# ECE 637 Lab 5 - Eigen-decomposition of Images

Yifan Fei

Electrical Engineering
Purdue University

Instructor: Prof. Charles A. Bouman

Spring 2018

# Section 2: Multivariate Gaussian Distributions and Whitening

## 2.1:Exercise: Generating Gaussian random vectors

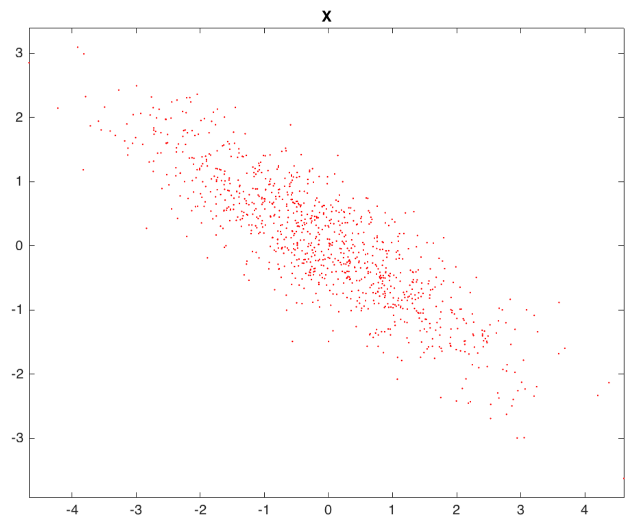


Figure 1: n = 1000, 2D scatter plot of X

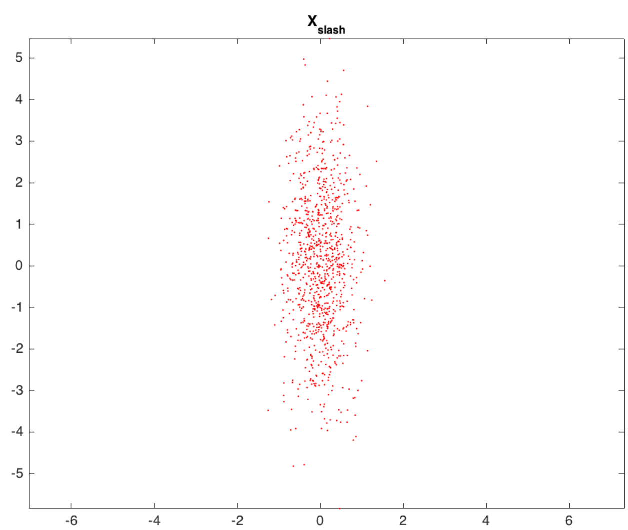


Figure 2: n = 1000, 2D scatter plot of X slash

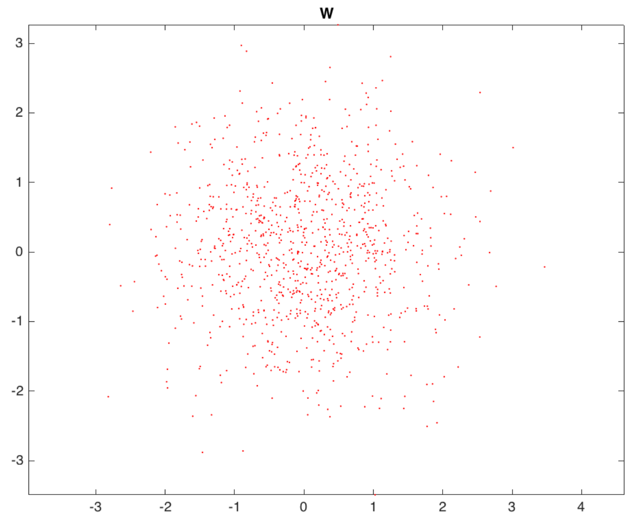


Figure 3: n = 1000, 2D scatter plot of W

## 2.2: Exercise: Covariance Estimation and Whitening

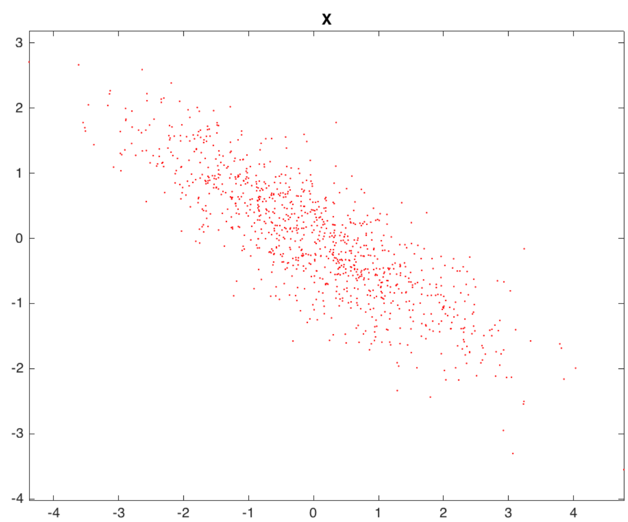


