Least Squares and QR Decomposition

L. Eldén

LiU

September 2020

Least Squares - Example Elasticity

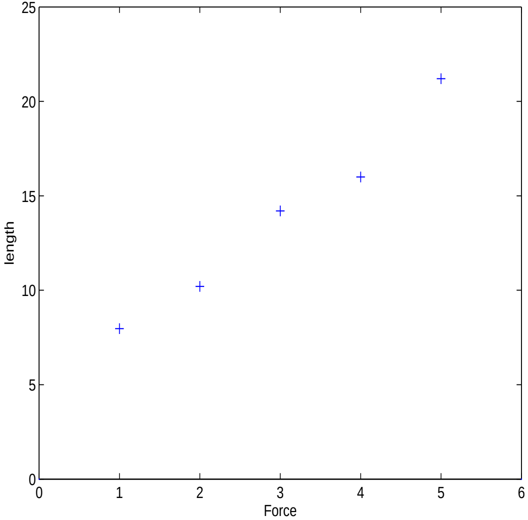
Hooke's law:

$$e + \kappa F = l,$$

where κ is the elasticity constant.
Data:

F	1	2	3	4	5
l	7.97	10.2	14.2	16.0	21.2

Data



Overdetermined System

$$\begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{pmatrix} \begin{pmatrix} e \\ \kappa \end{pmatrix} = \begin{pmatrix} 7.97 \\ 10.2 \\ 14.2 \\ 16.0 \\ 21.2 \end{pmatrix}.$$

Determine an approximation of the elasticity constant of the spring using the least squares method.

Overdetermined System

Let $A \in \mathbb{R}^{m \times n}$, $m > n$. The system

$$Ax = b$$

More equations than unknowns!
In general no solution exists.

($m = 3$ and $n = 2$) We want to find a linear combination of the vectors such that

$$x_1 a_{\cdot 1} + x_2 a_{\cdot 2} = b.$$

The Least Squares Method

Make the vector $r = b - x_1 a_{\cdot 1} - x_2 a_{\cdot 2} = b - Ax$ as small as possible.
$b - Ax$ is called the residual vector
The least squares method: Euclidean distance.
Solve the minimization problem

$$\min_x \|b - Ax\|_2. \qquad (1)\text{span}$$

Make r orthogonal to the columns of A.

Normal Equations

Make r orthogonal to the columns of A.
General case:

$$r^T a_{\cdot j} = 0, \qquad j = 1, 2, \dots, n.$$

Equivalently,

$$r^T \begin{pmatrix} a_{\cdot 1} & a_{\cdot 2} & \cdots & a_{\cdot n} \end{pmatrix} = r^T A = 0.$$

Normal equations:

$$A^T A x = A^T b,$$

for the determining the coefficients in x.

Theorem

If the columns vectors of A are linearly independent, then the normal equations

$$A^T A x = A^T b.$$

are non-singular and have a unique solution.

The Example

Matlab:

```
>> C=A'*A            % Normal equations

C =   5     15
     15     55

>> x=C\(A'*b)

x = 4.2360
    3.2260
```

Drawbacks of Normal Equations

1. Forming $A^T A$ leads to loss of information.
2. The condition number $A^T A$ is the square of that of A:

$$\kappa(A^T A) = (\kappa(A))^2.$$

Condition numbers

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \end{pmatrix}$$

```
cond(A) = 8.3657
cond(A'*A) = 69.9857
```

Drawbacks of Normal Equations

Worse example:

$$A = \begin{pmatrix} 1 & 101 \\ 1 & 102 \\ 1 & 103 \\ 1 & 104 \\ 1 & 105 \end{pmatrix}$$

```
cond(A) = 7.5038e+03
cond(A'*A) = 5.6307e+07
```

A linear model

$$l(x) = c_0 + c_1 x,$$

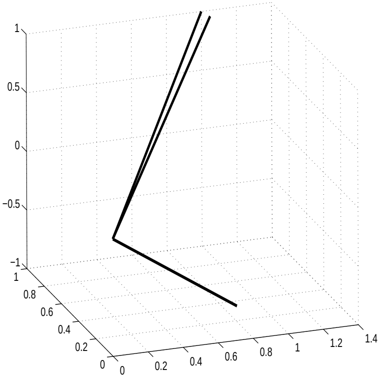
with data vector $x = (101\,102\,103\,104\,105)^T$, should be replaced by the model

$$l(x) = b_0 + b_1(x - 103),$$

In the latter case the normal equations become diagonal and much better conditioned.

