ with corresponding eigenvectors, thus it has positive curvature of f(x) along the direction of those two eigenvectors, which makes it a "bowl' shape.

For $b = 5$, A has one zero eigenvalues and one positive eigenvalues of $\lambda_1 = 0, \lambda_2 = 10$, with corresponding eigenvectors, thus it has positive curvature of f(x) along the direction of the eigenvector associated with $\lambda_2 = 10$, and has no curvature of f(x) along the direction of the eigenvector associated with $\lambda_1 = 0$, which makes it a "one-dimensional" shape.

For $b = 10$, A has one negative and one positive eigenvalues of $\lambda_1 = -5, \lambda_2 = 15$, with corresponding eigenvectors, thus it has negative curvature of f(x) along the direction of the eigenvector associated with $\lambda_1 = -5$, and has positive curvature along the direction of the eigenvector associated with $\lambda_2 = 10$, which makes it a "saddle" shape.

Problem 2.

Consider a random variable X distributed according to a Gaussian mixture

$$P_X(\mathrm{x}) = \sum_{i=1}^{C} \pi_i G(\mathrm{x}, \mu_i, \Sigma_i)$$

where the covariance matrices Σ_i are diagonal.

(a).

consider the case where $\Sigma_i = \sigma_i^2 I$. Determine the principal components (both orientation and length) of X, as a function of the principal components of a dataset whose covariance is the scatter matrix of the means of X

$$S_x = \sum_{i=1}^{C} \pi_i[(\mu_i - \mu_x)(\mu_i - \mu_x)^T],$$

and any other necessary parameters of the mixture.

Solution:

As the statement in last assignment, we prove that,

$$\begin{aligned}
\Sigma_x &= E_X[(\mathrm{x}-\mu_x)(\mathrm{x}-\mu_x)^T] \\
&= \sum_{i=1}^{C} \pi_i [\Sigma_i + (\mu_i - \mu_i)(\mu_i - \mu_x)^T] \\
&= \sum_{i=1}^{C} \pi_i \Sigma_i + \sum_{i=1}^{C} \pi_i [(\mu_i - \mu_i)(\mu_i - \mu_x)^T] \\
&= \sum_{i=1}^{C} \pi_i \Sigma_i + S_x
\end{aligned}$$

since $\Sigma_i = \sigma_i^2 I$,

$$\Sigma_x = \sum_{i=1}^{C} \pi_i \sigma_i^2 I + S_x$$

To determine the principal components of X in terms of S_x, it is same to find the eigenvalue decomposition of S_x.

Suppose the eigenvalue decomposition of S_x is $S_x = PDP^T$, where the columns of P are the eigenvectors of S_x with $PP^T = I$ and D is diagonal matrix with all eigenvalues of S_x on diagonal entries.

Then,

$$\begin{aligned}
\Sigma_x &= \sum_{i=1}^{C} \pi_i \sigma_i^2 I + S_x \\
&= \sum_{i=1}^{C} \pi_i \sigma_i^2 I + PDP^T \\
&= \sum_{i=1}^{C} \pi_i \sigma_i^2 PP^T + PDP^T \\
&= P(\sum_{i=1}^{C} \pi_i \sigma_i^2 I)P^T + PDP^T \\
&= P(\sum_{i=1}^{C} \pi_i \sigma_i^2 I + D)P^T
\end{aligned}$$

Therefore, the principal components of X is all eigenvectors of Σ_x and principal values of X is all eigenvalues of Σ_x.

- for the principal components of X, it is just the eigenvectors of S_x,
- for the principal values of X, it is just the eigenvalues of Σ_x. Denote the eigenvalues of S_x as λ_j, then the eigenvalues of Σ_x is $\sum_{i=1}^{C} \pi_i \sigma_i^2 + \lambda_j$.

(b).

suppose that $S_x = I$, X is two-dimensional, and

$$\Sigma_i = \Sigma = \begin{bmatrix} c & 0 \\ 0 & 5 \end{bmatrix}$$

for all i. If Σ_x is the covariance matrix of X, plot the contours $\mathrm{x}^T S_x^{-1}\mathrm{x} = 1$, and $\mathrm{x}^T \Sigma_x^{-1}\mathrm{x} = 1$ for $c = 1, c = 5, c = 10$. How do the individual component covariances Σ_i affect the covariance of X ?