Figure 4: n = 1000, 2D scatter plot of estimated X

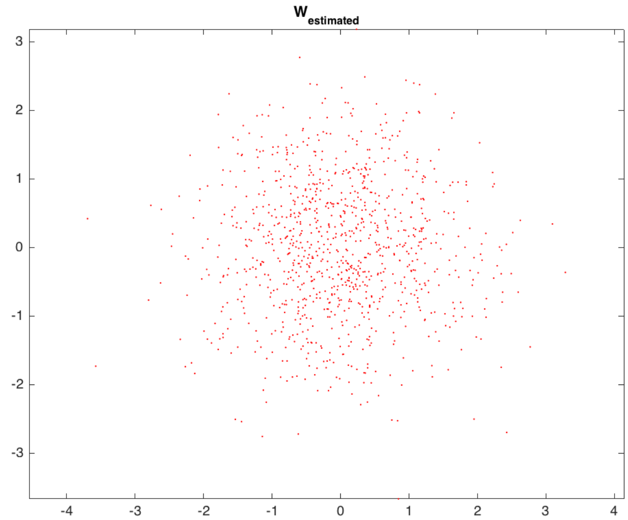


Figure 5: n = 1000, 2D scatter plot of estimated W

The theoretical value of the covariance matrix, $R_X$ is: $\begin{bmatrix} 2 & \text{-1.2} \\ \text{-1.2} & 1 \end{bmatrix}$

The numerical listing of my covariance estimate $R_{Xest}$ is: $\begin{bmatrix} 1.9059 & -1.1004 \\ -1.1004 & 0.9205 \end{bmatrix}$ Which is close to $R_X$.

the covariance estimate of W is: $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ Which is quite close to the ture value.

