第九次课堂作业

南子谦 3210104676

codes are on MATLAB

Exercise 7.1

third right singular vector:

```
ans =

   -0.1559
    0.4814
    0.1814
   -0.7085
   -0.3543
   -0.2890
```

second singular value:

ans =

```
0.1994
```

fourth left singular vector:

```
ans =

    0.1255
    0.0156
    0.0436
    0.2362
    0.2106
    0.3250
   -0.2208
   -0.1704
   -0.0099
    0.1839
   -0.3154
    0.1080
    0.1406
    0.1384
   -0.1896
   -0.0855
    0.0532
    0.0863
   -0.1979
    0.0593
    0.1471
    0.2691
    0.1068
   -0.1467
   -0.0916
```

```
    0.3265
   -0.3989
    0.0989
   -0.0148
   -0.0856
```

rank of A: 6

F_norm =

```
0.0016
```

two_norm =

```
0.0016
```

F_norm_PCA =

```
0.0014
```

two_norm_PCA =

```
0.0013
```

code:

```
% p173 exercise 7.1
clear
clc

A = readmatrix("A.csv");
[m,n] = size(A);
[U,Sigma,V] = svd(A);
% third right singular vector
disp("third right singular vector: ")
V(:,3)
% second singular value
disp("second singular value: ")
Sigma(2,2)
% fourth left singular vector
disp("fourth left singular vector: ")
U(:,4)
% rank of A
disp("rank of A: "+rank(Sigma))

% compute Ak
k=2;
Ak = zeros(m,n);
for i = 1:k
    Ak = Ak + U(:,i)*Sigma(i,i)*V(:,i)';
end
% F-norm^2 of A-Ak
F_norm = 0;
for i = 1:n
    F_norm = F_norm + norm(A(:,i)-Ak(:,i),2)^2;
end
F_norm = F_norm
% 2-norm^2 of A-Ak
two_norm = norm(A-Ak,2)^2

% center A, run PCA
A_centered = A - mean(A,2);
[U_centered,Sigma_centered,V_centered] = svd(A_centered);
B = V_centered(:,1:k);
% report
pi_B=A_centered*B*B';
F_norm_PCA = 0;
for i = 1:n
    F_norm_PCA = F_norm_PCA + norm(A_centered(:,i)-pi_B(:,i),2)^2;
end
```

```
F_norm_PCA = F_norm_PCA
two_norm_PCA = norm(A_centered-pi_B,2)^2
```

Exercise 7.5

1

the distance is a metric

$$
\begin{gathered}
||x_i - x_j||_2 \geq 0 \\
||x_i - x_j||_2 = 0 \Leftrightarrow x_i = x_j \\
||x_i - x_j||_2 = ||x_j - x_i||_2 \\
||a + b||_2 \leq ||a||_2 + ||b||_2
\end{gathered}
$$

the fourth follows directlly from Euclidean distance.

