Numerical Linear Algebra
Programming Assignment #11

2015-17231
박우정

Exercise 5.11. Implement the SVD algorithm.

Singular Value Decomposition을 Gram_Schmidt decomposition을 이용하여 $A = U\Sigma V^*$로 decompose하는 MATLAB Code를 구현해보았다. 먼저 사용한 Gram_Schmidt 함수에 대한 MATLAB 코드는 아래와 같다.

```
function [Q,R]=QR_H(A)
n=size(A,1); Q=zeros(n); R=zeros(n);
for j=1:n
    v=A(:,j);
 for i=1:j-1
       R(i,j)=Q(:,i)'*A(:,j);
       v=v-R(i,j)*Q(:,i);
 end
 R(j,j)=norm(v);
 Q(:,j)=v/R(j,j);
end
```

이 함수를 이용한 SVD Code는 다음과 같이 작성하였다.

```
function [U,S,V]=svd_ex(A)
Err=realmax; %Preventing for stopping the program.
sizeA=size(A);
cntmax=100*max(sizeA);cnt=0; %For precision
U=eye(sizeA(1));S=A';V=eye(sizeA(2));
while cnt<cntmax
    [Q,S]=QR_H(S');U=U*Q; %QR_H is the decomposition using Gram_Schmidt
    [Q,S]=QR_H(S');V=V*Q;
    e=triu(S,1);
    E=norm(e(:));
    F=norm(diag(S));
    if F==0, F=1; end
    Err=E/F;
    cnt=cnt+1;
end
%For fixing the sign of Sigma
sigma=diag(S);
for i=1:length(sigma)
    sigmad=sigma(i);
    sigma(i,i)=abs(sigmad);
    if sigmad<0
       u(:,i)=-u(:,i);
    end
end
return;
```

프로그램이 알맞게 작동하는지 검토하기 위해, 임의로 0과 1사이의 실수를 element로 갖는 n by n행렬을 생성해주는 MATLAB 내장함수 RAND(n)을 이용하여 $n=4,5,6,7$일 때 검정해보았다. 결과는 아래와 같다.

```
tol=1.0e-13;
for n=4:7
    A=rand(n);
    [u,s,v]=svd_ex(A);
    tol=tol/10;
    if(norm(A-u*s*v')>tol)
    fprintf('The computed solution seems to be wrong at %f \n',n);
    end
end
```

이를 실행하면

```
>> svd_ex_test
The computed solution seems to be wrong at 5.000000
The computed solution seems to be wrong at 6.000000
The computed solution seems to be wrong at 7.000000
```

이를 얻는데, 이는 $(A-U\Sigma V^*)$의 l^2-norm이 주어진 tolerance를 10^{-14}일 때만 유의하고, 나머지는 허용오차를 통과한다는 뜻이므로, 허용오차=10^{-14} 이내에서 매우 유의미한 분해라고 결론내릴 수 있다. 이는 충분히 작으므로 알맞게 프로그래밍되었다고 할 수 있다.

Exercise 6.1.

6-th order이므로 Least Square가 되게 하는 coefficient를 편미분을 이용하여 정규방정식을 만들어서 식을 구하면 다음과 같다.

$$
\begin{bmatrix}
\sum_{k=1}^{10} t_k^{12} & \sum_{k=1}^{10} t_k^{11} & \sum_{k=1}^{10} t_k^{10} & \sum_{k=1}^{10} t_k^{9} & & & \\
\sum_{k=1}^{10} t_k^{11} & \sum_{k=1}^{10} t_k^{10} & \sum_{k=1}^{10} t_k^{9} & \sum_{k=1}^{10} t_k^{8} & \cdots & & \sum_{k=1}^{10} t_k^{6} \\
\sum_{k=1}^{10} t_k^{10} & \sum_{k=1}^{10} t_k^{9} & \sum_{k=1}^{10} t_k^{8} & \sum_{k=1}^{10} t_k^{7} & & & \vdots \\
\sum_{k=1}^{10} t_k^{9} & \sum_{k=1}^{10} t_k^{8} & \sum_{k=1}^{10} t_k^{7} & \sum_{k=1}^{10} t_k^{6} & & & \\
 & & & & \ddots & \vdots & \vdots \\
 & \vdots & & & \cdots & \sum_{k=1}^{10} t_k^{2} & \sum_{k=1}^{10} t_k^{1} \\
 & \sum_{k=1}^{10} t_k^{6} \cdots & & & \cdots & \sum_{k=1}^{10} t_k^{1} & \sum_{k=1}^{10} t_k^{0}
\end{bmatrix}
\begin{bmatrix} a_6 \\ a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix}
=
\begin{bmatrix} \sum_{k=1}^{10} y_k t_k^6 \\ \sum_{k=1}^{10} y_k t_k^5 \\ \vdots \\ \vdots \\ \sum_{k=1}^{10} y_k \end{bmatrix}
$$

이를 다음에 대응시켜서 direct method로 주어진 linear system을 풀어보자.

$$Ax=b$$

이제, 이를 풀기 위한 MATLAB 코드는 다음과 같다.

```
A=zeros(7);b=zeros(7,1);
for k=1:10
    t(k)=k*pi/10;
    y(k)=cos(t(k));
end
for i=1:7
    for j=1:7
        ipower=14-i-j;
        for k=1:10
        A(i,j)=A(i,j)+t(k)^ipower;
        end
    end
end
for i=1:7
    for k=1:10
        b(i)=b(i)+t(k)^(7-i)*y(k);
    end
end
saveA=A;saveb=b;
G=zeros(7);count=0;
for i=1:7 %Cholesky decomposition
   G(i,i)=sqrt(A(i,i)-dot(G(i,1:i-1),G(i,1:i-1)));
   for j=(i+1):7
      G(j,i)=(A(j,i) -dot(G(j,1:i-1),G(i,1:i-1)))/G(i,i);
   end
end
for j=1:7 %Solve Gy=b where y=G'x
    b(j)=(b(j)-dot(G(j,1:j-1),b(1:j-1)))/G(j,j); %forward substitution
end
G=G';
for j=7:-1:1 %Solve G'x=y
    b(j)=(b(j)-dot(G(j,j+1:7),b(j+1:7)))/G(j,j); %backward substitution overwriting.
end
for j=1:7 %solution check
if(abs(saveb(j)-dot(saveA(j,1:7),b(1:7)))>5*10^-12)
    count=count+1;
end
end
if(count~=0)
    fprintf('The computed solution seems to be wrong at %f \n',j);
end
disp('The solution x is as follows: ')
disp(b)
range = linspace(0,pi,10);
title('Exercise 6_11','fontsize',12,'fontname','arial');
xlabel('t');
ylabel('solution');
plot(t,y,'-o');
hold on
x=linspace(0,pi,1000);poly=zeros(1000,1);
for i=1:1000
    poly(i,1)=b(1)*x(i)^6+b(2)*x(i)^5+b(3)*x(i)^4+b(4)*x(i)^3+b(5)*x(i)^2+b(6)*x(i)+b(7);
end
plot(x,poly);
hold off
```

이를 실행하면 다음과 같은 계수를 얻는다. 위쪽부터 순서대로 $a_6, a_5, \cdots, a_0$이다.

```
>> Ex6_1
The solution x is as follows:
    0.0002
   -0.0097
    0.0681
   -0.0384
   -0.4700
   -0.0114
    1.0016
```

그림을 통해 비교해보면 다음을 얻고, 따라서 적절하게 fitting되었다고 할 수 있다.