## Section4: Eigenimages, PCA, and Data Reduction

### 4.1.1: Tthe figure with the first 12 eigenimages.

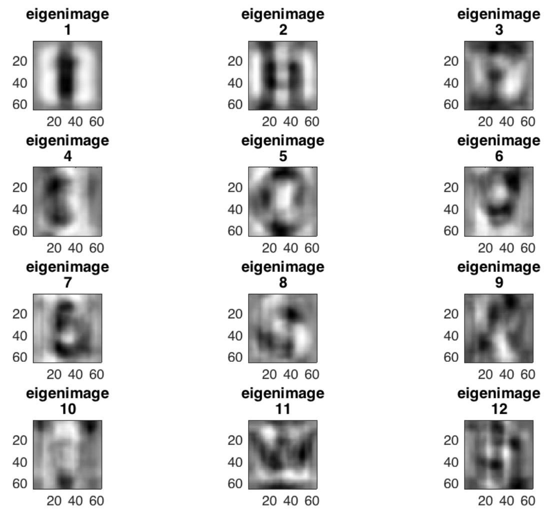


Figure 6: First 12 eigenimages

### 4.1.2: the plots of projection coefficients vs. eigenvector number

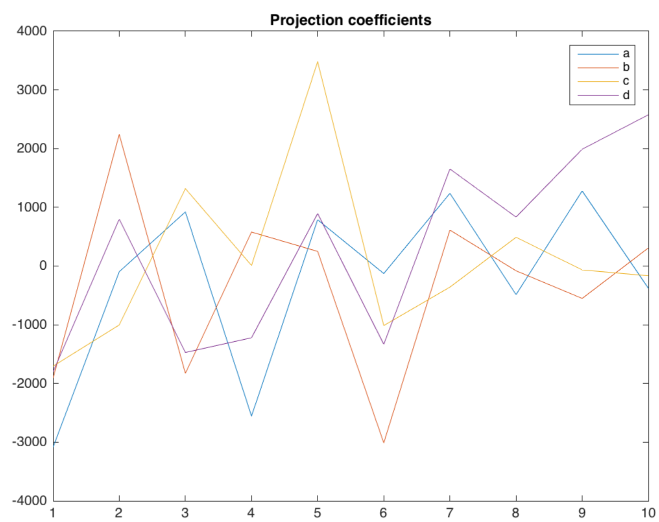


Figure 7: projection coefficients vs. eigenvector number

### 4.1.3: original image, and the 6 resynthesized versions

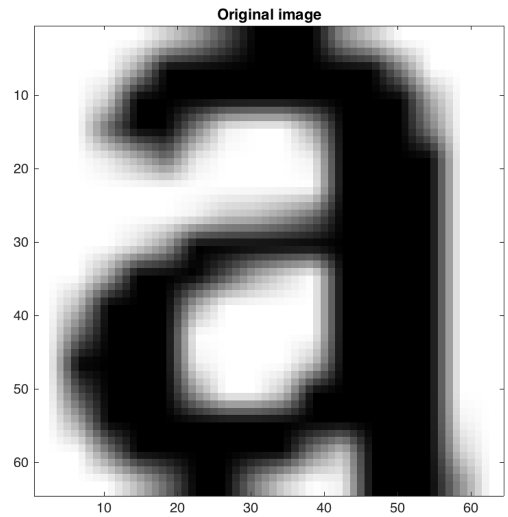


Figure 8: original image 'a'

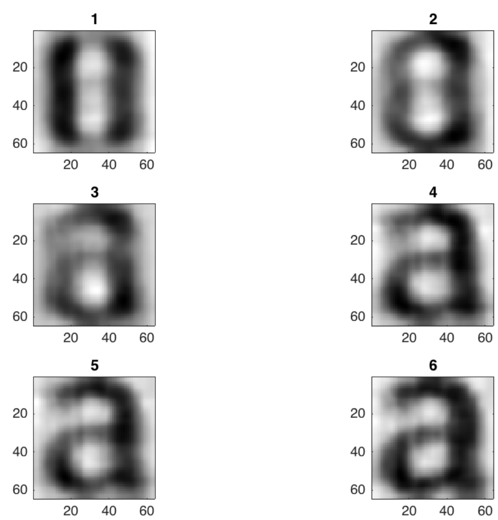

Figure 9: 6 re-synthesized versions 'a'

# 5:Image Classification

## 5.1: Exercise: Classification and PCA

Here is a 2-column table showing for each mis-classified input image: (1) the input character, and (2) the output from the classifier

```
k_table =

     1     1
     2     2
     3     3
     4     1
     5     5
     6     6
     7     7
     8     8
     9     9
    10    25
    11    11
    12     9
    13    13
    14    22
    15    15
    16     5
    17     1
    18    18
    19    19
    20    20
    21     1
    22    22
    23    23
    24    24
    25    22
    26    26

--------
It is 4 But wrong prediction is 1
--------
It is 10 But wrong prediction is 25
--------
It is 12 But wrong prediction is 9
--------
It is 14 But wrong prediction is 22
--------
It is 16 But wrong prediction is 5
--------
It is 17 But wrong prediction is 1
--------
It is 21 But wrong prediction is 1
--------
It is 25 But wrong prediction is 22
```

Figure 10: mis-classified

Table 1: **mis-classified using original R**

| **Input letter** | **mis-classified output** |
|---|---|
| d | a |
| j | y |
| l | i |
| n | v |
| p | e |
| q | a |
| u | a |
| y | v |

