
ECE 637 Laboratory Exercise 5

Eigen-decomposition of Images

Tong Shen

February 24, 2017

1 MULTIVARIATE GAUSSIAN DISTRIBUTIONS AND WHITENING

1.1 GENERATING GAUSSIAN RANDOM VECTORS

In this section, our goal will be to use Matlab to generate independent Gaussian random vectors, which have certain covariance. The formula is:

$$X = E\Lambda^{1/2}W \quad (1.1)$$

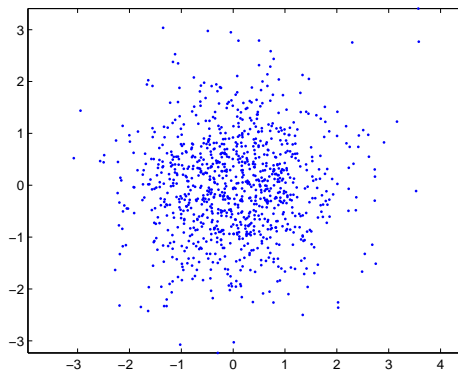


Figure 1.1: Plot of X , uncorrelated Gaussian Random Vector

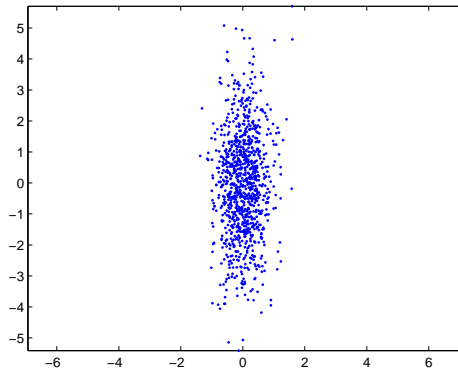


Figure 1.2: Plot of \tilde{X} , Scaled Random Vector

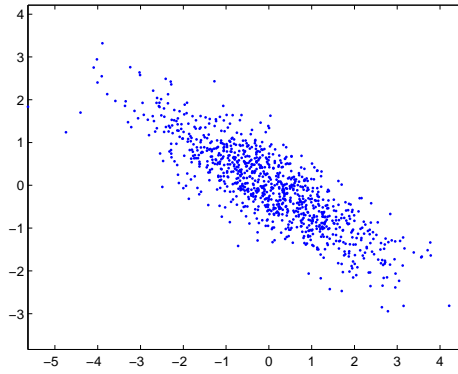


Figure 1.3: Plot of W , Random Vectors with a Covariance R

1.2 COVARIANCE ESTIMATION AND WHITENING

In this section, we will use some method to estimate the covariance and get the decorrelation done.

The given theoretical value of the covariance matrix, R_x is as following:

$$R_x = \begin{bmatrix} 2 & -1.2 \\ -1.2 & 1 \end{bmatrix} \quad (1.2)$$

While my numerical covariance estimate \hat{R}_x is:

$$\hat{R}_x = \begin{bmatrix} 1.989 & -1.147 \\ -1.147 & 0.9414 \end{bmatrix} \quad (1.3)$$

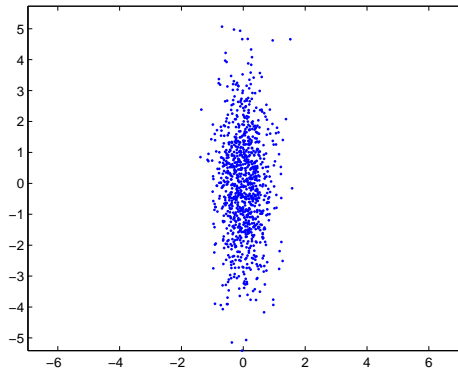


Figure 1.4: Plot of \tilde{X} , the decorrelated random vectors

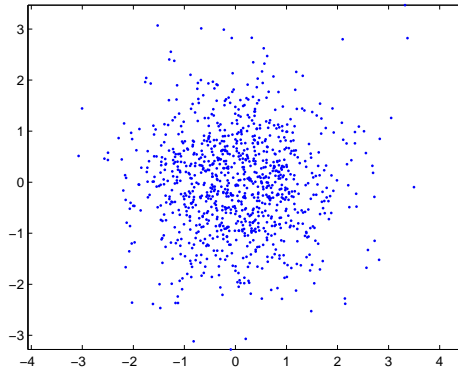


Figure 1.5: Plot of W_i , whitened random vectors

The numerical covariance estimate \tilde{R}_W is:

$$\tilde{R}_w = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad (1.4)$$

2 ESTIMATION OF EIGENVECTORS AND EIGENVALUES USING THE SINGULAR VALUE DECOMPOSITION

In this part, nothing is needed to report.