Orthogonality

$$A = \begin{pmatrix} 1 & 1.05 \\ 1 & 1 \\ 1 & 0.95 \end{pmatrix}, \qquad B = \begin{pmatrix} 1 & 1/\sqrt{2} \\ 1 & 1 \\ 1 & -1/\sqrt{2} \end{pmatrix},$$



Plane rotations

$$G = \begin{pmatrix} c & s \\ -s & c \end{pmatrix}, \qquad c^2 + s^2 = 1,$$

A plane rotation can be used to zero the second element of a vector x by choosing $c = x_1/\sqrt{x_1^2 + x_2^2}$ and $s = x_2/\sqrt{x_1^2 + x_2^2}$:

$$\frac{1}{\sqrt{x_1^2 + x_2^2}} \begin{pmatrix} x_1 & x_2 \\ -x_2 & x_1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \sqrt{x_1^2 + x_2^2} \\ 0 \end{pmatrix}.$$

Embed a two-dimensional rotation in a larger unit matrix:

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c & 0 & s \\ 0 & 0 & 1 & 0 \\ 0 & -s & 0 & c \end{pmatrix}$$

We can choose c and s so that we zero element 4 in a vector $x \in \mathbb{R}^4$ *by a rotation in plane* (2, 4).

Plane rotations in MATLAB

```
function [c,s]=rot(x,y);
 % Construct a plane rotation that zeros the second
 % component in the vector [x;y]' (x and y are scalars)
 sq=sqrt(x^2 + y^2);
 c=x/sq; s=y/sq;

function [X]=approt(c,s,i,j,X);
 % Apply a plane (Givens) rotation in plane (i,j)
 % to a matrix X
 X([i,j],:)=[c s; -s c]*X([i,j],:);
```

Zero all the elements in a vector

Given $x \in \mathbb{R}^4$, transform it to κe_1. First zero the last element:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & c_1 & s_1 \\ 0 & 0 & -s_1 & c_1 \end{pmatrix} \begin{pmatrix} \times \\ \times \\ \times \\ \times \end{pmatrix} = \begin{pmatrix} \times \\ \times \\ * \\ 0 \end{pmatrix}.$$

Then, zero the element in position 3:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c_2 & s_2 & 0 \\ 0 & -s_2 & c_2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \times \\ \times \\ \times \\ 0 \end{pmatrix} = \begin{pmatrix} \times \\ * \\ 0 \\ 0 \end{pmatrix}.$$

Finally, the second element is annihilated:

$$\begin{pmatrix} c_3 & s_3 & 0 & 0 \\ -s_3 & c_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \times \\ \times \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \kappa \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Plane rotations

Reduce the vector x to a multiple of the standard basis vector e_1:

```
x=[1;2;3;4];
for i=3:-1:1
  [c,s]=rot(x(i),x(i+1));
  x=approt(c,s,i,i+1,x);
end

>> x = 5.47722557505166
      -0.00000000000000
      -0.00000000000000
      -0.00000000000000
```

After the reduction the first component of x is equal to $\| x \|_2$.

Householder Transformations

Householder transformation:

$$P = I - 2uu^T, \qquad \|u\|_2 = 1.$$

P is symmetric and orthogonal
Multiplication by P:

$$Px = x - (2u^T x)u$$

Householder transformations in MATLAB I

```
function u=househ(x)
  % Compute the Householder vector u such that
  % (I - 2 u * u')x = k*e_1, where
  % |k| is equal to the euclidean norm of x
  % and e_1 is the first unit vector
  n=length(x);       % Number of components in x
  kap=norm(x); v=zeros(n,1);
  v(1)=x(1)+sign(x(1))*kap;
  v(2:n)=x(2:n);
  u=(1/norm(v))*v;

function Y=apphouse(u,X);
  % Multiply the matrix X by a Householder matrix
  % Y = (I - 2 * u * u') * X
  Y=X-2*u*(u'*X);
```

Householder transformations in MATLAB II

Zero the first three elements of the vector $x = (1, 2, 3, 4)^T$:

```
>>    x=[1; 2; 3; 4];
>>    u=househ(x);
>>    y=apphouse(u,x)

    y = -5.4772
             0
             0
             0
```

```
u=househ(A(2:m,2)); A(2:m,2:n)=apphouse(u,A(2:m,2:n));

>> A = -0.8992   -0.6708   -0.7788   -0.9400
      -0.0000    0.3299    0.7400    0.3891
      -0.0000    0.0000   -0.1422   -0.6159
      -0.0000   -0.0000    0.7576    0.1632
      -0.0000   -0.0000    0.3053    0.4680
```

Theorem (QR decomposition)

Any matrix $A \in \mathbb{R}^{m \times n}$, $m \geq n$, can be transformed to upper triangular form by an orthogonal matrix. The transformation is equivalent to a decomposition

$$A = Q \begin{pmatrix} R \\ 0 \end{pmatrix},$$

where $Q \in \mathbb{R}^{m \times m}$ is orthogonal and $R \in \mathbb{R}^{n \times n}$ is upper triangular. If the columns of A are linearly independent, then R is nonsingular.

