NLA 2021-2022
Least squares problem

Martin Sombra

15 October 2021

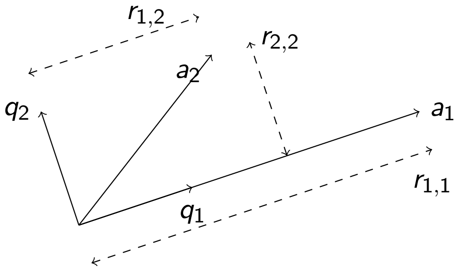
LSP and orthogonalization

The columns of A are independent ($\operatorname{rank}(A) = n$) but not orthogonal

If this were the case, $x_{\min}$ would be easy to find!

Gram-Schmidt orthogonalization

Set $a_j = \mathrm{col}_j(A) \in \mathbb{R}^m$, $j = 1, \ldots, n$



GS produces orthonormal m-vectors q_j, $j = 1, \ldots, n$, such that

$$\mathrm{span}(q_1, \ldots, q_j) = \mathrm{span}(a_1, \ldots, a_j), \quad j = 1, \ldots, n$$

Gram-Schmidt orthogonalization (cont.)

First step:

$$q_1 \leftarrow \frac{a_1}{\|a_1\|_2} \quad \text{(normalize)}$$

GO step:

$$\tilde{a}_2 \leftarrow a_2 - \langle a_2, q_1 \rangle q_1 \quad \text{(orthogonalize)}$$

$$q_2 \leftarrow \frac{\tilde{a}_2}{\|\tilde{a}_2\|_2} \quad \text{(normalize)}$$

GO step:

$$\tilde{a}_3 \leftarrow a_3 - \langle a_3, q_1 \rangle q_1 - \langle a_3, q_2 \rangle q_2$$

$$q_3 \leftarrow \frac{\tilde{a}_3}{\|\tilde{a}_3\|_2}$$

The QR factorization

We can write the a_j's in terms of the q_j's:

$$\begin{aligned}
a_1 &= \|\tilde{a}_1\|_2\, q_1 \\
a_2 &= \langle a_2, q_1 \rangle\, q_1 + \|\tilde{a}_2\|_2\, q_2 \\
a_3 &= \langle a_3, q_1 \rangle\, q_1 + \langle a_3, q_2 \rangle\, q_2 + \|\tilde{a}_3\|_2\, q_3 \\
&\dots
\end{aligned}$$

$$\begin{bmatrix} a_1 & a_2 & a_3 & \cdots \end{bmatrix} = \begin{bmatrix} q_1 & q_2 & q_3 & \cdots \end{bmatrix} \begin{bmatrix} r_{1,1} & r_{1,2} & r_{1,3} & \cdots \\ 0 & r_{2,2} & r_{2,3} & \cdots \\ 0 & 0 & r_{3,3} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

that is

$$\boxed{A = Q\,R}$$

with

- Q orthogonal $m \times n$
- R upper triangular $n \times n$ matrix with positive diagonal entries

This factorization is unique!

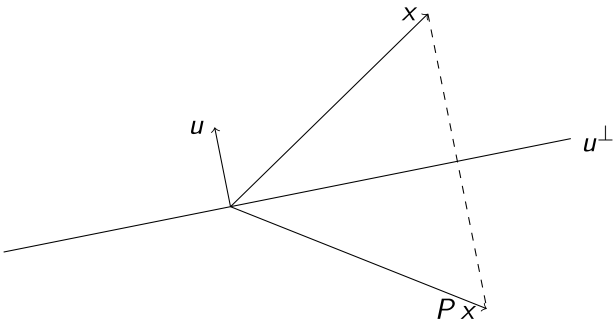
Solving the LSP with the QR factorization

The QR factorization solves the normal equations, and so the LSP:

$$
\begin{aligned}
x_{\min} &= (A^T A)^{-1} A^T b \\
&= ((QR)^T Q\, R)^{-1} (Q\, R)^T b \\
&= (R^T R)^{-1} R^T Q\, b \\
&= R^{-1} Q^T\ b
\end{aligned}
$$

- The GS algorithm is not stable when the columns of A are close to rank deficient

Householder reflections



A *Householder reflection* is

$$P = \mathbb{1}_m - 2\,u\,u^T \in \mathbb{R}^{m\times m}$$

for a unit vector $u \in \mathbb{R}^m$

It is symmetric ($P^T = P$) and orthogonal ($P^T P = \mathbb{1}_m$)

Householder reflections (cont.)