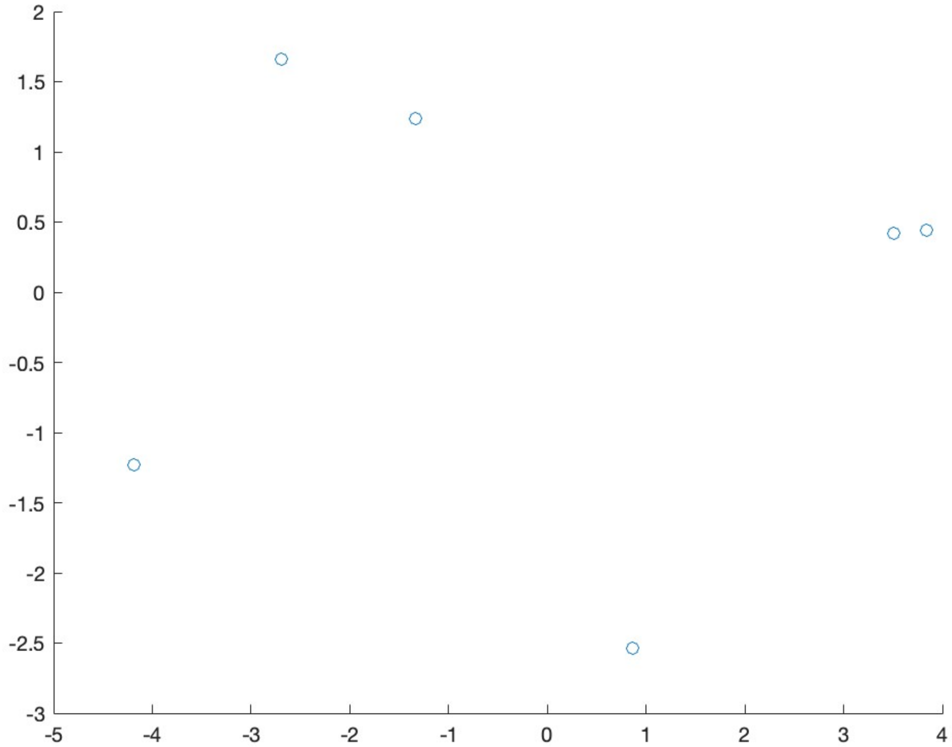
2

weirdly, the eigenvalues of M are not all positive, I have to choice the two largest positive eigenvalues.

```
Q =

    3.5030    0.4194
    3.8424    0.4428
    0.8664   -2.5330
   -1.3332    1.2367
   -2.6909    1.6641
   -4.1877   -1.2300
```

3



error of ZZT-XXT

0	-0.859809432689687	0.558299627451818	-0.195269207723412	0.117668553371989	0
-0.859809432689687	0	1.50851526503427	0.436078683284014	0.346400461382906	0
0.558299627451818	1.50851526503427	0	-0.0354841167309905	0.00180432489856575	1
-0.195269207723412	0.436078683284014	-0.0354841167309905	0	-0.976635029010074	1
0.117668553371989	0.346400461382906	0.00180432489856575	-0.976635029010074	0	2
0.165576535013777	0.00246284470576796	1.91936369872212	1.27271116253566	2.35832431445521	0

code:

```
% exercise 7.5
clear
clc

D = [0,1.2,3.4,5.1,6.2,7.7;
    1.2,0,2.7,4.8,6.3,8.2;
    3.4,2.7,0,4.4,5.5,3.3;
    5.1,4.8,4.4,0,2.4,2.5;
    6.2,6.3,5.5,2.4,0,0.9;
    7.7,8.2,3.3,2.5,0.9,0
];

[m,~] = size(D);

Cn = eye(m)-ones(m,m)/m;
M = Cn*(D.^2)*Cn/(-2);
[L,V] = eig(M);
eigen_value = diag(V);
[~,id] = sort(eigen_value,"DESCEND");
Q = L(:,id(1:2))*diag(eigen_value(id(1:2)).^(0.5))

scatter(Q(:,1),Q(:,2))
W = zeros(m,m);
% W = {||xi-xj||}
for j = 1:m
    W(:,j) = sum((Q-Q(j,:)).^2,2);
end
error = sqrt(W)-D
```

Exercise 7.10

1

$Q = A * V_k * V_k^T$

2

$$||D - D_Q||_F^2 = \sum_{i,j}(D_{i,j}^2 - 2D_{i,j}D_{Qi,j} + D_{Q,i,j}^2)$$

即证

$$\sum(D_{i,j} - D_{Qi,j})D_{Qi,j} \geqslant 0$$

即证

$$\begin{aligned} D_{i,j}^2 - D_{Qi,j}^2 &\geqslant 0 \\ \because D_{i,j}^2 - D_{Qi,j}^2 &= ||a_i - a_j||_2^2 - ||(a_i - b_j)V_kV_k^T||_2^2 \\ &= ||\sum_i^d < b, v_i > v_i||_2^2 - ||\sum_i^k < b, v_i > v_k||_2^2 \end{aligned}$$

$$= \sum_{k+1}^{d} < b, v_i >^2 \geq 0$$

得证

3

The reason F norm error is small is that I make distinct points number to be less than 3, so when PCA no information is lost, F norm error is small.

The reason why inf norm error is large is that the numbers of samples can be arbitrarily large, so the distance between two points is not small.

code:

```
clear
clc
%construct samples
m = 1000;n = 3;k = 2;
distinct = 2;

fea = zeros(m,n);
fea(1:distinct,:) = 10000000*rand(distinct,n);
for i = distinct+1:m
    rand_int = ceil(rand(1,1)*distinct);
    fea(i,:) = fea(rand_int,:);
end

%column normalize
for i = 1:n
    fea(:,i) = fea(:,i) - mean(fea(:,i));
end

% PCA
W = zeros(3,k);
[~,~,V] = svd(fea);
W = V(:,1:k);
Distance = zeros(m,m);
Y = fea*W*W';
% W = {||xi-xj||}
for j = 1:m
    Distance(:,j) = sum((fea-fea(j,:)).^2,2);
end

Distance_Q = zeros(m,m);
for j = 1:m
    Distance_Q(:,j) = sum((Y-Y(j,:)).^2,2);
end

Fnorm_error = norm(Distance_Q-Distance,'fro')^2
infnorm_error = max(max(abs(Distance-Distance_Q)))
```