

# Homework Set 2, CPSC 8420, Fall 2023

Yue Zhang

**Due 10/26/2023, Thursday, 11:59PM EST**

## 1 Problem 1

For PCA, from the perspective of maximizing variance, please show that the solution of  $\phi$  to maximize  $\|\mathbf{X}\phi\|_2^2$ , *s.t.*  $\|\phi\|_2 = 1$  is exactly the first column of  $\mathbf{U}$ , where  $[\mathbf{U}, \mathbf{S}] = \text{svd}(\mathbf{X}^T \mathbf{X})$ . (Note: you need prove why it is optimal than any other reasonable combinations of  $\mathbf{U}_i$ , say  $\hat{\phi} = 0.8 * \mathbf{U}(:, 1) + 0.6 * \mathbf{U}(:, 2)$  which also satisfies  $\|\hat{\phi}\|_2 = 1$ .)

Suppose

$$\phi = \sum_1^n \alpha_i \mathbf{U}_i, \text{ where } \sum_0^n \alpha_i^2 = 1$$

Then

$$\|\mathbf{X}\phi\|_2^2 = \phi^T (\mathbf{U} \mathbf{S} \mathbf{U}^T) \phi = \phi^T (\sum \mathbf{S}_{ii} \mathbf{U}_i \mathbf{U}_i^T) \phi \quad (1)$$

$$\text{when, } i = 1 \rightarrow \|\mathbf{X}\phi\|_2^2 = \mathbf{S}_{11} \quad (2)$$

$$\text{otherwise, } m \leq n \rightarrow \|\mathbf{X}\phi\|_2^2 = \sum_1^m \alpha_i^2 \mathbf{S}_{ii} < \mathbf{S}_{11} \sum_1^m \alpha_i^2 = \mathbf{S}_{11} \quad (3)$$

So taking  $\phi = \mathbf{U}_1$  is optimal than any other reasonable combinations of  $\mathbf{U}$

## 2 Problem 2

Given matrix  $\mathbf{X} \in \mathbb{R}^{n \times p}$  (assume each column is centered already), where  $n$  denotes sample size while  $p$  feature size. To conduct PCA, we need find eigenvectors to the largest eigenvalues of  $\mathbf{X}^T \mathbf{X}$ , where usually the complexity is  $\mathcal{O}(p^3)$ . Apparently when  $n \ll p$ , this is not economic when  $p$

is large. Please consider conducting PCA based on  $\mathbf{X}\mathbf{X}^T$  and obtain the eigenvectors for  $\mathbf{X}^T\mathbf{X}$  accordingly and use experiment to demonstrate the acceleration.

## 2.1 eVec

Assume  $\mathbf{v}$  is an eigenvector of  $\mathbf{X}\mathbf{X}^T$  to eigenvalue  $\lambda$ . Then it holds

$$\mathbf{X}\mathbf{X}^T\mathbf{v} = \lambda\mathbf{v}$$

and

$$\mathbf{X}^T\mathbf{X}\mathbf{X}^T\mathbf{v} = \mathbf{X}^T\lambda\mathbf{v} = \lambda\mathbf{X}^T\mathbf{v}$$

, hence  $\mathbf{X}^T\mathbf{v}$  is an eigenvector of  $\mathbf{X}^T\mathbf{X}$  with eigenvalue  $\lambda$

## 2.2 Exp

```
n = 3; p = 10;
X = rand(n,p);
[V,D] = svd(X*X');
%[nV,nD] = svd(X'*X);

err = zeros(1,n);
for i = 1:n
    % ith eVec and EVAL
    v = V(:,i);
    lambda = D(i,i);

    nV = X'*v;
    err(i) = norm(X'*X*nV - lambda*nV,2);
end
err
```

```
err =
1.0e-14 *
0.9770 0.1713 0.1028
```

## 3 Problem 3

Let's revisit Least Squares Problem: minimize  $\frac{1}{2}\|\mathbf{y} - \mathbf{A}\boldsymbol{\beta}\|_2^2$ , where  $\mathbf{A} \in \mathbb{R}^{n \times p}$ .

