A BRIEF ILLUSTRATION OF PCA

INT301 Bio-computation, Week 10, 2025

Data Simulation

```
function PCA_demo

%%  generate the data
%%  define the direction of the real data
rd          =       [1;1];
rd          =       rd/norm(rd);            %normlize the data direction
rd_std      =       1.5;                    %define the standard deviation of the real data
rd_num      =       500;                    %define the number of data
noi_std     =       0.05;                   %define the standard deviation of the noise

%%  simulate the observations
rd_std_vec          =       randn([1,rd_num])*rd_std;
for k =1:rd_num
    rd_data(:,k)    =       rd*rd_std_vec(k);
end
noi_data            =       randn([length(rd),rd_num])*noi_std;
ob_data             =       rd_data + noi_data;

%%  show the data
figure;
plot(ob_data(1,:), ob_data(2,:),'b*')
legend('Observations')
disp('The direction of the data distribution:')
disp(rd)
```

向量 (1;1)

规一到 → $\left(\frac{1}{\sqrt{2}}; \frac{1}{\sqrt{2}}\right)$

→ std (data)

→ num (data)

→ 噪音std

$N(0,I)$

生成一个 1×500 行向量，每一个元素服从 $N(0,1.5)$

执行500次

i为k 随机变量和单位向量相乘. rd_data 列表，存入第k列

2×500 $N(0,0.05)$ 队列

$\left(\frac{z_1}{\sqrt{2}}, \frac{z_1}{\sqrt{2}}\right) + (z_2, z_2)$

$\Rightarrow$

Data Simulation

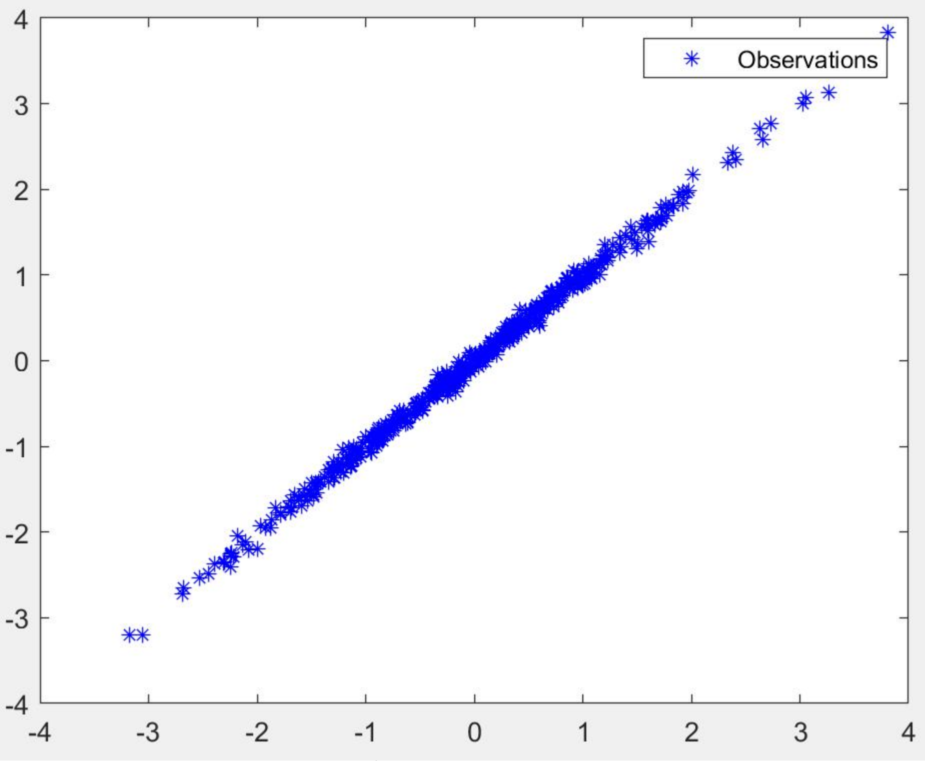
```
%% generate the data
%% define the direction of the real data
rd          =     [1;1];
rd          =     rd/norm(rd);          %normlize
rd_std      =     1.5;                  %define th
rd_num      =     500;                  %define th
noi_std     =     0.05;                  %define t

%% simulate the observations
rd_std_vec        =      randn([1,rd_num])*rd_
for k =1:rd_num
    rd_data(:,k)  =      rd*rd_std_vec(k);
end
noi_data          =      randn([length(rd),rd_
ob_data           =      rd_data + noi_data;

%% show the data
figure;
plot(ob_data(1,:), ob_data(2,:),'b*')
legend('Observations')
disp('The direction of the data distribution:')
disp(rd)
```

