Midterm Report
Chenchen Zhou

State of Problem
Assume that we have some facial images of known people and our goal is to recognize people from those new facial images(of the same set of people) by comparing to those images we have recognized and labeled.

The data of images is stored in a matrix of size $(s_1 \times s_2) \times (n_1 \times n_2)$. Each image (observation) is stored in a column of size $(s_1 \times s_2)$. We have n_2 persons, each person have n_1 images. And they are arranged as follows: first n_2 columns are images of person 1, next n_2 columns are images of person 2, and so on,with a total of $n_1 \times n_2$ columns.

There are two sets of images available
(i) training set: these images are already recognized and labeled, and (ii) test images: these are new images that are to be recognized and labeled. The goal is to use the similarities between the test and the training images to label the test images.

Methodology
To analysis images which are very high dimensional, we have two essential procedures.

- Feature Extraction,reduce dimensionality of images by using Principal Component Analysis(PCA)
- Classification, recognize the new image by using the Nearest Neighbor algorithm.

In Feature extraction, first by PCA, computes the principal components of the training images and compute the projection to their principal k-dimensional subspace. That is, represent each training image which is a vector of size $(s_1 \times s_2)$by a small vector of size k.
In Classification, we take a test image(select a column from test images's matrix), use the same projection in Feature extraction step to represent each test image by a k vector. Find the nearest image in the training set by comparing their k vectors.

Matlab programs
I have two program files. midterm.m and midtermtest.m . The code is attached in the following.

First one is to compute an average performance for the two projections: PCA and simple projection(for comparison) given different k(dimension of subspace which project images to).
To compute an average performance, take an image randomly from the test set for 100 times and recognize those images by applying our approach. Compute the percentage of successful recognition. Call this number F(k). Generate this plot for each of the two projections: PCA and the simple projection of F(k) versus k for k = 1, 2, . . . , 40.

Second file is to demonstrate by a few examples the test image and the closest image in the training set. That is , by selecting a specific k, this program would select randomly a image from test set and find the nearest image in training set by PCA projection and simple projection. The program would show five images, i.e. the randomly selected test image, projection onto the principal subspace of this test image for PCA and simple projection, and nearest image in the training set which is found by PCA projection and the nearest image in the training set which is found by simple projection. What's more, it would show the right label of this test image and the computed label by PCA and simple projection, so in this way, we can tell which projection give the right classification.

Results

Tabulate the average performance of correct recognition versus the number k.

k	1	2	3	4	5	6	7	8	9	10
PCA(%)	9	29	51	59	63	64	81	82	85	84
Simple projection(%)	19	21	26	22	26	33	49	35	48	47

Table 1: percentage of successful recognition:F(k)

k	11	12	13	14	15	16	17	18	19	20
PCA(%)	83	83	92	84	86	81	79	85	85	85
Simple projection(%)	37	49	46	48	50	50	45	45	48	42

Table 2: percentage of successful recognition:F(k)

k	21	22	23	24	25	26	27	28	29	30
PCA(%)	84	79	88	91	91	84	88	91	83	89
Simple projection(%)	41	44	50	50	41	47	55	55	53	56

Table 3: percentage of successful recognition:F(k)

k	31	32	33	34	35	36	37	38	39	40
PCA(%)	94	93	89	87	91	83	86	88	86	90
Simple projection(%)	59	55	61	49	51	59	59	63	57	59

Table 4: percentage of successful recognition:F(k)

a few example and graphs attache following the report

Summary

(1)The effectiveness of both projections.

(1)From the table and graph which I listed, the PCA projection is significantly better recognizing image than Simple Projection.

(2)The PCA projection keeps a high percentage of successful recognition, to be more clearly, it keeps a percentage successful recognition around the range of $[80\%, 90\%]$ when $k \geqslant 7$. That is it vibrates about in the range of $[80\%, 90\%]$

While the simple projection has much lower percentage of successful recognition comparing to PCA. It vibrates in the range of $(35\%, 65\%)$ and the percentage of successful recognition grows gradually (not very significantly) as k increase. It starts to reach about 60% when k grows to about 30.

(3)The PCA only has a lower percentage of successful recognition only when k=1, after that, the percentage of successful recognition of PCA increases rapidly and increase up to 80% from 9% as k increases from 1 to 7.

While the percentage of successful recognition of simple projection also increases from about 20% to about 50% as k increases from 1 to 7. After that, it grows very slowly and reach about 60% when k increases to about 31.

(2)Comment of the variation of F(k) versus k in each case.

- PCA :

  - It has a very low percentage of successful recognition when k is small, the percentage of successful recognition is below 30% when $k \leqslant 2$, and below 70% when $k \leqslant 6$.
  - When $k \leqslant 7$, as k increases, the percentage of successful recognition increases rapidly. It increases from about 10% to 80% when k increases from 1 to 7.
  - When $k \geqslant 7$, as k increases, the percentage of successful recognition doesn't change much. It just keeps at a stable level which means vibrates about in the range of $(80\%, 90\%)$.

- Simple projection:
  - It has a very low percentage of successful recognition when $k \leqslant 2$, the percentage of successful recognition is below 20% when k=1.
  - The percentage of successful recognition grows gradually as k increases. When k reaches 7, it reaches about 50%, but it increases very slowly.
  - The percentage of successful recognition is much lower than PCA. It varies in the range of $(35\%, 65\%)$ and only reaches about 60% when k is big enough, say k is at least 30.

