
Table of Contents

Q2	1
A: Covariance Matrix	1
B: Covariance matrix Eigen values and vectors	1
C: Sorting Eigen Vectors by Eigen Values descending	2
D: Mapped Data using Covariance Eigen vectors	2
E: Show the new Data is linear independent	3
F: Show original Data as linear combination of new Data and Eigen Vectors	3
G: Map data to one feature data.	4

Q2

```
clear
close all
clc
```

A: Covariance Matrix

```
X = [7 4 3
      4 1 8
      6 3 5
      8 6 1
      8 5 7
      7 2 9
      5 3 3
      9 5 8
      7 4 5
      8 2 2];
S = covarianceMat(X)

S =

    2.3222    1.6111   -0.4333
    1.6111    2.5000   -1.2778
   -0.4333   -1.2778    7.8778
```

B: Covariance matrix Eigen values and vectors

```
[eigVect, eigVals] = eig(S)

eigVect =

    0.7017    0.6990   -0.1376
   -0.7075    0.6609   -0.2505
   -0.0842    0.2731    0.9583
```

```
eigVals =  
  
    0.7499         0         0  
         0    3.6761         0  
         0         0    8.2739
```

C: Sorting Eigen Vectors by Eigen Values descending

```
eigVectOrg = eigVect;  
d = diag(eigVals);  
[~, ind] = sort(d, 'descend');  
eigVect = eigVect(:,ind)
```

```
eigVect =  
  
   -0.1376    0.6990    0.7017  
   -0.2505    0.6609   -0.7075  
    0.9583    0.2731   -0.0842
```

D: Mapped Data using Covariance Eigen vectors

```
mappedX = (X * eigVectOrg)  
  
fprintf("\t\t\t a * eig1 + b * eig2 + c * eig3 = (mappedX1 mappedX2  
    mappedX3)\n")  
n = size(X,1);  
for i=1:n  
    fprintf("%g * (%4.2f %4.2f %4.2f) + %g * (%4.2f %4.2f %4.2f) + %g  
    * (%4.2f %4.2f %4.2f) = (%4.1f %4.1f %4.1f)\n", ...  
        X(i,1), eigVectOrg(1,:), X(i,2), eigVectOrg(2,:), X(i,3),  
        eigVectOrg(3,:), mappedX(i,:));  
end
```

```
mappedX =  
  
    1.8298    8.3561    0.9101  
    1.4262    5.6417    6.8657  
    1.6672    7.5423    3.2147  
    1.2849    9.8307   -1.6450  
    1.4874   10.8083    4.3553  
    2.7397    8.6728    7.1608  
    1.1338    6.2971    1.4357  
    2.1050   11.7804    5.1760  
    1.6615    8.9022    2.8267
```

```

4.0306    7.4602    0.3151

a * eig1 + b * eig2 + c * eig3 = (mappedX1 mappedX2 mappedX3)
7 * (0.70 0.70 -0.14) + 4 * (-0.71 0.66 -0.25) + 3 * (-0.08 0.27 0.96)
= ( 1.8  8.4  0.9)
4 * (0.70 0.70 -0.14) + 1 * (-0.71 0.66 -0.25) + 8 * (-0.08 0.27 0.96)
= ( 1.4  5.6  6.9)
6 * (0.70 0.70 -0.14) + 3 * (-0.71 0.66 -0.25) + 5 * (-0.08 0.27 0.96)
= ( 1.7  7.5  3.2)
8 * (0.70 0.70 -0.14) + 6 * (-0.71 0.66 -0.25) + 1 * (-0.08 0.27 0.96)
= ( 1.3  9.8 -1.6)
8 * (0.70 0.70 -0.14) + 5 * (-0.71 0.66 -0.25) + 7 * (-0.08 0.27 0.96)
= ( 1.5 10.8  4.4)
7 * (0.70 0.70 -0.14) + 2 * (-0.71 0.66 -0.25) + 9 * (-0.08 0.27 0.96)
= ( 2.7  8.7  7.2)
5 * (0.70 0.70 -0.14) + 3 * (-0.71 0.66 -0.25) + 3 * (-0.08 0.27 0.96)
= ( 1.1  6.3  1.4)
9 * (0.70 0.70 -0.14) + 5 * (-0.71 0.66 -0.25) + 8 * (-0.08 0.27 0.96)
= ( 2.1 11.8  5.2)
7 * (0.70 0.70 -0.14) + 4 * (-0.71 0.66 -0.25) + 5 * (-0.08 0.27 0.96)
= ( 1.7  8.9  2.8)
8 * (0.70 0.70 -0.14) + 2 * (-0.71 0.66 -0.25) + 2 * (-0.08 0.27 0.96)
= ( 4.0  7.5  0.3)

