# Assignment 4: Linear Least Squares Problems

Gregory Smetana
ID 1917370
ACM 106a

November 6, 2013

## 1 The Pseudoinverse

Suppose that $A \in \mathbf{R}^{m \times n}$ with $m > n$ and full column rank ($\text{rank}(A) = n$).

### a)

The Frobenius norm is defined as

$$\|M\|_F^2 = \sum_i \sum_j M_{ij}^2 \tag{1}$$

If the columns of $M = [m_1, m_2, ..., m_m]$, the Frobenius norm of M may be expressed as the sum of the vector 2-norms of the columns of M

$$\|M\|_F^2 = \sum_j \|m_j\|_2^2 \tag{2}$$

The Frobenius norm of the matrix product $AX - I$ is

$$\|AX - I\|_F^2 = \sum_j^m \|(AX - I)_j\|_2^2 \tag{3}$$

If the columns of $X = [x_1, x_2, ..., x_m]$, and $e_j$ are the standard basis vectors, this may be written

$$\|AX - I\|_F^2 = \sum_j^m \|(Ax_j - e_j)\|_2^2 \tag{4}$$

All norms have the property $\|X\| \geq 0$, so the $X$ that minimizes $\|AX - I\|_F^2$ will have columns $x_j$ that minimize each term in the sum. As discussed in class, the $x$ that minimizes $\|Ax - b\|_2^2$ for rank deficient $A$ is

$$x = A^\dagger b \tag{5}$$

where the Moore-Penrose generalized inverse inverse is $A^\dagger = (A^TA)^{-1}A^T$. Therefore, the columns of $X$ are the columns of $A^\dagger$, and we have shown that $X = A^\dagger$ minimizes

$$\min_{X\in\mathbf{R}^{n\times m}} \|AX - I\|_F \tag{6}$$

## b)

The SVD of $A$ states

$$A = U\Sigma V^T \tag{7}$$

where

- $U \in \mathbf{R}^{m\times n}$ such that $U^TU = I$
- $V \in \mathbf{R}^{n\times n}$ such that $V^TV = I$
- $\Sigma = \text{Diag}(\sigma_1, \sigma_2, ..., \sigma_n)$ where $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_n \geq 0$.

The corresponding pseudoinverse of $A$ is

$$A^\dagger = V\Sigma^{-1}U^T \tag{8}$$

Expanding the product $(AA^\dagger)A$,

$$\begin{aligned}(AA^\dagger)A &= (U\Sigma V^T)(V\Sigma^{-1}U^T)(U\Sigma V^T)\\ &= U\Sigma\Sigma^{-1}\Sigma V^T\\ &= U\Sigma V^T\\ &= A\end{aligned} \tag{9}$$

$$\boxed{(AA^\dagger)A = A} \tag{10}$$

## c)

Expanding the product $A^T(AA^\dagger)$,

$$\begin{aligned}A^T(AA^\dagger) &= (U\Sigma V^T)^T(U\Sigma V^T)(V\Sigma^{-1}U^T)\\ &= (V\Sigma^TU^T)(U\Sigma V^T)(V\Sigma^{-1}U^T)\\ &= V\Sigma^T\Sigma\Sigma^{-1}U^T\\ &= V\Sigma^TU^T\\ &= (U\Sigma V^T)^T\\ &= A^T\end{aligned} \tag{11}$$

$$\boxed{A^T(AA^\dagger) = A^T} \tag{12}$$

## 2 The underdetermined problem

Suppose that $A \in \mathbf{R}^{m\times n}$ with $m < n$. If $A$ is full row rank (rank($A$) = m), the matrix $A^T \in \mathbf{R}^{n\times m}$ has full column rank. Using the SVD of $A^T$,

$$A^T = U\Sigma V^T \tag{13}$$

where

- $U \in \mathbf{R}^{n\times m}$ such that $U^TU = I$
- $V \in \mathbf{R}^{m\times m}$ such that $V^TV = I$
- $\Sigma = \text{Diag}(\sigma_1, \sigma_2, ..., \sigma_m)$ where $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_m \geq 0$.