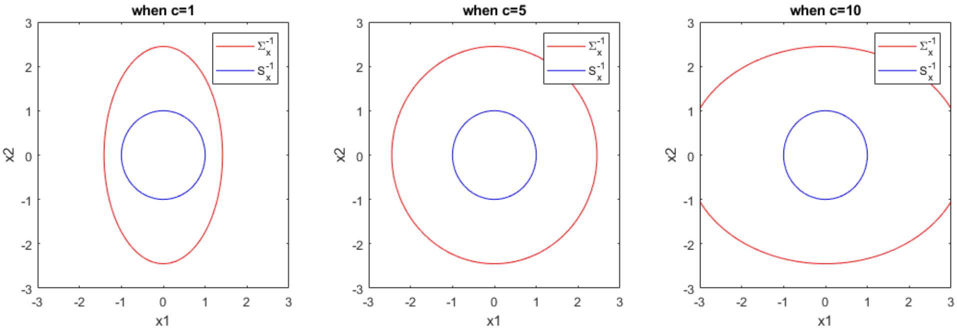
Solution:

- contours of $\mathrm{x}^T S_x^{-1}\mathrm{x} = 1$ where

$$S_x = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- contour of $\mathrm{x}^T \Sigma_x^{-1}\mathrm{x} = 1$ where

$$\Sigma_x = \sum_{i=1}^{C} \pi_i \Sigma_i + S_x = \Sigma \sum_{i=1}^{C} \pi_i + S_x = \Sigma + S_x = \begin{bmatrix} c+1 & 0 \\ 0 & 6 \end{bmatrix}$$

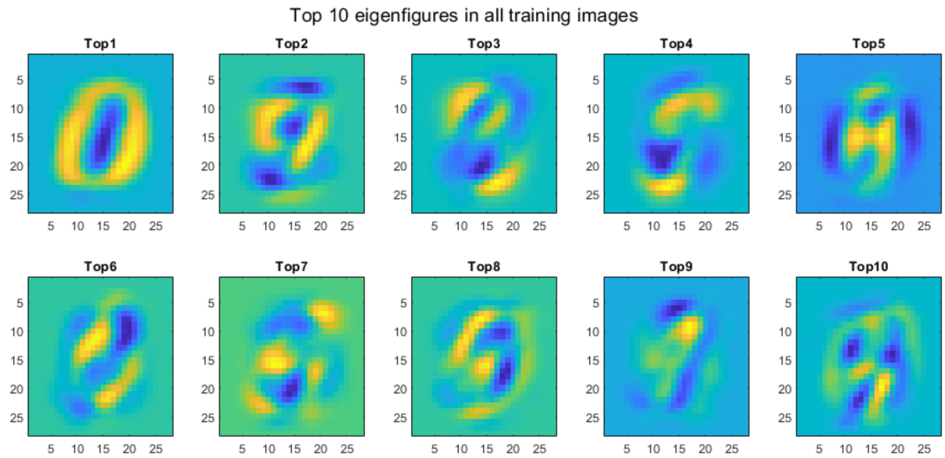


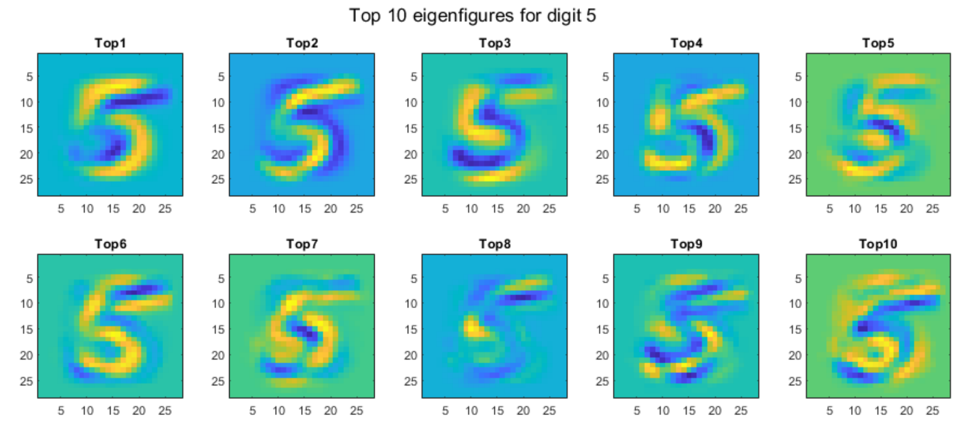
Problem 3.