添加图例

创建绘图窗口

blue.

命令行显示




```
>> PCA_demo
The direction of the data distribution:
    0.7071
    0.7071
```

SVD

```
%%  SVD analysis
[U, S, V]        =      svd(ob_data);
singular_vec     =      diag(S);
disp('The eigenvectors:')
disp(U)

disp('The singular values are given:')
disp(singular_vec(1:2))

disp('The estimations of std:')
disp(singular_vec(1:2)/sqrt(rd_num))
```

SVD

2×2 2×500 500×500 2×500

```
%% SVD analysis
[U, S, V]          =      svd(ob_data);
singular_vec       =      diag(S);
disp('The eigenvectors:')
disp(U)

disp('The singular values are given:')
disp(singular_vec(1:2))

disp('The estimations of std:')
disp(singular_vec(1:2)/sqrt(rd_num))
```

→提取特征值.

接近$(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$

```
The eigenvectors:
   -0.7065    0.7077
   -0.7077   -0.7065

The singular values are given:
   33.6941
    1.0605

The estimations of std:
    1.5068
```

$\Rightarrow \sigma_i = \frac{\sqrt{\lambda_i N}}{\sqrt{N}}$

```
>> PCA_demo
The direction of the data distribution:
    0.7071
    0.7071
```

SVD

```
%%  SVD analysis
[U, S, V]          =     svd(ob_data);
singular_vec       =     diag(S);
disp('The eigenvectors:')
disp(U)

disp('The singular values are given:')
disp(singular_vec(1:2))

disp('The estimations of std:')
disp(singular_vec(1:2)/sqrt(rd_num))
```

```
The eigenvectors:
   -0.7065    0.7077
   -0.7077   -0.7065

The singular values are given:
   33.6941
    1.0605

The estimations of std:
    1.5068
```

```
%%  generate the data
%%  define the direction of the real data
rd          =     [1;1];
rd          =     rd/norm(rd);      %normlize the data direction
rd_std      =     1.5;              %define the standard deviation of the real data
rd_num      =     500;              %define the number of data
noi_std     =     0.05;              %define the standard deviation of the noise
```

Diagonal decomposition of the correlation matrix

```
%%  Diagonal decomposition of the correlation matrix
R_Mat  2x2        = ob_data*ob_data';=ob_data^T
disp('the correlation matrix of the data is given:');
disp(R_Mat)
矩阵
   对角matrix
[V, D]            = eig(R_Mat);
disp('The eigenvectors are given:')
disp(V)
disp('The eigenvalues are given:')
disp(diag(D))
disp('The singular vaules are given in another form:')
disp(sqrt(diag(D)))
```

```
the correlation matrix of the data is given:
  567.1940   567.0836
  567.0836   569.2243

The eigenvectors are given:
   -0.7077     0.7065
    0.7065     0.7077

The eigenvalues are given:
   1.0e+03 *

    0.0011
    1.1353

The singular vaules are given in another form:
    1.0605
   33.6941
```