3 EIGENIMAGES, PCA, AND DATA REDUCTION

In this exercise we will compute eigenvectors associated with images (also called eigenimages) of typed English letters.

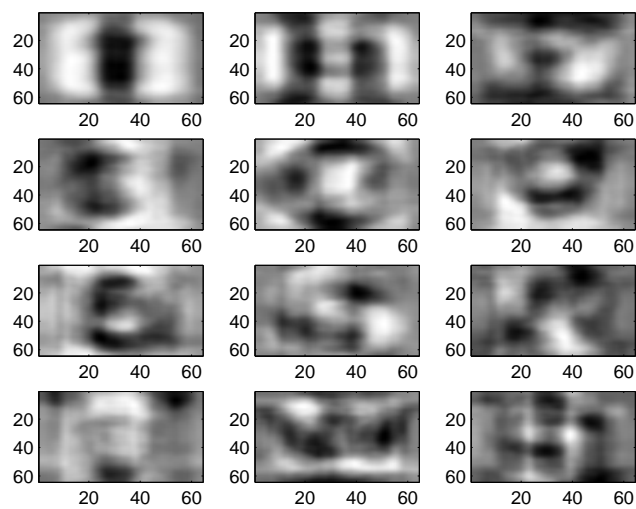


Figure 3.1: Display of the First 12 Eigenimages

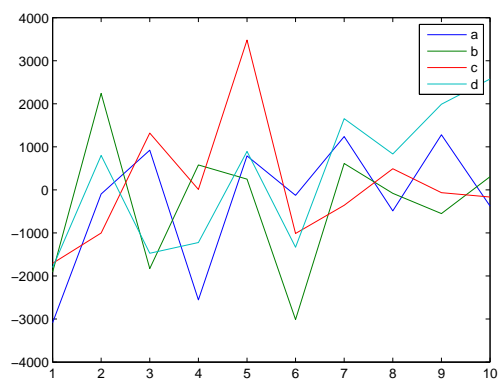


Figure 3.2: the Plots of Projection Coefficients vs. Eigenvector Number

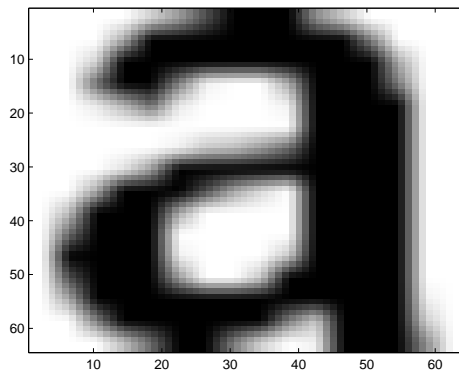


Figure 3.3: the Original Image

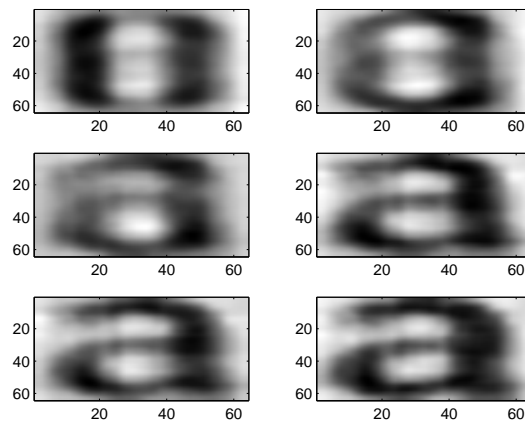


Figure 3.4: the 6 Resynthesized Versions.

4 IMAGE CLASSIFICATION

In this exercise, we will implement a classifier using the text character images from the last section as a training set. In this context the classifier will accept an input image, assumed to be of a lower-case English letter, and determine which of the 26 English letters it represents.