```

E: Show the new Data is linear independent

The covariance matrix is diagonal

```
mappedS = covarianceMat(mappedX)
```

```
mappedS =
```

```

0.7499    -0.0000    0.0000
-0.0000    3.6761   -0.0000
0.0000   -0.0000    8.2739

```

F: Show original Data as linear combination of new Data and Eigen Vectors

```

fprintf("\t\t a * eig1 + b * eig2 + c * eig3 = (X1 X2 X3)\n")
n = size(X,1);
eigVectInverse = eigVectOrg^-1;
for i=1:n
    fprintf("%g * (%4.2f %4.2f %4.2f) + %g * (%4.2f %4.2f %4.2f) + %g * (%4.2f %4.2f %4.2f) = (%4.1f %4.1f %4.1f)\n", ...
        mappedX(i,1), eigVectInverse(1,:), mappedX(i,2),
        eigVectInverse(2,:), mappedX(i,3), eigVectInverse(3,:), X(i,:));
end

a * eig1 + b * eig2 + c * eig3 = (X1 X2 X3)

```

```

1.82978 * (0.70 -0.71 -0.08) + 8.35606 * (0.70 0.66 0.27) + 0.910074 *
(-0.14 -0.25 0.96) = ( 7.0  4.0  3.0)
1.42616 * (0.70 -0.71 -0.08) + 5.64168 * (0.70 0.66 0.27) + 6.86568 *
(-0.14 -0.25 0.96) = ( 4.0  1.0  8.0)
1.66719 * (0.70 -0.71 -0.08) + 7.54229 * (0.70 0.66 0.27) + 3.21471 *
(-0.14 -0.25 0.96) = ( 6.0  3.0  5.0)
1.28492 * (0.70 -0.71 -0.08) + 9.83071 * (0.70 0.66 0.27) + -1.64502 *
(-0.14 -0.25 0.96) = ( 8.0  6.0  1.0)
1.4874 * (0.70 -0.71 -0.08) + 10.8083 * (0.70 0.66 0.27) + 4.35525 *
(-0.14 -0.25 0.96) = ( 8.0  5.0  7.0)
2.73972 * (0.70 -0.71 -0.08) + 8.67276 * (0.70 0.66 0.27) + 7.16081 *
(-0.14 -0.25 0.96) = ( 7.0  2.0  9.0)
1.13378 * (0.70 -0.71 -0.08) + 6.29709 * (0.70 0.66 0.27) + 1.43568 *
(-0.14 -0.25 0.96) = ( 5.0  3.0  3.0)
2.10497 * (0.70 -0.71 -0.08) + 11.7804 * (0.70 0.66 0.27) + 5.17599 *
(-0.14 -0.25 0.96) = ( 9.0  5.0  8.0)
1.66146 * (0.70 -0.71 -0.08) + 8.90222 * (0.70 0.66 0.27) + 2.82668 *
(-0.14 -0.25 0.96) = ( 7.0  4.0  5.0)
4.03058 * (0.70 -0.71 -0.08) + 7.46024 * (0.70 0.66 0.27) + 0.31512 *
(-0.14 -0.25 0.96) = ( 8.0  2.0  2.0)

```

G: Map data to one feature data.

```
reductionMappedX = X * eigVect(:,1:2)
```

```
reductionMappedX =
```

```

0.9101    8.3561
6.8657    5.6417
3.2147    7.5423
-1.6450    9.8307
4.3553   10.8083
7.1608    8.6728
1.4357    6.2971
5.1760   11.7804
2.8267    8.9022
0.3151    7.4602

```

The Feature Reduction Error is:

```
err = min(diag(eigVals))
```

```
err =
```

```
0.7499
```

Published with MATLAB® R2017a