Diagonal decomposition of the correlation matrix

```
%%  Diagonal decomposition of the correlation matrix
R_Mat               = ob_data*ob_data';
disp('the correlation matrix of the data is given:');
disp(R_Mat)

[V, D]              = eig(R_Mat);
disp('The eigenvectors are given:')
disp(V)
disp('The eigenvalues are given:')
disp(diag(D))
dis                                          r form:')
dis
```

```
The eigenvectors:
   -0.7065    0.7077
   -0.7077   -0.7065

The singular values are given:
   33.6941
    1.0605

The estimations of std:
    1.5068
```

```
the correlation matrix of the data is given:
  567.1940   567.0836
  567.0836   569.2243

The eigenvectors are given:
   -0.7077    0.7065
    0.7065    0.7077

The eigenvalues are given:
   1.0e+03 *

    0.0011
    1.1353

The singular vaules are given in another form:
    1.0605
   33.6941
```

Diagonal decomposition of the correlation matrix

```
%%  Diagonal decomposition of the correlation matrix
R_Mat               = ob_data*ob_data';
disp('the correlation matrix of the data is given:');
disp(R_Mat)

[V, D]              = eig(R_Mat);
disp('The eigenvectors are given:')
disp(V)
disp('The eigenvalues are given:')
disp(diag(D))
disp('The singular vaules
disp(sqrt(diag(D)))
```

```
The eigenvectors:
   -0.7065    0.7077
   -0.7077   -0.7065

The singular values are given:
   33.6941
    1.0605

The estimations of std:
    1.5068
```

```
the correlation matrix of the data is given:
  567.1940  567.0836
  567.0836  569.2243

The eigenvectors are given:
   -0.7077    0.7065
    0.7065    0.7077

The eigenvalues are given:
   1.0e+03 *

    0.0011
    1.1353

The singular vaules are given in another form:
    1.0605
   33.6941
```

Projection and reconstruction

```
%% Reconstruction with eigenvectors
%% Projection with the maximum eignvalue related eigenvector
ob_pro              = V(:,2)'*ob_data;
pro_std             = std(ob_pro)*sqrt(rd_num);
disp('the std of the projections');
disp(pro_std)

%% reconstructions with the maximum eignvalue related eigenvector
pro_data             = V(:,2)*V(:,2)'*ob_data;
figure;
plot(rd_data(1,:), rd_data(2,:),'b*')
hold on
plot(pro_data(1,:), pro_data(2,:),'ro')
legend('real data', 'reconstructions')
```

降维

重构

Projection and reconstruction

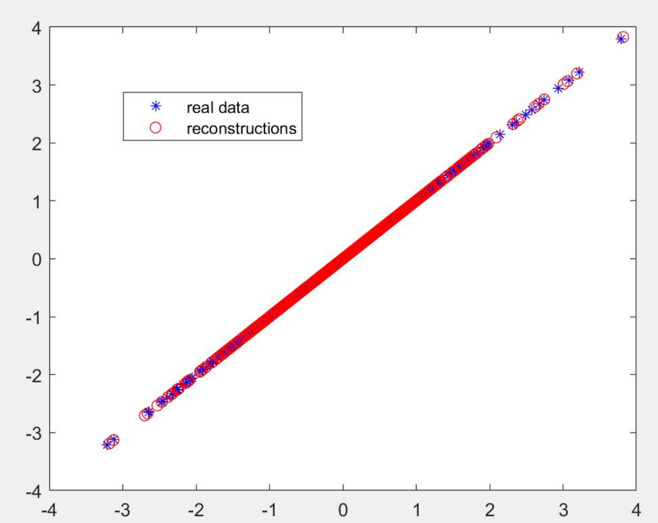
```
%% Reconstruction with eigenvectors
%% Projection with the maximum eignvalue related eigenvector
ob_pro              =  V(:,2)'*ob_data;
pro_std             =  std(ob_pro)*sqrt(rd_num);
disp('the std of the projections');
disp(pro_std)

%% reconstructions with the maximum eignvalue related eigenvector
pro_data              = V(:,2)*V(:,2)'*ob_data;
figure;
plot(rd_data(1,:), rd_data(2,:),'b*')
hold on
plot(pro_data(1,:), pro_data(2,:),'ro')
legend('real data', 'reconstructions')
```