Table 4.1: mis-classified input image for R_x

Actual Letter Image	Mis-classified Letter Image
d	a
j	y
l	i
n	v
p	e
q	a
u	a
y	v

Table 4.2: mis-classified input image of Λ_x

Actual Letters Images	Mis-classified Letters Images
i	l
y	v

Table 4.3: mis-classified input image of R_{wc}

Actual Letters Images	Mis-classified Letters Images
g	q
y	v

Table 4.4: mis-classified input image of Λ

Actual Letters Images	Mis-classified Letters Images
f	t
y	v

Table 4.5: mis-classified input image of I

Actual Letters Images	Mis-classified Letters Images
f	t
g	q
y	v

5 CODE LISTING

```
1
2
3
4 W = zeros(2,1000);
5 for i = 1:1000
6 W(1,i) = normrnd(0,1);
7 W(2,i) = normrnd(0,1);
8 end
9 figure(1)
10 plot(W(1,:),W(2,:),'.')
11 axis('equal')
12
13 R_x = [2 -1.2
14 -1.2 1];
15 [eigV eigS] = eig(R_x);
16
17 X_hat = eigS.^(1/2)*W;
18 figure(2)
19 plot(X_hat(1,:),X_hat(2,:),'.')
20 axis('equal')
21
22 X_i = eigV * X_hat;
23 figure(3)
24 plot(X_i(1,:),X_i(2,:),'.')
25 axis('equal')
26
27 R_n = zeros(2);
28 for i = 1:1000
29 R_n(1,1) = R_n(1,1) + X_i(1,i)*X_i(1,i)/999;
30 R_n(1,2) = R_n(1,2) + X_i(1,i)*X_i(2,i)/999;
31 R_n(2,1) = R_n(2,1) + X_i(2,i)*X_i(1,i)/999;
32 R_n(2,2) = R_n(2,2) + X_i(2,i)*X_i(2,i)/999;
33 end
34
35 [eigV_Rn eig_Rn] = eig(R_n);
36 Xi_uncorr = eigV_Rn'*X_i;
37 figure(4)
38 plot(Xi_uncorr(1,:),Xi_uncorr(2,:),'.')
39 axis('equal')
40
41 Xi_hat = inv(eig_Rn).^(1/2)*Xi_uncorr;
```

```

42 figure(5)
43 plot(Xi_hat(1,:),Xi_hat(2,:),'.')
44 axis('equal')
45
46 X_hi = zeros(2);
47 for i = 1:1000
48 X_hi(1,1) = X_hi(1,1) + Xi_hat(1,i)*Xi_hat(1,i)/999;
49 X_hi(1,2) = X_hi(1,2) + Xi_hat(1,i)*Xi_hat(2,i)/999;
50 X_hi(2,1) = X_hi(2,1) + Xi_hat(2,i)*Xi_hat(1,i)/999;
51 X_hi(2,2) = X_hi(2,2) + Xi_hat(2,i)*Xi_hat(2,i)/999;
52 end
53
54 [height length] = size(X);
55 meanImage = mean(X,2);
56 X_center = zeros(4096,312);
57 for i = 1:312
58 X_center(:,i) = X(:,i) - meanImage;
59 end
60 %X_center = X_center/sqrt(311);
61
62 [U S V] = svd(X_center,0);
63
64 figure(6)
65 for i = 1:12
66 X_hat12 = U(:,i);
67 temp = vec2mat(X_hat12,64)';
68 colormap(gray(256));
69 subplot(4,3,i)
70 imagesc(temp);
71 end
72
73
74 figure(7)
75 for i = 1:4
76 temp = vec2mat(X(:,i),64)';
77 colormap(gray(256));
78 subplot(2,2,i)
79 imagesc(temp);
80 end
81
82 figure(8)
83 Projection_co = zeros(4,10);
84 for i = 1:10
85 for j = 1:4

```



```

86 Projection_co(j,i) = U(:,i)'*(X(:,j) - meanImage);
87 end
88 end
89 t = 1:10;
90 plot(t,Projection_co);
91 legend('a','b','c','d')
92
93 m = [1 5 10 15 20 30];
94 figure(9)
95 for i = 1:6
96     temp = m(i);
97     X_resyn = U(:,1:temp)*U(:,1:temp)'*X_center(:,1);
98     x = vec2mat(X_resyn + meanImage,64)';
99     colormap(gray(256))
100    subplot(3,2,i)
101    imagesc(x);
102 end
103
104 figure(10)
105 x = vec2mat(X(:,1),64)';
106 colormap(gray(256))
107 imagesc(x);
108
109 A = U(:,1:10);
110
111 empty_cell=cell(26,2);
112 params=cell2struct(empty_cell,{ 'M' , 'R' },2);
113
114 for j = 1:26
115     Y = zeros(10,12);
116     u = zeros(10,1);
117     for i = 1:12
118         Y(:,i) = A'*X_center(:,(i-1)*26 + j);
119         u = mean(Y,2);
120         R = 0;
121         for p = 1:12
122             R = R + (Y(:,p) - u)*(Y(:,p) - u)';
123         end
124         R = R/11;
125     end
126     params(j).M = u;
127     params(j).R = R;
128 end
129