## 5.2: Improvement by using B

$B_k = \Lambda_k$

```
k_table =

     1     1
     2     2
     3     3
     4     4
     5     5
     6     6
     7     7
     8     8
     9    12
    10    10
    11    11
    12    12
    13    13
    14    14
    15    15
    16    16
    17    17
    18    18
    19    19
    20    20
    21    21
    22    22
    23    23
    24    24
    25    22
    26    26

--------
It is 9 But wrong prediction is 12
--------
It is 25 But wrong prediction is 22
```

Figure 11: mis-classified by using $B_k = \Lambda_k$

Table 2: **mis-classified using $B_k = \Lambda_k$**

| Input letter | mis-classified output |
|---|---|
| i | l |
| y | v |

$B_k = R_{wc}$

```
k_table =

     1     1
     2     2
     3     3
     4     4
     5     5
     6     6
     7    17
     8     8
     9     9
    10    10
    11    11
    12    12
    13    13
    14    14
    15    15
    16    16
    17    17
    18    18
    19    19
    20    20
    21    21
    22    22
    23    23
    24    24
    25    22
    26    26

--------
It is 7 But wrong prediction is 17
--------
It is 25 But wrong prediction is 22
```

Figure 12: mis-classified by using $B_k = R_{wc}$

Table 3: **mis-classified using $B_k = R_{wc}$**

| Input letter | mis-classified output |
|---|---|
| g | q |
| y | v |

$B_k = \Lambda$

```
k_table =

     1     1
     2     2
     3     3
     4     4
     5     5
     6    20
     7     7
     8     8
     9     9
    10    10
    11    11
    12    12
    13    13
    14    14
    15    15
    16    16
    17    17
    18    18
    19    19
    20    20
    21    21
    22    22
    23    23
    24    24
    25    22
    26    26

--------
It is 6 But wrong prediction is 20
--------
It is 25 But wrong prediction is 22
```

Figure 13: mis-classified by using $B_k = \Lambda$

Table 4: **mis-classified using** $B_k = \Lambda$

| Input letter | mis-classified output |
|---|---|
| f | t |
| y | v |

$B_k = I$

```
k_table =

     1     1
     2     2
     3     3
     4     4
     5     5
     6    20
     7    17
     8     8
     9     9
    10    10
    11    11
    12    12
    13    13
    14    14
    15    15
    16    16
    17    17
    18    18
    19    19
    20    20
    21    21
    22    22
    23    23
    24    24
    25    22
    26    26

--------
It is 6 But wrong prediction is 20
--------
It is 7 But wrong prediction is 17
--------
It is 25 But wrong prediction is 22
```

Figure 14: mis-classified by using $B_k = I$

Table 5: **mis-classified using** $B_k = \Lambda$

| **Input letter** | **mis-classified output** |
|---|---|
| f | t |
| g | q |
| y | v |

## 5.3: Which works well?

1. From above classifiers in this experiment, $B_k = \Lambda_k, B_k = \Lambda, B_k = R_{wc}$ have two wrong predictions, while $B_k = I$ has three wrong prediction.

2. In constraining the covariance, what is the trade off between the accuracy of the data model and the accuracy of the estimates?

The trade off is about the complexity of the model and the accuracy of the estimates. As we can see, when the model is very accurate when $B_k = R_k$, the result has 8 wrong predicitions. However, too simple model like $B_k = I$, the results has 3 wrong predicitons. Only we have appropriate models with appropriate complexity can we achieve relatively better results. To acquiesce good estimations, we need to choose a model that is not too accrate but also not too simple.