1. Please show that if  $p > n$ , then vanilla solution  $(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{y}$  is not applicable any more.
2. Let's assume  $\mathbf{A} = [1, 2, 4; 1, 3, 5; 1, 7, 7; 1, 8, 9]$ ,  $\mathbf{y} = [1; 2; 3; 4]$ . Please show via experiment results that Gradient Descent method will obtain the optimal solution with Linear Convergence rate if the learning rate is fixed to be  $\frac{1}{\sigma_{max}(\mathbf{A}^T\mathbf{A})}$ , and  $\boldsymbol{\beta}_0 = [0; 0; 0]$ .

3. Now let's consider ridge regression: minimize  $\frac{1}{2}\|\mathbf{y} - \mathbf{A}\boldsymbol{\beta}\|_2^2 + \frac{\lambda}{2}\|\boldsymbol{\beta}\|_2^2$ , where  $\mathbf{A}, \mathbf{y}, \boldsymbol{\beta}_0$  remains the same as above while learning rate is fixed to be  $\frac{1}{\lambda + \sigma_{\max}(\mathbf{A}^T \mathbf{A})}$  where  $\lambda$  varies from 0.1, 1, 10, 100, 200, please show that Gradient Descent method with larger  $\lambda$  converges faster.
1.  $\mathbf{A}^T \mathbf{A}$  is a  $p \times p$  matrix, but the  $\text{rank}(\mathbf{A}^T \mathbf{A}) \leq \min(n, p) < n \implies \mathbf{A}^T \mathbf{A}$  is not invertable
2. See figure 1

```

Itr=50000;
err=zeros(Itr,1);

A=[1 2 4;1 3 5; 1 7 7; 1 8 9];
y=[1;2;3;4];

beta_star = (A'*A)\(A'*y);
opt = 0.5*norm(y-A*beta_star)^2;

[U,S,V]=svd(A'*A);
L = S(1,1);
beta = [0;0;0];

for i=1:Itr
    beta = beta - 1/L*(A'*A*beta-A'*y);
    err(i)=0.5*norm(y-A*beta)^2-opt;
end
%plot(1:Itr,err)
semilogy(1:Itr,err)

```

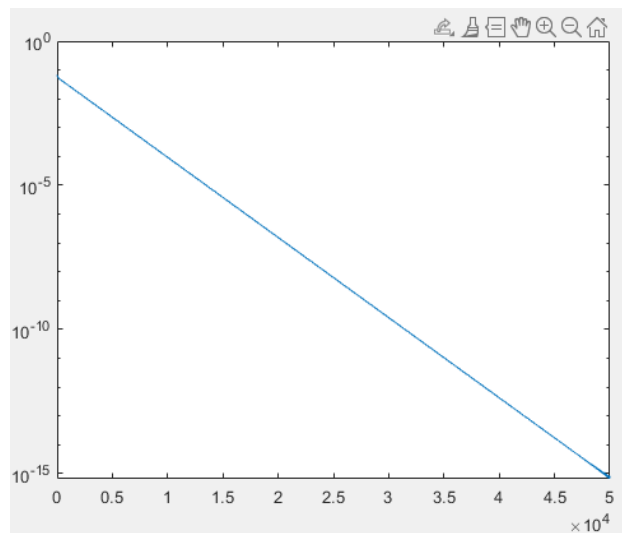


Figure 1: Q3-2

:

3. See figure 2

```
Itr=1000;
err=zeros(Itr,1);

A=[1 2 4;1 3 5; 1 7 7; 1 8 9];
y=[1;2;3;4];
%lambda_list=[200];
lambda_list=[0.1, 1 , 10, 100, 200];

for lambda = lambda_list
    beta_star = (A'*A + lambda*eye(3))\ (A'*y);
    opt = 0.5*norm(y-A*beta_star)^2 + 0.5*lambda*norm(beta_star)^2;

    [U,S,V]=svd(A'*A);
    L = S(1,1) + lambda;
    beta = [0;0;0];
    for i=1:Itr
        beta = beta - 1/L*((A'*A+lambda*eye(3))*beta-A'*y);
        err(i)=0.5*norm(y-A*beta)^2 + 0.5*lambda*norm(beta)^2 - opt;
    end
    x = 1:Itr;
    %plot(x,err)
    semilogy(1:Itr,err)
    hold on
end
```