The matrix $A$ may be expressed

$$A = V\Sigma U^T \tag{14}$$

Any solution $x$ to the linear system $Ax = b$ may be written in the form $x = x_1 + x_2$, where $x_1 \in \text{Col}(A^T)$ and $x_2 \in \text{Null}(A)$

$$\|Ax - b\|_2^2 = \|V\Sigma U^T(x_1 + x_2) - b\|_2^2 \tag{15}$$

The columns of $U$ are a set of basis vectors for the range of $A^T$, so the vector $x_2$ is orthogonal to the columns of $U$

$$\|Ax - b\|_2^2 = \|V\Sigma U^T x_1 - b\|_2^2 \tag{16}$$

Choosing the minimum norm solution from this subspace,

$$\begin{aligned} x &= U\Sigma^{-1}V^Tb \\ &= U\Sigma\Sigma^{-1}\Sigma^{-1}V^Tb \\ &= U\Sigma V^TV^{-T}\Sigma^{-1}UU^T\Sigma^{-1}V^Tb \\ &= U\Sigma V^T(V\Sigma U^TU\Sigma V^T)^{-1}b \\ &= A^T(AA^T)^{-1}b \end{aligned} \tag{17}$$

So we have shown the minimum 2-norm solution of $Ax = b$ is equal to

$$\boxed{x = A^\dagger b} \tag{18}$$

where $A^\dagger = A^T(AA^T)^{-1}$

## 3 Polynomial approximation

Polynomials of order $m$ may be fit to functions sampled with values $b_i$ at points $t_i$ by solving the least squares problem

$$x = \arg \min_{x \in \mathbf{R}^m} \|Ax - b\|_2 \tag{19}$$

where $A \in \mathbf{R}^{n \times m}$ is the matrix with $(i,j)th$ entry equal to $A_{i,j} = t_i^{j-1}$

### a)

The normal equations $A^T Ax = A^T b$ were solved using a Cholesky factorization in the first part of this problem.

### b)

A Householder-based QR factorization of $A$ was used to solve the least squares problem in the second part. A version of the algorithm was used that avoids any explicit computation of the projection matrices. In both algorithms, triangular systems were solved using the Matlab backslash command. The code is attached as an Appendix.

The function $f(t) = \sin(4t)$ and was sampled at $n = 51$ points on the interval [0,1]. The function and approximating polynomials of each method for values $m = 3, 7, 12$ are plotted in Figures 1-3. The results for the solution coefficient vector $x$, the norm of the residual, and time required to obtain the approximation for each method for values $m = 3, 7, 12$ are summarized in Tables 1-3.

| | Cholesky factorization | QR factorization |
|---|---|---|
| solution vector $x$ | 5.773287988489395e-02<br>4.281921761790140e+00<br>-5.349578523794193e+00 | 5.773287988488140e-02<br>4.281921761790207e+00<br>-5.349578523794256e+00 |
| $\|Ax - b\|_2$ | 5.811882658961149e-01 | 5.811882658961154e-01 |
| Time required [$\mu$s] | 810 | 1516 |
| Condition number | $\kappa_2(A^T A)$ = 4.867618438401785e+02 | $\kappa_2(A)$ = 2.206267988799603e+01 |