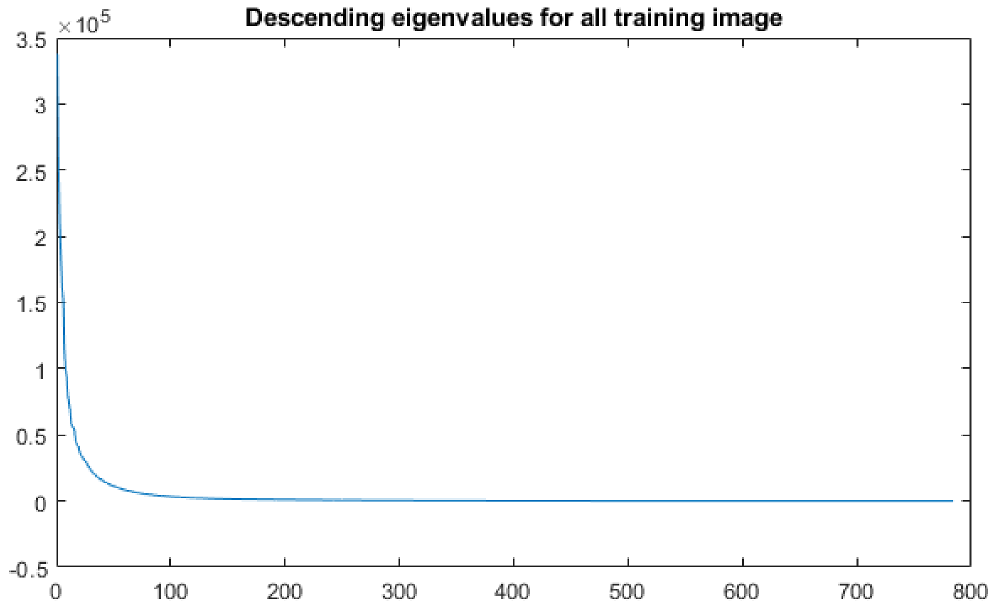
While working with the Gaussian classifier in problem set 3, we were unable to utilize the information given in the covariance matrix and assumed the covariance matrix to be identity. This time we will apply PCA on the training data to reduce the dimensionality of our feature space.

part 1.

Implement the PCA algorithm and find out the principal components for the entire dataset $imageTrain$. Plot the top 10 principal components as 28 × 28 images. Repeat the above, but this time instead of the entire dataset use images from only one class - 'digit5'. Also, for the entire dataset, plot the eigenvalues in a decreasing order.







part 2.

Classification using the PCA subspace on the $imageTest$ dataset.

(a)

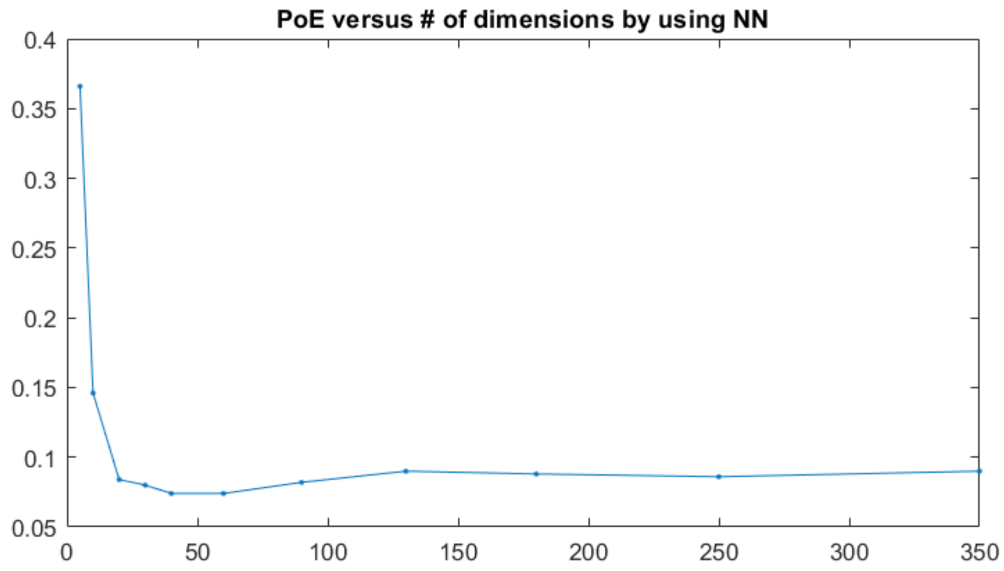
From the plot of the eigenvalues above, what subspace dimension would be best for classification?