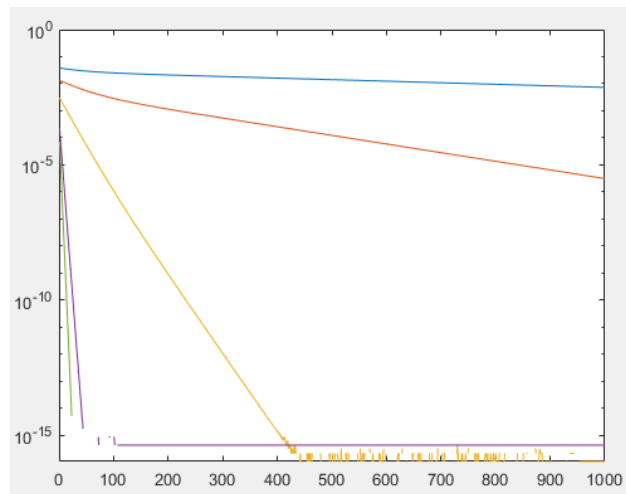


Figure 2: Q3-3

:

## 4 Problem 4

We consider matrix completion problem. As we discussed in class, the main issue of *softImpute* (*Matrix Completion via Iterative Soft-Thresholded SVD*) is when the matrix size is large, conducting *SVD* is computational demanding. Let's recall the original problem where  $\mathbf{X}, \mathbf{Z} \in \mathbb{R}^{n \times d}$ :

$$\min_{\mathbf{Z}} \frac{1}{2} \|P_{\Omega}(\mathbf{X}) - P_{\Omega}(\mathbf{Z})\|_F^2 + \lambda \|\mathbf{Z}\|_* \quad (4)$$

People have found that instead of finding optimal  $\mathbf{Z}$ , it might be better to make use of *Burer-Monteiro* method to optimize two matrices  $\mathbf{A} \in \mathbb{R}^{n \times r}, \mathbf{B} \in \mathbb{R}^{d \times r} (r \geq \text{rank}(\mathbf{Z}^*))$  such that  $\mathbf{AB}^T = \mathbf{Z}$ . The new objective is:

$$\min_{\mathbf{A}, \mathbf{B}} \frac{1}{2} \|P_{\Omega}(\mathbf{X} - \mathbf{AB}^T)\|_F^2 + \frac{\lambda}{2} (\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2). \quad (5)$$

- Assume  $[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \text{svd}(\mathbf{Z})$ , show that if  $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}^{\frac{1}{2}}, \mathbf{B} = \mathbf{V}\mathbf{\Sigma}^{\frac{1}{2}}$ , then Eq. (5) is equivalent to Eq. (4).
- The *Burer-Monteiro* method suggests if we can find  $\mathbf{A}^*, \mathbf{B}^*$ , then the optimal  $\mathbf{Z}$  to Eq. (4) can be recovered by  $\mathbf{A}^* \mathbf{B}^{*T}$ . It boils down to solve Eq. (5). Show that we can make use of least squares with ridge regression to update  $\mathbf{A}, \mathbf{B}$  row by row in an alternating minimization manner as below. Assume  $n = d = 2000, r = 200$ , please write program to find  $\mathbf{Z}^*$ .

```
T ← 100, i ← 1 % you can also set T to be other number instead of 100
if i ≤ T then
    update A row by row while fixing B
    update B row by row while fixing A
    i ← i + 1
end if
```

### 4.1

It is easy to prove that the parts in front of the plus sign in the two objects are equal

$$\frac{1}{2} \|P_{\Omega}(\mathbf{X}) - P_{\Omega}(\mathbf{Z})\|_F^2 = \frac{1}{2} \|P_{\Omega}(\mathbf{X} - \mathbf{AB}^T)\|_F^2 \quad (6)$$

since  $P_{\Omega}(\mathbf{X}) - P_{\Omega}(\mathbf{Z}) = P_{\Omega}(\mathbf{X} - \mathbf{Z}) = P_{\Omega}(\mathbf{X} - \mathbf{AB}^T)$ .