```

```

130 datachar= 'abcdefghijklmnopqrstuvwxyz';
131 k = 1;
132 X_test = zeros(4096,26);
133 for ch = datachar
134 fname=sprintf('test_data/veranda/%s.tif',ch);
135 test_image = reshape(imread(fname),1,4096);
136 X_test(:,k) = test_image;
137 k = k + 1;
138 end
139 result = zeros(26,1);
140 Cri = zeros(26,26);
141 for i = 1:26
142 y = A'*(X_test(:,i) - meanImage);
143 for j = 1:26
144 t = y - params(j).M;
145 Cri(i,j) = t'*inv(params(j).R)*t + log(det(params(j).R));
146 end
147 [x loc] = min(Cri(i,:));
148 result(i) = loc;
149 end
150
151 result_1 = zeros(26,1);
152 Cri = zeros(26,26);
153 for i = 1:26
154 y = A'*(X_test(:,i) - meanImage);
155 for j = 1:26
156 t = y - params(j).M;
157 R_diag = diag(diag(params(j).R));
158 Cri(i,j) = t'*inv(R_diag)*t + log(det(R_diag));
159 end
160 [x loc] = min(Cri(i,:));
161 result_1(i) = loc;
162 end
163
164 result_2 = zeros(26,1);
165 Cri = zeros(26,26);
166 R_wc = zeros(10);
167 for i = 1:26
168 R_wc = R_wc + params(i).R;
169 end
170 R_wc = R_wc/26;
171
172 for i = 1:26
173 y = A'*(X_test(:,i) - meanImage);

```

```

174 for j = 1:26
175     t = y - params(j).M;
176     Cri(i,j) = t'*inv(R_wc)*t + log(det(R_wc));
177 end
178 [x loc] = min(Cri(i,:));
179 result_2(i) = loc;
180 end
181
182 R_wcDiag = diag(diag(R_wc));
183 result_3 = zeros(26,1);
184 Cri = zeros(26,26);
185 for i = 1:26
186     y = A'*(X_test(:,i) - meanImage);
187     for j = 1:26
188         t = y - params(j).M;
189         Cri(i,j) = t'*inv(R_wcDiag)*t + log(det(R_wcDiag));
190     end
191     [x loc] = min(Cri(i,:));
192     result_3(i) = loc;
193 end
194
195 result_4 = zeros(26,1);
196 Cri = zeros(26,26);
197 for i = 1:26
198     y = A'*(X_test(:,i) - meanImage);
199     for j = 1:26
200         t = y - params(j).M;
201         Cri(i,j) = t'*eye(10)*t + 1;
202     end
203     [x loc] = min(Cri(i,:));
204     result_4(i) = loc;
205 end
206 temp = zeros(26,2);
207 display('This is the misclassified letter using R_x')
208 for i = 1:26
209     if result(i) ~= i
210         temp(i,1) = (datachar(i));
211         temp(i,2) = (datachar(result(i)));
212         char(temp(i,:))
213     end
214 end
215
216 display('This is the misclassified letter using Lambda_x')
217

```

```

218 for i = 1:26
219     if result_1(i) ~= i
220         temp(i,1) = (datachar(i));
221         temp(i,2) = (datachar(result_1(i)));
222         char(temp(i,:))
223     end
224 end
225
226 display('This is the misclassified letter using R_wc')
227
228 for i = 1:26
229     if result_2(i) ~= i
230         temp(i,1) = (datachar(i));
231         temp(i,2) = (datachar(result_2(i)));
232         char(temp(i,:))
233     end
234 end
235
236 display('This is the misclassified letter using Lambda')
237
238 for i = 1:26
239     if result_3(i) ~= i
240         temp(i,1) = (datachar(i));
241         temp(i,2) = (datachar(result_3(i)));
242         char(temp(i,:))
243     end
244 end
245
246 display('This is the misclassified letter using I, the unit matrix')
247
248 for i = 1:26
249     if result_4(i) ~= i
250         temp(i,1) = (datachar(i));
251         temp(i,2) = (datachar(result_4(i)));
252         char(temp(i,:))
253     end
254 end

```