Solution:

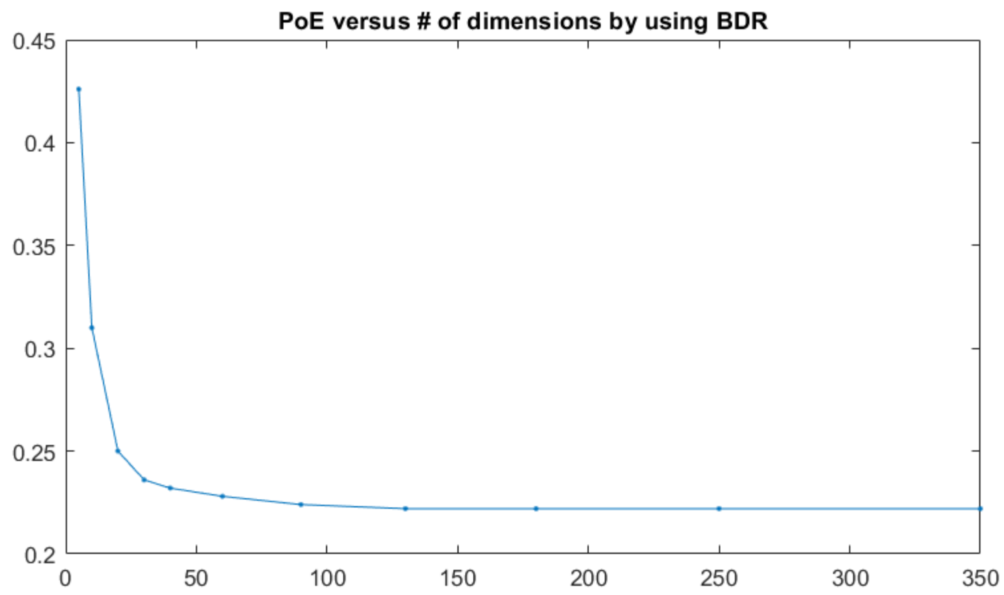
subspace with dimension of 80 would be best for classification as after dimension of 80 the eigenvalues stay almost same, which can be negligible.

(b)

Calculate the total error rate using subspaces of following dimensions: [5, 10, 20, 30, 40, 60, 90, 130, 180, 250, 350]. Plot these error rates.




```
error_list=
0.3660     0.1460     0.0840     0.0800     0.0740     0.0740     0.0820     0.0900
 0.0880    0.0860     0.0900
```




```
error_list=
0.4260     0.3100     0.2500     0.2360     0.2320     0.2280     0.2240     0.2220
 0.2220 0.2220     0.2220
```

(c)

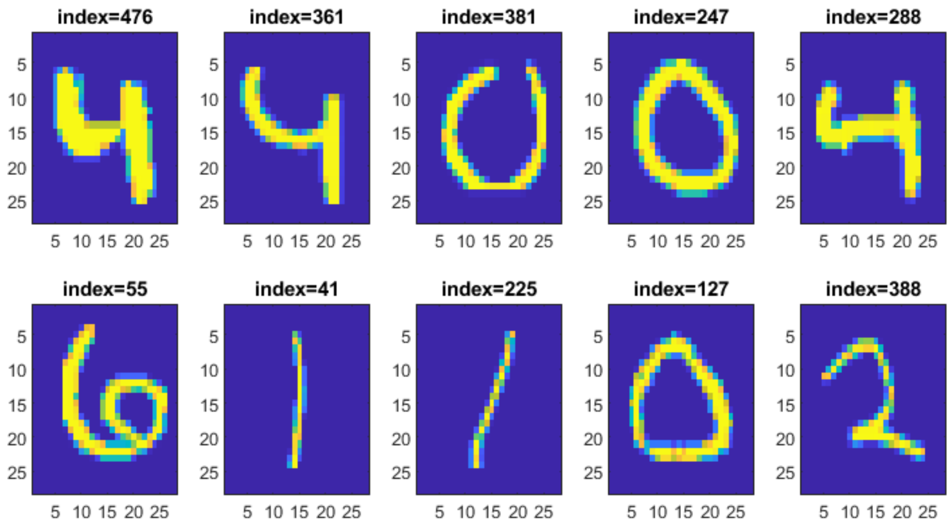
Compare your results with the final error rate obtained in problem set 3.