For the part behind the addition sign, since

$$\begin{aligned} \|\mathbf{A}\|_F^2 &= \|\mathbf{U}\mathbf{\Sigma}^{\frac{1}{2}}\|_F^2 = \text{trace}(\mathbf{\Sigma}^{\frac{1}{2}}\mathbf{U}^T\mathbf{U}\mathbf{\Sigma}^{\frac{1}{2}}) = \text{trace}(\mathbf{\Sigma}) \\ \|\mathbf{B}\|_F^2 &= \|\mathbf{V}\mathbf{\Sigma}^{\frac{1}{2}}\|_F^2 = \text{trace}(\mathbf{\Sigma}^{\frac{1}{2}}\mathbf{V}^T\mathbf{V}\mathbf{\Sigma}^{\frac{1}{2}}) = \text{trace}(\mathbf{\Sigma}) \\ \|\mathbf{Z}\|_* &= \text{trace}(\mathbf{\Sigma}) \end{aligned}$$

, we can get

$$\frac{\lambda}{2} (\|\mathbf{A}\|_F^2 + \|\mathbf{B}\|_F^2) = \lambda \|\mathbf{Z}\|_* \quad (7)$$

From (6) and (7), we can tell (4) is equivalent to (5)

## 4.2

see figure 3

```
n = 2000;d = 2000;
r = 200;
T = 100;
alpha = 0.00001;
lambda = 1.0;

R = sprand(n,d,0.01);

A = rand(n,r);
B = rand(d,r);

B_t = B';

errlist=zeros(T,1);

for t = 1:T
%   for row = 1:n
%       for col = 1:r
%           if not(R(row,col)==0)
%               eij = R(row,col) - A(row,:)*B_t(:,col);
%               for rr = 1:r
%                   A(row,rr) = A(row,rr) + alpha *(eij*B_t(rr,col) - lambda*A(row,rr));
%                   B_t(rr,col) = B_t(rr,col) + alpha *(eij*A(row,rr) - lambda*B_t(rr,col));
%               end
%           end
%       end
%   end
%   A(row,:) = A(row,:)+alpha *(eij*B(col,:) - lambda*A(row,:));
%   B(col,:) = B(col,:) + alpha *(eij*A(row,:) - lambda*B(col,:));
%   end

[ row,col] = find(R);
for i = 1:size(row)
    disp([t,i]);
    eij = R(row(i),col(i)) - A(row(i,:)*B_t(:,col(i));
    A(row(i,:)) = A(row(i,:)+alpha *(eij*B(col(i),:) - lambda*A(row(i,:));
    B(col(i,:)) = B(col(i,:)) + alpha *(eij*A(row(i,:)) - lambda*B(col(i,:));
end

errlist(t)= norm(R-A*B_t,"fro");
end
plot(1:T,errlist);
Z_star = A*B_t;

:
```

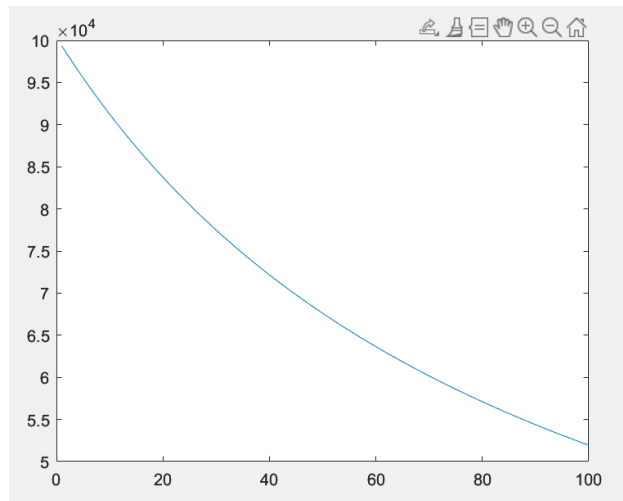


Figure 3: Q4-2