Table 1: Comparison of Cholesky and QR methods for $m = 3$

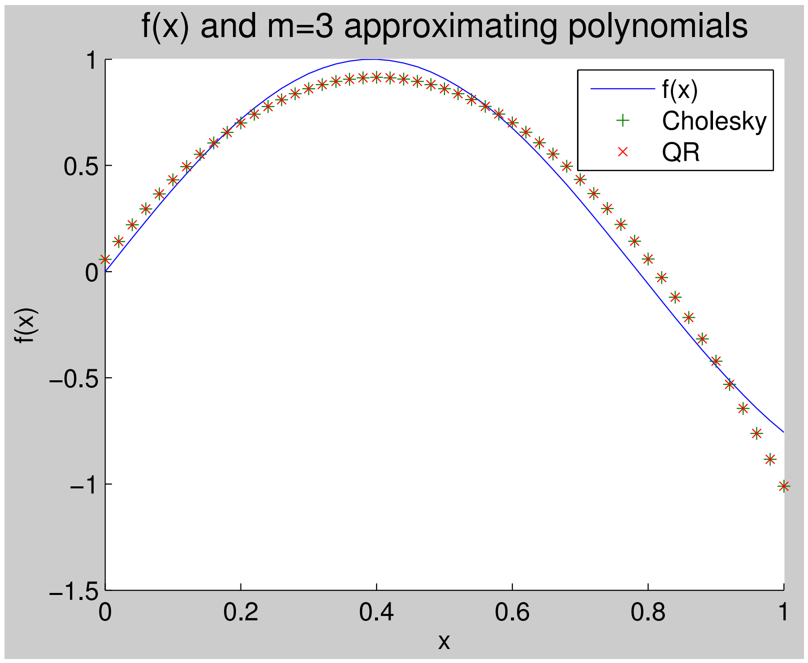

Figure 1

| | Cholesky factorization | QR factorization |
|---|---|---|
| solution vector $x$ | 1.656380936174698e-04<br>3.989420649671397e+00<br>1.229658456163471e-01<br>-1.112693941751450e+01<br>2.760391884291479e-01<br>1.048354404629816e+01<br>-4.502277252395971e+00 | 1.656380962030799e-04<br>3.989420649508054e+00<br>1.229658474860089e-01<br>-1.112693942551486e+01<br>2.760392040232638e-01<br>1.048354403224540e+01<br>-4.502277247641723e+00 |
| $\|Ax - b\|_2$ | 7.468336144166766e-04 | 7.468336144165733e-04 |
| Time required [$\mu$s] | 127 | 580 |
| Condition number | $\kappa_2(A^TA)$ = 3.915604824294044e+08 | $\kappa_2(A)$ = 1.978788728823597e+04 |