Solution:

when we conduct PCA with 40 dimensions, the total error rate hits the minimum around 0.074, which performs way better than the results of 0.222 we obtained in problem set 3 where we used full dimensional feature space by BDR.

part 3.

Using the principal components calculated for the class of digit 5 in part 1, find the image from the dataset $imageTest$ that is least like a 5. The least 5-like image, would be the one that has maximum energy in the direction orthogonal to the subspace spanned by the principal components of class 5. Assume that the top 40 eigenvectors are the principal components.



Code:

problem 1.

```
%%% hw6 Problem 1
%%(a)
[x1,x2]=meshgrid(-100:1:100);
A1=[5 1;1 5];
[V1,D1]=eig(A1);
%when b=1;A=[5 1;1 5]
fx=2.*x1.*x2+5*x1.^2+5*x2.^2;
figure
subplot(2,1,1)
meshc(x1,x2,fx);
subplot(2,1,2)
contour(x1,x2,fx);
grid on
hold on
line([0 40*V1(1,1)],[0,40*V1(2,1)])
line([0 40*V1(1,2)],[0,40*V1(2,2)])
sgtitle('contour plot when b=1')

%when b=5;A=[5 5;5 5]
fx=10.*x1.*x2+5*x1.^2+5*x2.^2;
A2=[5 5;5 5];
[V2,D2]=eig(A2);
figure
subplot(2,1,1)
meshc(x1,x2,fx);
subplot(2,1,2)
contour(x1,x2,fx);
```

```
grid on
hold on
line([0 40*V2(1,1)],[0,40*V2(2,1)])
line([0 40*V2(1,2)],[0,40*V2(2,2)])
sgtitle('contour plot when b=5')

%when b=10; A=[5 10;10 5]
fx=20.*x1.*x2+5*x1.^2+5*x2.^2;
A3=[5 10;10 5];
[V3,D3]=eig(A3);
figure
subplot(2,1,1)
meshc(x1,x2,fx);
subplot(2,1,2)
contour(x1,x2,fx);
grid on
hold on
line([0 40*V3(1,1)],[0,40*V3(2,1)])
line([0 40*V3(1,2)],[0,40*V3(2,2)])
sgtitle('contour plot when b=10')

%%(b)
y1=(-10:1:10);
y2=(-10:1:10);

figure
subplot(1,3,1)
f1_y=6*y2.^2;
f2_y=4*y1.^2;
plot(y2,f1_y)
grid on
hold on
plot(y1,f2_y)
title('f(y) when b=1')
legend('y_1=0','y_2=0')

subplot(1,3,2)
f1_y=10*y2.^2;
f2_y=0*y1.^2;
plot(y2,f1_y)
grid on
hold on
plot(y1,f2_y)
title('f(y) when b=5')
legend('y_1=0','y_2=0')

subplot(1,3,3)
f1_y=15*y2.^2;
f2_y=-5*y1.^2;
plot(y2,f1_y)
grid on
hold on
plot(y1,f2_y)
title('f(y) when b=10')
legend('y_1=0','y_2=0')
```

problem 2.

```
%%problem 2
figure
c=[1,5,10];
%when c=1;
for i=1:3
    syms x1 x2
    subplot(1,3,i)

    cova=[c(i) 0;0 5];
    I=eye(2);
    y1=[x1;x2]'*inv(cova+I)*[x1;x2];
    y2=[x1;x2]'*inv(I)*[x1;x2];
    [x1,x2]=meshgrid(-3:0.1:3);
    y1=eval(y1);
    y2=eval(y2);

    contour(x1,x2,y1,[1,1],'linecolor','r')
    hold on
    contour(x1,x2,y2,[1,1],'linecolor','b')
    legend('\Sigma_x^{-1}','S_x^{-1}');
    title(['when c=',num2str(c(i))])
    xlabel('x1')
    ylabel('x2')
end
```