Code

```
%%% Perform the PCA analysis on training data %%%
clear;
load mid_train;
load mid_test;
n1=40;  n2=5;

[n, m] = size(Ytrain);

% 1. Compute sample covariance
C = cov(Ytrain');

% 2. SVD on C
[U, S, V] = svd(C);
for k=1:40

% 3. select the first k columns of U,U is a n by n matrix
U1=U(:,1:k);  % PCA
U2=eye(n,k);    % simple projection
```

```
% feature extraction
%Y1 = U1'*x;
U1=U1';U2=U2';
for l=1:k
for j=1:m
 Y1(l,j)=U1(l,:)*Ytrain(:,j); %PCA
 Y2(l,j)=U2(l,:)*Ytrain(:,j); % simple projection
end
end
num1=0; num2=0;



%classification
for i=1:100
%select a column I randomly from Ytest
colum=ceil(m*rand(1));
I=Ytest(:,colum);
%Compute the projection of I
I1=U1*I; % PCA
I2=U2*I; % simple

%Compute the distant between I and each column of Y
for j = 1:m
     d1(j) = norm(I1-Y1(:,j)); % PCA
     d2(j) = norm(I2-Y2(:,j)); % simple
end

%Find Nearest Neighbors
%PCA
[dis1,dis_ind1]=sort(d1);
index1=dis_ind1(1);
% simple
[dis2,dis_ind2]=sort(d2);
index2=dis_ind2(1);

% Now verify wether the classification is right.
lab1=ceil(index1/n2);% PCA
lab2=ceil(index2/n2);% simple
```

```
labeltr=ceil(colum/n2);% true label
if lab1==labeltr
    num1(i)=1;
else
    num1(i)=0;
end
if  lab2==labeltr
    num2(i)=1;
else
    num2(i)=0;
end
end

sucnum1(k)=sum(num1);
sucnum2(k)=sum(num2);
F1(k)=sum(num1)/100;
F2(k)=sum(num2)/100;
end

figure(1);
title('percentage of successful recognition F(k) vs k')
axis([0 41 0 1.1]);
axis equal;

plot(F1,'+b');
hold on;
plot(F1,'b');
plot(F2,'xr');
plot(F2,'r');
xlabel('dimension of projection space k');
ylabel('percentage of successful recognition');

%%% Perform the PCA analysis on training data %%%
clear;
load mid_train;
load mid_test;
n1=40;  n2=5;

[n, m] = size(Ytrain);
```

```
% 1. Compute sample covariance
C = cov(Ytrain');

% 2. SVD on C
[U, S, V] = svd(C);
k=35

% 3. select the first k columns of U,U is a n by n matrix
U1=U(:,1:k);  % PCA
U2=eye(n,k);     % simple projection

% feature extraction
%Y1 = U1'*x;
U1=U1';U2=U2';
for l=1:k
for j=1:m
 Y1(l,j)=U1(l,:)*Ytrain(:,j); %PCA
 Y2(l,j)=U2(l,:)*Ytrain(:,j); % simple projection
end
end

%classification
%select a column I randomly from Ytest
colum=ceil(m*rand(1));
I=Ytest(:,colum);
%Compute the projection of I
I1=U1*I; % PCA
I2=U2*I; % simple

%Compute the distant between I and each column of Y
for j = 1:m
      d1(j) = norm(I1-Y1(:,j)); % PCA
      d2(j) = norm(I2-Y2(:,j)); % simple
end

%Find Nearest Neighbors
%PCA
[dis1,dis_ind1]=sort(d1);
index1=dis_ind1(1);
% simple
```

```
[dis2,dis_ind2]=sort(d2);
index2=dis_ind2(1);

% Now veryfy wether the classification is right.
lab1=ceil(index1/n2)% PCA
lab2=ceil(index2/n2)% simple
labeltr=ceil(colum/n2)% true label
y1=zeros(n,1);
y2=zeros(n,1);

for i=1:k
    y1=y1+I1(i)*U(:,i);
    y2=y2+I2(i)*U(:,i);
end
subplot(2,3,1);
gf0=reshape(Ytest(:,labeltr),28,23);
imagesc(gf0);
colormap(gray)       % a image randomly took from test data.
axis equal;
xlabel('the selected test image')

subplot(2,3,2);
g1=reshape(y1,28,23);   % projection image for PCA
imagesc(g1);
colormap(gray)
axis equal;
xlabel('projection by PCA')
subplot(2,3,3);
g2=reshape(y2,28,23);        % projection for simple projection
imagesc(g2);
colormap(gray)
axis equal;
xlabel('projection by simple projection')

subplot(2,3,4);
gf1=reshape(Ytrain(:,lab1),28,23);% closest image in training date
                                  %according to PCA
imagesc(gf1);
colormap(gray)
axis equal;
```

```
xlabel('closest image in training date by PCA')

subplot(2,3,6);
gf2=reshape(Ytrain(:,lab2),28,23);% closest image in training data
                                   %according to simple projection

imagesc(gf2);
colormap(gray)
axis equal;
xlabel('closest image in training date by simple projection')
```