Table 2: Comparison of Cholesky and QR methods for $m = 7$

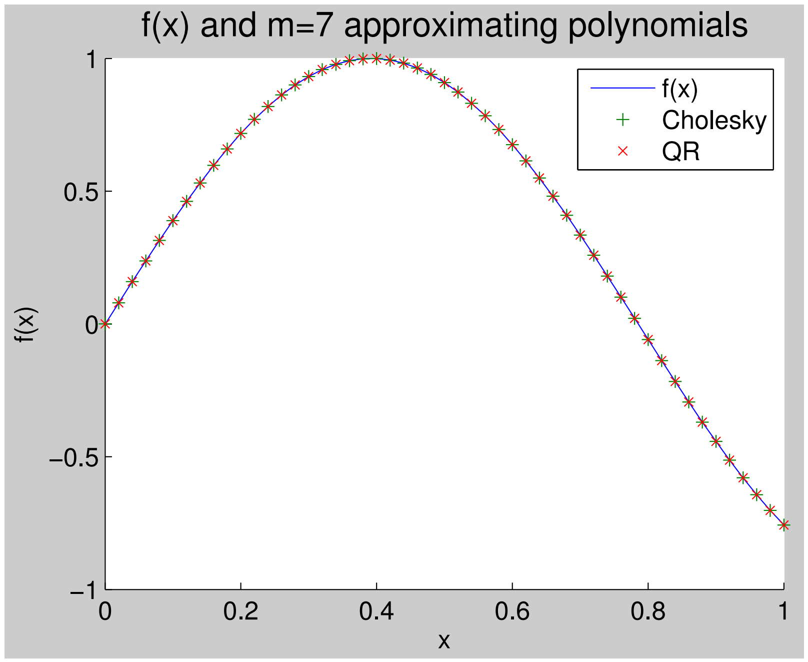

Figure 2

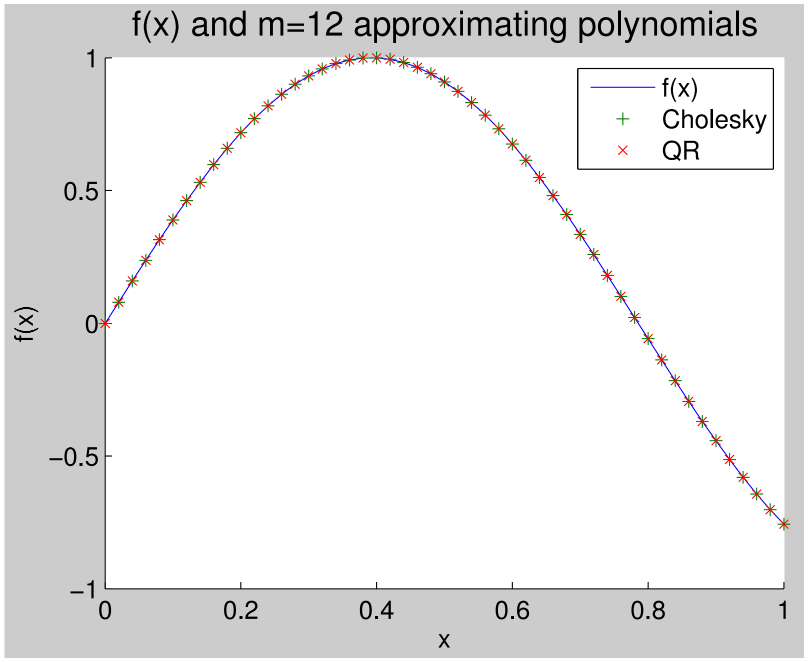

Figure 3

| | Cholesky factorization | QR factorization |
|---|---|---|
| solution vector $x$ | -6.671131127896551e-09 | -2.665211567614457e-09 |
| | 4.000002325585443e+00 | 4.000001205956912e+00 |
| | -9.800632713093781e-05 | -5.645186915616958e-05 |
| | -1.066503428065786e+01 | -1.066564308017487e+01 |
| | -1.430283744924931e-02 | -9.655654437952179e-03 |
| | 8.608393589186797e+00 | 8.587449519298678e+00 |
| | -2.525855838422283e-01 | -1.932254237565274e-01 |
| | -2.686800494913550e+00 | -2.795535730790732e+00 |
| | -8.430701648396042e-01 | -7.145141657078831e-01 |
| | 1.550721728828908e+00 | 1.456008851886011e+00 |
| | -5.014643416298591e-01 | -4.619239047002904e-01 |
| | 4.743557764659213e-02 | 4.029233913304647e-02 |
| $\|Ax - b\|_2$ | 2.448105425893064e-08 | 1.743034423482307e-08 |
| Time required [$\mu$s] | 288 | 2596 |
| Condition number | $\kappa_2(A^T A)$ = 1.280614687904160e+16 | $\kappa_2(A)$ = 1.171893555129648e+08 |

Table 3: Comparison of Cholesky and QR methods for $m = 12$

Visually, the quality of the approximation of both methods for $m > 3$ appears to be very good. The values of the residual and coefficients at $m = 3$ and $m = 7$ are similar to at least 10 digits for both methods, but the residual for the Cholesky factorization is higher than the QR factorization at $m = 12$. The product $A^T A$ squares the condition number. Therefore, the Cholesky factorization solving the normal equations is less well conditioned than the QR factorization of $A$ and is not well suited for large values of $m$.

As discussed in class, the QR decomposition requires $2n^2m - 2/3n^3 + O(m^2) + O(n^2)$ operations. The cost of computing the product $A^T A$ and performing the Cholesky factorization is $m^2n + 1/3m^3 + O(m^2)$. This analysis leads us to believe that the time required by the Cholesky method should increase more rapidly with $m$ than the QR method, but the times required for the Cholesky and QR factorization for different values of $m$ do not show any clear trends. This is likely because the the cost of memory operations is relatively large, and the time will be sensitive to background programs running. An average of many trials may give better results.