Given $y \in \mathbb{R}^m$ there is a reflection that zeroes all but the first entry:

$$P\,y = \begin{bmatrix} c \\ 0 \\ \vdots \\ 0 \end{bmatrix} = c\,e_1 \in \mathbb{R}^m$$

Since P is orthogonal

$$|c| = \|P\,y\|_2 = \|y\|_2$$

Householder reflections (cont.)

To compute u:

$$P\, y = (\mathbb{1}_m - 2\, u\, u^T)\, y = y - 2\langle u, y\rangle\, u = \pm\|y\|_2\, e_1$$

thus

$$\boxed{2\langle u, y\rangle\, u = y \pm \|y\|_2\, e_1}$$

Choose the sign so to avoid cancellations: u is a scalar multiple of

$$\widetilde{u} = y \pm \|y\|_2\, e_1 = \begin{bmatrix} y_1 + \mathrm{sign}(y_1)\, \|y\|_2 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$$

and so we set

$$u = \mathrm{House}(y) := \frac{\widetilde{u}}{\|\widetilde{u}\|_2}$$

QR factorization with Householder reflections

Set $m = 4$ and $n = 3$:

$$A = \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}$$

- Choose P_1 such that

$$A_1 \leftarrow P_1 A = \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix}$$

QR factorization with Householder reflections (cont.)

- Choose $P_2 = \begin{array}{cc} & \begin{array}{cc} 1 & 2 \end{array} \\ \begin{array}{c} 1 \\ 3 \end{array} & \begin{bmatrix} 1 & \mathbb{0} \\ \mathbb{0} & P_2' \end{bmatrix} \end{array}$ such that

$$A_2 \leftarrow P_2 \, A_1 = \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & \times \end{bmatrix}$$

- Choose $P_3 = \begin{array}{cc} \begin{array}{c} 1 \\ 1 \\ 2 \end{array} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \mathbb{0} & \mathbb{0} & P_3' \end{bmatrix} \end{array}$ such that

$$A_3 \leftarrow P_3 \, A_2 = \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & 0 \end{bmatrix}$$

QR factorization with Householder reflections (cont.)

$P_3 \, P_2 \, P_1 \, A = \widetilde{R} (= A_4)$ is *upper triangular*. Hence

$$\boxed{A = P_1^T \, P_2^T \, P_3^T \, \widetilde{R} = Q \, R}$$

with

- Q the first three columns of $P_1^T \, P_2^T \, P_3^T$
- R the first three rows of $\widetilde{R}$

Example

Let

$$A = \begin{bmatrix} 1 & -3 \\ 0 & 2 \\ -1 & -1 \end{bmatrix}$$

Set

$$\widetilde{u}_1 = \begin{bmatrix} 1 + 2^{1/2} \\ 0 \\ -1 \end{bmatrix} \quad \text{and} \quad u_1 = \text{House} \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} = \frac{\widetilde{u}_1}{\|\widetilde{u}_1\|_2} = \begin{bmatrix} 0.92 \\ 0 \\ -0.38 \end{bmatrix}$$

Then

$$P_1 = \mathbb{1}_3 - 2\, u_1\, u_1^T = \begin{bmatrix} -0.71 & 0 & 0.71 \\ 0 & 1 & 0 \\ 0.71 & 0 & 0.71 \end{bmatrix}, \quad A_1 = P_1\, A = \begin{bmatrix} -1.41 & 1.41 \\ 0 & 2 \\ 0 & -2.83 \end{bmatrix}$$

Example (cont.)