## Attachments: code

### Q2

```
clear all;
clc;
n = 1000;
W = randn(2,n);
Rx = [2 -1.2;-1.2 1];
[E,Gama] = eig(Rx);
X_ = sqrt(Gama)*W;
X = E*X_;

figure(1);
plot(W(1,:),W(2,:),'.r','LineWidth',3)
title('W')
axis('equal')

figure(2);
plot(X_(1,:),X_(2,:),'.r','LineWidth',3)
title('X_{slash}')
axis('equal')

figure(3);
plot(X(1,:),X(2,:),'.r','LineWidth',3)
title('X')
axis('equal')

u = mean(X,2);
u = repmat(u,1,n);
R_est = (X-u)*(X-u)'/(n-1)

[E_est,Gama_est] = eig(R_est);
W_est = sqrt(Gama_est)^(-1)*E_est'*X;

u_W = mean(W_est,2);
u_W = repmat(u_W,1,n);
Rw_est = (W_est-u_W)*(W_est-u_W)'/(n-1)


figure(4);
plot(W_est(1,:),W_est(2,:),'.r','LineWidth',3)
title('W_{estimated}')
axis('equal')

% figure(5);
% plot(est_X_(1,:),est_X_(2,:),'.r','LineWidth',3)
```

```
% title('X_{slash}')
% axis('equal')

figure(6);
plot(X(1,:),X(2,:),'.r','LineWidth',3)
title('X')
axis('equal')
```

## Q4

```
%
% read_data.m
%
% ECE637
% Prof. Charles A. Bouman
% Image Processing Laboratory: Eigenimages and Principal Component Analysis
%
% Description:
%
% This is a Matlab script that reads in a set of training images into
% the Matlab workspace. The images are sets of English letters written
% in various fonts. Each image is reshaped and placed into a column
% of a data matrix, "X".
%
clear all;
clc;

% The following are strings used to assemble the data file names
datadir='.';    % directory where the data files reside
dataset={'arial','bookman_old_style','century','comic_sans_ms','courier_new',...
  'fixed_sys','georgia','microsoft_sans_serif','palatino_linotype',...
  'shruti','tahoma','times_new_roman'};
datachar='abcdefghijklmnopqrstuvwxyz';

Rows=64;    % all images are 64x64
Cols=64;
n=length(dataset)*length(datachar);  % total number of images
p=Rows*Cols;   % number of pixels

X=zeros(p,n);  % images arranged in columns of X
k=1;
for dset=dataset
for ch=datachar
  fname=sprintf('%s/%s/%s.tif',datadir,char(dset),ch);
  img=imread(fname);
  X(:,k)=reshape(img,1,Rows*Cols);
  k=k+1;
end
end
%
% display samples of the training data, all 'a'
for k=1:length(dataset)
  img=reshape(X(:,26*(k-1)+1),64,64);
  figure(20); subplot(3,4,k); imshow(img);
```

```
    axis('image'); colormap(gray(256));
    title(dataset{k},'Interpreter','none');
end

u = mean(X,2);
u = repmat(u,1,n);
Z = X - u;
[U,S,V]=svd(Z,0);
% Rx = (X-u)*(X-u)'/(n-1);
% [E,Gama] = eig(Rx);
% [U,S,V]= svd(1/sqrt(n)*X,0);
% R_est = U*S^2*U';

% [U,S,V]= svd(X,0);
U12 = U(:,1:12);
% S12 = S(1:12,:);
% V12 = V(1:12,:);
% X12 = U12*S12*V12';

% X1111 = U12*U12'*X;

% display samples of the training data
for k=1:12
    img=reshape(U12(:,k),64,64);
    figure(30); subplot(4,3,k); imagesc(img);
    axis('image'); colormap(gray(256));
    title(['eigenimage' {k}],'Interpreter','none');
end

Y = U'*(X-u);
Y4 = Y(1:10,1:4);
figure(40);
t = 1:10;
plot(t,Y4(:,1),t,Y4(:,2),t,Y4(:,3),t,Y4(:,4))
legend('a','b','c','d');
xlim([1,10]);
title('Projection coefficients');

m = [1 5 10 15 20 30];
k = 1;
for i = m
      Um = U(:,1:i);
      Ym = Um'*(X-u);
      X_est = Um*Ym;
      X1_est = X_est(:,1);
      X1_est = X1_est + u(:,1);
      img=reshape(X1_est,64,64);
      figure(12); subplot(3,2,k); imagesc(img);
      axis('image'); colormap(gray(256));
      title({k},'Interpreter','none');
      k = k+1;
end

figure(10);
```

```
img=reshape(X(:,1),64,64);
imagesc(img);
axis('image'); colormap(gray(256));
title('Original image');
```