## A Smetana_Gregory_1917370_A4_P3_DIARY.txt

```
run('Smetana_Gregory_1917370_A4_P3.m')

x_a =

     5.773287988489395e-02
     4.281921761790140e+00
    -5.349578523794193e+00


t_a =

   810


r_a =

     5.811882658961149e-01


cond_a =

     4.867618438401785e+02


x_b =

     5.773287988488140e-02
     4.281921761790207e+00
    -5.349578523794256e+00


t_b =

        1516


r_b =

     5.811882658961154e-01


cond_b =

     2.206267988799603e+01
```

```
x_a =

     1.656380936174698e-04
     3.989420649671397e+00
     1.229658456163471e-01
    -1.112693941751450e+01
     2.760391884291479e-01
     1.048354404629816e+01
    -4.502277252395971e+00


t_a =

   127


r_a =

     7.468336144166766e-04


cond_a =

     3.915604824294044e+08


x_b =

     1.656380962030799e-04
     3.989420649508054e+00
     1.229658474860089e-01
    -1.112693942551486e+01
     2.760392040232638e-01
     1.048354403224540e+01
    -4.502277247641723e+00


t_b =

   580


r_b =

     7.468336144165733e-04


cond_b =

     1.978788728823597e+04
```

```
x_a =

    -6.671131127896551e-09
     4.000002325585443e+00
    -9.800632713093781e-05
    -1.066503428065786e+01
    -1.430283744924931e-02
     8.608393589186797e+00
    -2.525855838422283e-01
    -2.686800494913550e+00
    -8.430701648396042e-01
     1.550721728828908e+00
    -5.014643416298591e-01
     4.743557764659213e-02


t_a =

   288


r_a =

     2.448105425893064e-08


cond_a =

     1.280614687904160e+16


x_b =

    -2.665211567614457e-09
     4.000001205956912e+00
    -5.645186915616958e-05
    -1.066564308017487e+01
    -9.655654437952179e-03
     8.587449519298678e+00
    -1.932254237565274e-01
    -2.795535730790732e+00
    -7.145141657078831e-01
     1.456008851886011e+00
    -4.619239047002904e-01
     4.029233913304647e-02


t_b =
        2596


r_b =

     1.743034423482307e-08


cond_b =

     1.171893555129648e+08

diary off
```

## B Smetana_Gregory_1917370_A4_P3.m

```
%Smetana_Gregory_1917370_A4_P3
clear;
clc;
close all;
path(path,'export_fig/');
%solve normal equations A'*A = A'*b
n=51;
t = linspace(0, 1, n)';
b = sin(4 *t);
format longE
for m= [3,7,12]

A = zeros(n,m);

for(j = 1:m)
    A(:,j)= t.^(j-1);
end

%% part a

% factorize R' * R = A
tic;
R = chol(A'*A);
d = A'*b;
z = R'\d; % forward substitution
x_a = R\z % back substitution
t_a = toc * 1E6
r_a = norm(b - polyval(flipud(x_a),t))
cond_a = cond(A'*A)
%% part b
tic;
```

```
[U,R] = householder(A);
Qb = getQb(b, U);
x_b = R\Qb % back substitution
t_b = toc * 1E6
r_b = norm(b - polyval(flipud(x_b),t))
cond_b = cond(A)

%% plot
figure;
hold all;
plot(t,b)
plot(t,polyval(flipud(x_a),t),'+');
plot(t,polyval(flipud(x_b),t),'x');
xlabel('x','FontSize',12);
ylabel('f(x)','FontSize',12);
title(['f(x) and m=',num2str(m),' approximating polynomials'],'FontSize',16);
legend('f(x)', 'Cholesky', 'QR');
set(gca,'FontSize',12);
filename = ['report/p3_m=',num2str(m),'.pdf'];
export_fig(filename)
end;
```

## C householder.m

```
%Smetana_Gregory_1917370_A4_P3
function [U,R] = householder( A )
%HOUSEHOLDER performs triangularization of A
[m, n]= size(A);
U = zeros(m,n);
R = A;
for i = 1:min(m-1,n)
    % Extract the i-th column of A(i : m; i : n)
    x = R(i:m,i);
    e = zeros(length(x),1);
    e(1) = 1;
    % Compute direction ui
    u = sign(x(1))*norm(x)*e + x;
    % Normalize ui
    u = u./ norm(u);
    % Update A(i : m; i : n):
    R(i:m, i:n) = R(i:m,i:n)- 2*u*u'*R(i:m,i:n);
    U(i:m,i)= u;
end
% force triangular format
R = triu(R);
```

## D getQb.m

```
%Smetana_Gregory_1917370_A4_P3
function [ Qb ] = getQb( b, U )
%GETQB Computes the product Q'b
[m,n] = size(U);
for i=1:n
    u = U(i:m,i);
    g = -2 * u' * b(i:m);
    b(i:m) = b(i:m) + g*u;
end
Qb = b;
```