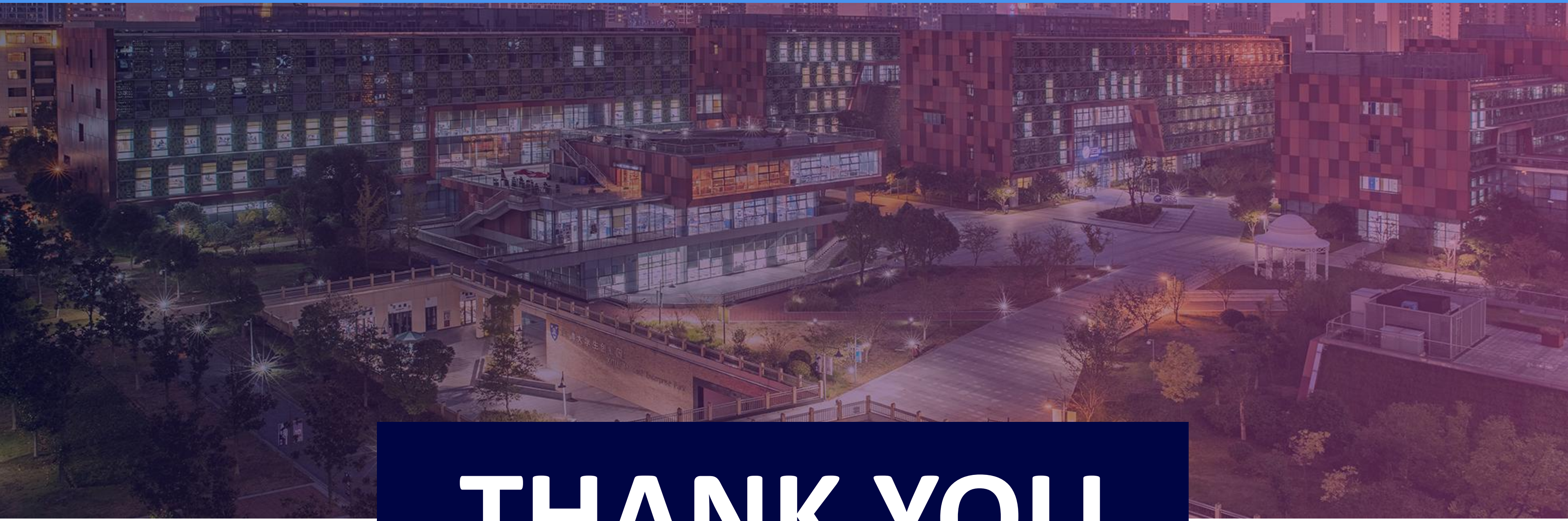
```
the std of the projections
   33.7202
```

```
The eigenvectors:
   -0.7065    0.7077
   -0.7077   -0.7065

The singular values are given:
   33.6941
    1.0605

The estimations of std:
    1.5068
    0.0474
```

Projection and reconstruction

```
%% Reconstruction with eigenvectors
%% Projection with the maximum eignvalue related eigenvector
ob_pro                  =  V(:,2)'*ob_data;
pro_std                 =  std(ob_pro)*sqrt(rd_num);
disp('the std of the projections');
disp(pro_std)

%% reconstructions with the maximum eignvalue related eigenvector
pro_data                 = V(:,2)*V(:,2)'*ob_data;
figure;
plot(rd_data(1,:), rd_data(2,:),'b*')
hold on
plot(pro_data(1,:), pro_data(2,:),'ro')
legend('real data', 'reconstructions')
```





THANK YOU

VISIT US

WWW.XJTLU.EDU.CN

FOLLOW US

@XJTLU

Xi'an Jiaotong-Liverpool University
西交利物浦大學