## Q5

```
%
% read_data.m
%
% ECE637
% Prof. Charles A. Bouman
% Image Processing Laboratory: Eigenimages and Principal Component Analysis
%
% Description:
%
% This is a Matlab script that reads in a set of training images into
% the Matlab workspace.  The images are sets of English letters written
% in various fonts.  Each image is reshaped and placed into a column
% of a data matrix, "X".
%
clear all;
clc;

% The following are strings used to assemble the data file names
datadir='.';    % directory where the data files reside
dataset={'arial','bookman_old_style','century','comic_sans_ms','courier_new',...
  'fixed_sys','georgia','microsoft_sans_serif','palatino_linotype',...
  'shruti','tahoma','times_new_roman'};
datachar='abcdefghijklmnopqrstuvwxyz';

Rows=64;    % all images are 64x64
Cols=64;
n=length(dataset)*length(datachar);  % total number of images
p=Rows*Cols;   % number of pixels

X=zeros(p,n);  % images arranged in columns of X
k=1;
for dset=dataset
for ch=datachar
  fname=sprintf('%s/%s/%s.tif',datadir,char(dset),ch);
  img=imread(fname);
  X(:,k)=reshape(img,1,Rows*Cols);
  k=k+1;
end
end

%%
u = mean(X,2);
u = repmat(u,1,n);
Z = (X-u)/sqrt(n-1);
[U,S,V]= svd(Z,0);
A = U(:,1:10);
Y = A'*(X-u);
```

```
%%
empty_cell=cell(26,2);
params=cell2struct(empty_cell,{'M','R'},2);
diag_ = cell2struct(empty_cell,{'M','R'},2);

for k = 1:26
    params(k).M = 0;
    params(k).R = 0;
    for j = 1:length(dataset)
    params(k).M = params(k).M + Y(:,26*(j-1)+k);
    end
    params(k).M = params(k).M/12;

    for j = 1:length(dataset)
    params(k).R = params(k).R + (Y(:,26*(j-1)+k)-params(k).M)*(Y(:,26*(j-1)+k)-params(k).M
    end
    params(k).R = params(k).R/11;
    diag_(k).R = diag(diag(params(k).R));
end


%% Read test data
% go to folders up the hierarchy:
upUpFolder = fileparts(pwd);
% go into another folder
folder = fullfile(upUpFolder, './test_data/veranda');

j = 1;
for ch=datachar
  fname=fullfile(folder,'/',[ch,'.tif']);
  img=imread(fname);
  X_test(:,j)=reshape(img,1,Rows*Cols);
  j=j+1;
end
X_test = double(X_test);
u_test = repmat(mean(X,2),1,26);
y_test = A'*((X_test-u_test));

%%
R_wc = zeros(size(params(1).R));
for k = 1:26
    R_wc = R_wc + params(k).R;
end
R_wc = R_wc/26;
B = R_wc;
% %%
B = diag(diag(R_wc));
%%
% B = eye(10);

%%
k_star = [];
k_right = 1:26;
k_star_pred = [];
```

```
for i = 1:26
    y_test_column = y_test(:,i);
    for k = 1:26
%      k_star(k) = (y_test_column - params(k).M)'* params(k).R^(-1)* (y_test_column - params
%      k_star(k) = (y_test_column - params(k).M)'* diag_(k).R^(-1)* (y_test_column - params(
    k_star(k) = (y_test_column - params(k).M)'* B^(-1)* (y_test_column - params(k).M)+ log(
    end
    k_star_pred(i) = find(k_star == min(k_star(:)));
end

k_table = [k_right',k_star_pred']
for i = 1:26
    if k_right(i) ~= k_star_pred(i)
        disp('---------')
        disp(['It is ',num2str(k_right(i)),' But wrong prediction is ',num2str(k_star_pred(
    end
end
```