Set then

$$\widetilde{u}_2 = \begin{bmatrix} 2 + (2^2 + (-2.83)^2)^{1/2} \\ -2.83 \end{bmatrix} \quad \text{and} \quad u_2 = \text{House}\begin{bmatrix} 2 \\ -2.83 \end{bmatrix} = \frac{\widetilde{u}_2}{\|\widetilde{u}_2\|_2} = \begin{bmatrix} 0.89 \\ -0.46 \end{bmatrix}$$

Then

$$P_2 = \begin{bmatrix} 1 & \mathbb{0} \\ \mathbb{0} & \mathbb{1}_2 - 2\, u_2\, u_2^T \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -0.58 & 0.82 \\ 0 & 0.82 & 0.58 \end{bmatrix}$$

$$P_2\, A_1 = \begin{bmatrix} -1.41 & 1.41 \\ 0 & -3.49 \\ 0 & 0 \end{bmatrix} = \widetilde{R}$$

Example (cont.)

Finally

$$
\begin{aligned}
A = P_1^T P_2^T \widetilde{R} = \widetilde{Q}\,\widetilde{R} &= \begin{bmatrix} -0.71 & 0.58 & 0.41 \\ 0 & -0.58 & 0.82 \\ 0.71 & 0.58 & 0.41 \end{bmatrix} \begin{bmatrix} -1.41 & 1.41 \\ 0 & -3.49 \\ 0 & 0 \end{bmatrix} \\
&= Q\,R = \begin{bmatrix} -0.71 & 0.58 \\ 0 & -0.58 \\ 0.71 & 0.58 \end{bmatrix} \begin{bmatrix} -1.41 & 1.41 \\ 0 & -3.49 \end{bmatrix}
\end{aligned}
$$

The algorithm

for $i = 1$ to $\min(m-1, n)$

$u_i \leftarrow \text{House}(A(i:m, i))$

$P_i \leftarrow \mathbb{1}_{m-i+1} - 2\, u_i\, u_i^T$

$A_i(i:m, i:n) \leftarrow P_i'\, A(i:m, i:n)$

- we do not really need P_i' but just the multiplication

$$(\mathbb{1}_{m-i+1} - 2\, u_i\, u_i^T)\, A(i:m, i:n) = A(i:m, i:n) - 2\, u_i\, (u_i^T A(i:m, i:n))$$

- each P_i can be “stored” as u_i
- Q can be stored as $P_1 \cdots P_{n-1}$

The algorithm (cont.)

The complexity of this algorithm is

$$2\,n^2\,m - \frac{2}{3}\,n^2 \text{ flops}$$

Compared with solving the normal equations via Cholesky's algorithm:

- twice the complexity (for $m \gg n$)
- more numerically stable

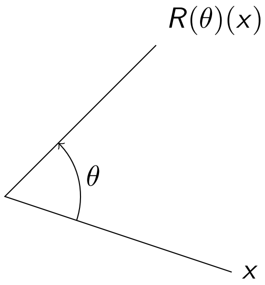
Plane rotations

A *rotation* on the plane with angle θ is the linear map

$$R(\theta)\colon \mathbb{R}^2 \to \mathbb{R}^2$$

given by the orthogonal matrix

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$



Plane rotations

A *Givens rotation*: matrix of a rotation on the (i, j)-plane of $\mathbb{R}^m$

$$R(i,j,\theta) = \begin{array}{cc} & \begin{array}{ccccc} \phantom{\mathbb{1}_{i-1}} & i & \phantom{\mathbb{1}_{j-i-1}} & j & \phantom{\mathbb{1}_{n-j-1}} \end{array} \\ \begin{array}{c} \\ i \\ \\ j \\ \\ \end{array} & \begin{bmatrix} \mathbb{1}_{i-1} & & & & \\ & \cos(\theta) & & -\sin(\theta) & \\ & & \mathbb{1}_{j-i-1} & & \\ & \sin(\theta) & & \cos(\theta) & \\ & & & & \mathbb{1}_{n-j-1} \end{bmatrix} \end{array}$$

QR factorization with Givens rotations

The QR factorization can be computed with Givens rotations similarly as with Householder reflections:

Given $x \in \mathbb{R}^m$ and $i > j$, we can zero x_j by choosing θ such that

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x_i \\ x_j \end{bmatrix} = \begin{bmatrix} (x_i^2 + x_j^2)^{1/2} \\ 0 \end{bmatrix}$$

or equivalently

$$\cos(\theta) = \frac{x_i}{(x_i^2 + x_j^2)^{1/2}} \quad \text{and} \quad \sin(\theta) = \frac{-x_j}{(x_i^2 + x_j^2)^{1/2}}$$

$$\boxed{\text{inverse trigonometric functions are not needed!}}$$

Example

Let

$$A = \begin{bmatrix} 1 & -3 \\ 0 & 2 \\ -1 & -1 \end{bmatrix}$$

Setting

$$R_1 = \begin{bmatrix} 0.71 & 0 & -0.71 \\ 0 & 1 & 0 \\ 0.71 & 0 & 0.71 \end{bmatrix}$$

then

$$A_1 = R_1\, A = \begin{bmatrix} 1.41 & -1.41 \\ 0 & 2 \\ 0 & -2.82 \end{bmatrix}$$

Example (cont.)

Then set

$$R_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.58 & -0.82 \\ 0 & 0.82 & 0.58 \end{bmatrix}$$

so that

$$A_2 = R_2\, A_1 = \begin{bmatrix} 1.41 & -1.41 \\ 0 & 3.47 \\ 0 & 0 \end{bmatrix} = \widetilde{R}$$

We conclude that $A = R_1^T R_2^T \widetilde{R} = Q\, R$ with

$$Q = \begin{bmatrix} 0.71 & -0.58 \\ 0 & 0.58 \\ 0.71 & -0.58 \end{bmatrix} \quad \text{and} \quad R = \begin{bmatrix} 1.41 & -1.41 \\ 0 & 2 \end{bmatrix}$$

Complexity

The complexity of the QR factorization using Givens rotations is

$$3\,m\,n^2 + O(m\,n)$$

It is useful in special situations, e.g. for Hessenberg matrices

$$A = \begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & 0 & \times & \times \end{bmatrix}$$

The condition number of a rectangular matrix

For a $n \times n$ matrix A we have that

$$\|A\|_2 = \lambda_{\max}(A^T A)^{1/2} \qquad (\lambda_{\max} \text{ the largest eigenvalue})$$

and so

$$\kappa_2(A) = \left(\frac{\lambda_{\max}(A^T A)}{\lambda_{\min}(A^T A)} \right)^{1/2}$$

When A is $m \times n$, we define its *condition number* as

$$\kappa_2(A) := \kappa_2(A^T A)^{1/2} = \left(\frac{\lambda_{\max}(A^T A)}{\lambda_{\min}(A^T A)} \right)^{1/2}$$

It measures how far is A from being *rank deficient*

Numerical aspects

The *forward error analysis* of the LSP is controlled by $\kappa_2(A)$

QR factorization via Householder reflections or Givens rotations is *backwards stable*: if

$$A = Q\,R$$

and $Q + \delta\,Q$ and $R + \delta\,R$ are the computed factors, then

$$A + \delta\,A = (Q + \delta\,Q)\,(R + \delta\,R)$$

where the relative error is bounded by

$$\frac{\|\delta\,A\|_2}{\|A\|_2} \leqslant O(n\,\varepsilon)$$

with ε the machine epsilon

QR factorization *versus* normal equations

Hence when $m \gg n$, the QR factorization via Householder or Givens solves the LSP with $\approx 2\,n^2\,m$ flops or $\approx 3\,n^2\,m$ flops respectively, and a loss of precision of

$$\approx \log_b \kappa_2(A) \text{ digits}$$

On the other hand, solving the normal equations via Cholesky solves the LSP with $\approx n^2 m$ flops and a loss of precision of

$$\approx \log_b \kappa_2(A^T A) = 2\,\log_b \kappa_2(A) \text{ digits}$$

Normal equations is the method of choice to solve the LSP when A is well-conditionned. If A is badly conditionned, then we might prefer applying applying the QR factorization via Householder or Givens, or the SVD *(to be discussed later)*