problem 3.

```
function [V,D] = eigenfigure(input)

n=size(input,2);
%compute the mean
avg=mean(input,2);
%comoute the covariance
avg_reshape=repmat(avg,1,n);
covariance=(input-avg_reshape)*(input-avg_reshape)';
covariance=covariance/n;
%compute the eigenvalues and eigenvectors of covariance matrix
[V,D]=eig(covariance);

end
```

```
clear
clc

%%problem 3
data=load('HW6_Data\data.mat');
label=load('HW6_Data\label.mat');
imageTest=data.imageTest;
imageTrain=data.imageTrain;
labelTest=label.labelTest;
```

```
labelTrain=label.labelTrain;

%%%%%part 1
%reshape training images to 784*5000 dimensions
imageTrain_reshape=reshape(imageTrain,[784,5000]);
imageTest_reshape=reshape(imageTest,[784,500]);
%%%%%compute the PC components for the training images
[V,D]=eigenfigure(imageTrain_reshape);
eigenfigureset = V(:,(end:-1:end-9));
[d,ind] = sort(diag(D),'descend');
%%%%%compute the PC components for digit 5
digit_ind=find(labelTrain==5);
digit_img=imageTrain_reshape(:,digit_ind);
[V,D]=eigenfigure(digit_img);
eigenfigureset = V(:,(end:-1:end-39));

figure
sgtitle('Top 10 eigenfigures for all training data')
for i =1:10
    subplot(2,5,i)
    eigenfigure=reshape(eigenfigureset(:,i),[28,28]);
    imagesc(eigenfigure)
    title(['Top',num2str(i)])
end

figure
plot(1:784,d)
title('Descending eigenvalues for all training image')
```

```
%%problem 3 part 2
clear
clc

data=load('HW6_Data\data.mat');
label=load('HW6_Data\label.mat');
imageTest=data.imageTest;
imageTrain=data.imageTrain;
labelTest=label.labelTest;
labelTrain=label.labelTrain;

dim=[5, 10, 20, 30, 40, 60,90, 130, 180, 250, 350];
imageTrain_reshape=reshape(imageTrain,[784,5000]);
imageTest_reshape=reshape(imageTest,[784,500]);
[V,D]=eigenfigure(imageTrain_reshape);
error_list=zeros(1,length(dim));

for iteration =1:1
    s=dim(iteration)-1;
    %obtain the first K dimensions
    eigenfigureset = V(:,(end:-1:end-s));
    %compute the average eigenfigure
    avg_eigenfigure=mean(imageTrain_reshape,2);
    %project testing data onto K dimension subspace
    Y=eigenfigureset'*(imageTest_reshape-repmat(avg_eigenfigure,1,500));
    %project training data onto K dimension subspace
```

```
    Z=eigenfigureset'*(imageTrain_reshape-repmat(avg_eigenfigure,1,5000));

    error=0;
    predicted_label=zeros(500,1);
    for i =1:500
        reduced_imageTest=Y(:,i);
        reduced_imageTest=repmat(reduced_imageTest,1,5000);
        dif=Z-reduced_imageTest;
        dif=sqrt(sum(dif.^2));
        [~,index]=min(dif);
        label=labelTrain(index);
        predicted_label(i)=label;
        if predicted_label(i) ~= labelTest(i)
            error=error+1;
        end
    end
    error_list(iteration)=error;
end

figure
plot(dim,error_list/500,'.-');
title('PoE versus # of dimensions by using NN');
```

```
%%%%%part 3
%compute the average eigenfigure
avg_eigenfigure=mean(digit_img,2);
%project testing data onto K dimension subspace
Y=eigenfigureset'*(imageTest_reshape-repmat(avg_eigenfigure,1,500));

%obtain the average reduced dimensional feature of digits 5
%avg_digits_5=mean(eigenfigureset,2);
avg_digits_5=eigenfigureset'*(avg_eigenfigure);
avg_digits_5=repmat(avg_digits_5,1,500);
dif=Y-avg_digits_5;
dif=sqrt((sum(dif.^2)));
[B,I]=maxk(dif,10);

figure
for i=1:10
    subplot(2,5,i)
    image=imageTest(:,:,I(i));
    imagesc(image)
    title(['index=',num2str(I(i))])
end
```