QR decomposition and least squares

QR decomposition:

$$A = Q \begin{pmatrix} R \\ 0 \end{pmatrix} = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R \\ 0 \end{pmatrix} = Q_1 R \qquad \text{(Thin QR)}$$

Least squares problem:

$$\begin{aligned} & \|Ax - b\|^2 = \|Q_1 Rx - b\|^2 = \|Q^T (Q_1 Rx - b)\|^2 \\ = & \left\| \begin{pmatrix} Q_1^T \\ Q_2^T \end{pmatrix} (Q_1 Rx - b) \right\|^2 = \left\| \begin{pmatrix} Rx - Q_1^T b \\ 0 - Q_2^b \end{pmatrix} \right\|^2 \\ = & \|Rx - Q_1^T b\|^2 + \|Q_2^T\|^2 \end{aligned}$$

Least squares solution: $x = R^{-1} Q_1^T b$

QR decomposition

```
A =   1      1      1
      1      2      4
      1      3      9
      1      4     16
 >> [Q,R]=qr(A)

Q =-0.5000    0.6708    0.5000    0.2236
   -0.5000    0.2236   -0.5000   -0.6708
   -0.5000   -0.2236   -0.5000    0.6708
   -0.5000   -0.6708    0.5000   -0.2236

R =-2.0000   -5.0000  -15.0000
         0   -2.2361  -11.1803
         0         0    2.0000
         0         0         0
```

Thin QR

```
>> [Q,R]=qr(A,0)

Q =-0.5000    0.6708    0.5000
   -0.5000    0.2236   -0.5000
   -0.5000   -0.2236   -0.5000
   -0.5000   -0.6708    0.5000

R =-2.0000   -5.0000  -15.0000
         0   -2.2361  -11.1803
         0         0    2.0000
```

```
A =   1     1              b = 7.9700
      1     2                 10.2000
      1     3                 14.2000
      1     4                 16.0000
      1     5                 21.2000
```

Thin QR and least squares solution

```
>> [Q1,R]=qr(A,0)       % thin QR

Q1 = -0.4472   -0.6325
     -0.4472   -0.3162
     -0.4472    0.0000
     -0.4472    0.3162
     -0.4472    0.6325
```

Least squares problem II

```
R = -2.2361   -6.7082
         0    3.1623

>> x=R\(Q1'*b)

x = 4.2360
    3.2260
```

Least squares problem in R I

```
> A<-matrix(c(1,1,1,1,1,1,2,3,4,5),nrow=5)
> A
     [,1] [,2]
[1,]    1    1
[2,]    1    2
[3,]    1    3
[4,]    1    4
[5,]    1    5
> b=c(7.97,10.2,14.2,16.0,21.2)
> b
[1]  7.97 10.20 14.20 16.00 21.20

> x=qr.solve(A,b)
> x
[1] 4.236 3.226
```

Least squares problem in R II

```
> A.qr
$qr
          [,1]       [,2]
[1,] -2.2360680 -6.7082039
[2,]  0.4472136  3.1622777
[3,]  0.4472136 -0.1954395
[4,]  0.4472136 -0.5116673
[5,]  0.4472136 -0.8278950

$rank
[1] 2

$qraux
[1] 1.447214 1.120788
```

Least squares problem in R III

```
$pivot
[1] 1 2

attr(,"class")
[1] "qr"


> Q=qr.Q(A.qr)
> Q
           [,1]          [,2]
[1,] -0.4472136 -6.324555e-01
[2,] -0.4472136 -3.162278e-01
[3,] -0.4472136  1.179070e-17
[4,] -0.4472136  3.162278e-01
```

Least squares problem in R IV

```
[5,] -0.4472136  6.324555e-01

> R=qr.R(A.qr)
> R
          [,1]      [,2]
[1,] -2.236068 -6.708204
[2,]  0.000000  3.162278
> q()
```

Ill-conditioned example

Let $\epsilon = 10^{-7}$, and define

$$A = \begin{pmatrix} 1 & 1 \\ \epsilon & 0 \\ 0 & \epsilon \end{pmatrix}$$

The condition number of A is of the order 10^7.

```
x=[1;1]; b=A*x;
xq=A\b;                % QR decomposition
xn=(A'*A)\(A'*b);      % Normal equations
xq  =  1.00000000000000   xn = 1.01123595505618
       1.00000000000000        0.98876404494382
```