The matrix 2-norm

◁ **Example:**

The 2-norm is the natural norm for LSQ problems (minimizing $\| \mathbf{r} \|_2$) $\Longrightarrow$ can no longer avoid the matrix 2-norm :

for a square matrix $\mathbf{A}$ (see 'MatrixNorms' for proof)

$$\|\mathbf{A}\|_2 = \sqrt{\lambda_{\max}(\mathbf{A}^T\mathbf{A})}$$

$\lambda_{\max}(\mathbf{A}^T\mathbf{A})$ is the largest eigenvalue of $\mathbf{A}^T\mathbf{A}$
(all eigenvalues are positive since $\mathbf{A}^T\mathbf{A}$ is positive definite).

Singular value decomposition SVD

It is easier to characterize the condition number in the 2-norm in terms of the singular values of **A**. To do that we need the

Definition

A $m \times n$ real matrix **A** has the **singular value decomposition**

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$$

where

1. **U** is $m \times m$ orthogonal matrix
2. Σ is a diagonal $m \times n$ real matrix
3. **V** is $n \times n$ orthogonal matrix

The non-negative diagonal entries $\{\sigma_k \geq 0\}$ in Σ are called the singular values of **A**.
In our case, where $m > n$, there are n positive singular values $\sigma_1 \geq \sigma_2 \ldots \geq \sigma_n > 0$, if **A** is of full rank.

The matrix 2-norm

Then from the definition above, we get

$$\boxed{\|\mathbf{A}\|_2 = \sigma_1(\mathbf{A})}$$

Proof:

The pseudoinverse

The pseudoinverse $\mathbf{A}^\dagger = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ can be expressed in terms of the SVD of $\mathbf{A}$:

$$\mathbf{A}^\dagger = \mathbf{V}\mathbf{\Sigma}^\dagger\mathbf{U}^T$$

where $\Sigma^\dagger$ is the $n \times m$ diagonal matrix, with entries $\{1/\sigma_k\}$.

Proof:

The condition number of a rectangular matrix

By the same argument, we get

$$\boxed{||\mathbf{A}^\dagger||_2 = 1/\sigma_n(\mathbf{A})}$$

By the same argument as before, the condition number of a rectangular matrix is given by

$$\kappa_2(\mathbf{A}) = ||\mathbf{A}||_2||\mathbf{A}^\dagger||_2 = \sigma_1(\mathbf{A})/\sigma_n(\mathbf{A})$$

Proof:

Sensitivity of the normal equations

The normal equations

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$$

are a linear system, so the sensitivity is given by the condition number of $\mathbf{A}^T\mathbf{A}$. But

$$\kappa_2(\mathbf{A}^T\mathbf{A}) = \kappa_2(\mathbf{A})^2$$

Proof:

Sensitivity of the LSQ problem

It turns out : condition number of LSQ problem is

- $\approx \kappa_2(\mathbf{A})$ if the fit to the data is good (not much scatter)
- $\approx \kappa_2(\mathbf{A})^2$ if the fit to the data is poor (a lot of scatter)

$\Longrightarrow$ using normal equations **worsens conditioning of problem** (if the fit is good)

Better ways to solve the LSQ problem

- If **A** is of full rank, use another matrix factorization — the **QR factorization**

- For rank-deficient matrices, use the **QR factorization with column pivoting** (MATLAB) aka Rank-revealing QR
or the **singular value decomposition SVD**

We'll assume **A** is of full rank.

QR factorization

The idea of ('economy-size') QR factorization is:

- form a factorization

$$\mathbf{A} = \mathbf{Q}\mathbf{R}$$

  where $\mathbf{Q}$ is orthogonal $m \times n$ matrix, $\mathbf{R}$ is upper triangular $n \times n$ matrix i.e. $\mathbf{Q}^T\mathbf{Q} = \mathrm{I}_n$

- to solve $\mathbf{A}\mathbf{x} = \mathbf{b}$,

$$\mathbf{Q}\mathbf{R}\mathbf{x} = \mathbf{b} \Rightarrow \mathbf{Q}^T\mathbf{Q}\mathbf{R}\mathbf{x} = \mathbf{Q}^T\mathbf{b} \Rightarrow \mathbf{R}\mathbf{x} = \mathbf{Q}^T\mathbf{b}$$

  so solve a triangular system for $\mathbf{x}$ in $O(n^2)$ ops!

Gram-Schmidt process

You've seen a QR factorization before (in disguise) in **Gram-Schmidt orthogonalization**:

> *given a set of linearly independent vectors* $\{a_1, a_2 \cdots a_n\}$ *forming an n-D subspace of* $\mathbb{R}^m$*, Gram-Schmidt orthogonalization produces a set of orthonormal vectors* $\{q_1, q_2 \cdots q_n\}$*, an orthonormal basis of the same subspace.*

$$\mathbf{A} = \mathbf{QR}$$

i.e. use triangular transformations to produce an orthogonal matrix. We don't do it this way because it's numerically unstable; instead we use orthogonal transformations to turn **A** into **R**.

Orthogonal transformations

Orthogonal transformations are good because:

- they involve perfectly-conditioned matrices
- they don't change the conditioning of the problem
- they don't change the solution of the LSQ problem

Proofs:

Complexity of QR

The QR factorization takes $n^2(m - n/3)$ ops
i.e. for $m \gg n$, $\approx$ twice as expensive as Cholesky factorization of normal equations
but allows us to handle a larger class of matrices.

Example

For square systems, can use QR (normwise backward stable) $\rightarrow$ takes $2n^3/3$ ops
twice as expensive as GEPP but no issues re growth factor etc.

QR in MATLAB

I have described what MATLAB calls 'economy-size QR' factorization.

$$\mathbf{A} = \mathbf{Q}\mathbf{R}$$

where $\mathbf{Q}$ is orthogonal $m \times n$ matrix, $\mathbf{R}$ is upper triangular $n \times n$ matrix.
This is all we need for the LSQ problem.

MATLAB by default produces the 'full QR' factorization

$$\mathbf{A} = \bar{\mathbf{Q}}\bar{\mathbf{R}}$$

where $\bar{\mathbf{Q}}$ is orthogonal $m \times m$ matrix, $\bar{\mathbf{R}}$ is upper triangular $m \times n$ matrix

$$\bar{\mathbf{Q}} = [\mathbf{Q} \mid \text{extra orthog. cols}]\,; \quad \bar{\mathbf{R}} = \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix}$$

so that $\bar{\mathbf{Q}}^T\bar{\mathbf{Q}} = \mathrm{I}_m$. The extra columns are never used in the LSQ problem.

Using QR

Since $\mathbf{A}^T\mathbf{A} = \mathbf{R}^T\mathbf{Q}^T\mathbf{Q}\mathbf{R} = \mathbf{R}^T\mathbf{R}$, $\mathbf{R}$ is the Cholesky factor of $\mathbf{A}^T\mathbf{A}$.
Hence to solve LSQ problem, we could:

1. `[q,r]=qr(A,0); x=r\(q'*b);`
   easiest to understand

2. `r=triu(qr(A)); x=r\(r'\(A'*b));`
   better since never need to form $\mathbf{Q}$

3. `x=A\b;`
   \ acting on overdetermined system does the same as 2 (unless $\mathbf{A}^T\mathbf{A}$ is rank-deficient)

The rank-deficient case

Suppose the matrix $\mathbf{A}$ has rank $k < n$ i.e. is rank-deficient or not of full rank. This means the columns of $\mathbf{A}$ are not linearly independent. In this case, there is no unique solution, and we usually use the solution with minimum 2-norm. This solution can be found by either of:

1. QR factorization with column pivoting aka Rank revealing QR (RR-QR), based on

$$\mathbf{AE} = \mathbf{QR}$$

   obtained in MATLAB by a 3-output call to qr

2. the SVD

In the latter case, let

$$\mathbf{A} = \mathbf{U\Sigma V}^T = [\mathbf{U}_1\ \mathbf{U}_2] \begin{bmatrix} \mathbf{\Sigma}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} [\mathbf{V}_1\ \mathbf{V}_2]$$

then

$$\mathbf{x}_{LS} = \mathbf{A}^{\dagger}\mathbf{b} = \mathbf{V}_1 \mathbf{\Sigma}_1^{-1} \mathbf{U}_1^T \mathbf{b}$$

End of Week 8