The matrix 2-norm

If A is sym ($A^T = A$), $\|A\|_2 = |\lambda_{max}(A)|$

◁ **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**

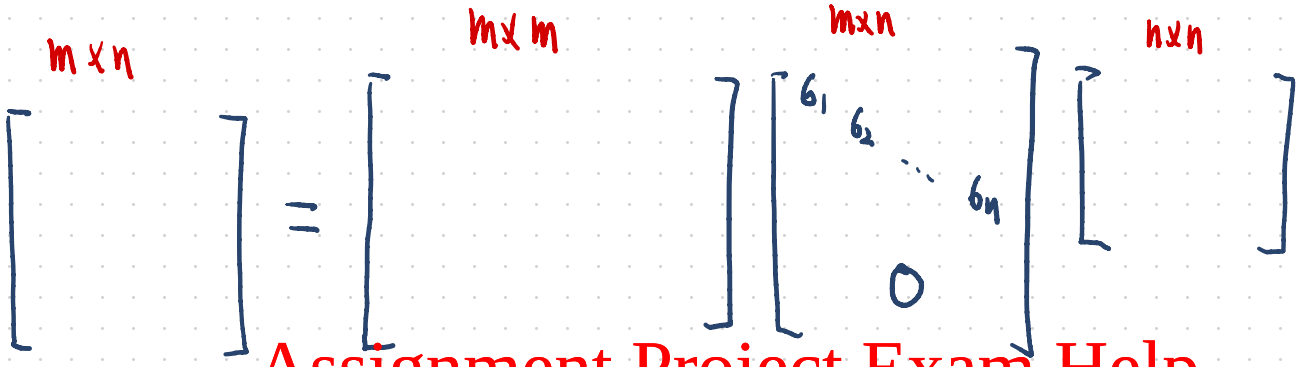
$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T$$

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.



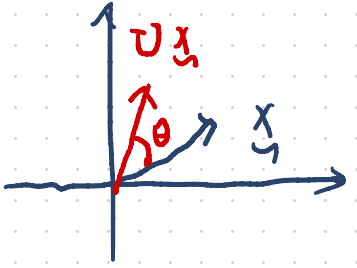
$$A = U \Sigma V^T$$

U & V are orthogonal, i.e. $U^T U = U U^T = I_{m \times m} \Leftrightarrow U^{-1} = U^T$

$V^T V = V V^T = I_{n \times n} \Leftrightarrow V^{-1} = V^T$

Example:

$$U = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$



orthogonal matrix

$\Leftrightarrow$ rotation transformation

$m > n$

$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots \geq \sigma_n \geq 0$

↓

singular value of matrix A

If A is full rank, $\sigma_n > 0$

Lemma: If U is an orthogonal matrix, then $\|Ux\|_2 = \|x\|_2$.

Proof: $\|Ux\|_2^2 = (Ux)^T Ux = x^T U^T U x = x^T x = \|x\|_2^2$

Lemma: If $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal, then $\|UA\|_2 = \|AV\|_2 = \|A\|_2$. $A \in \mathbb{R}^{m\times n}$

Proof: $\|UA\|_2 = \max_{x\neq 0} \frac{\|UAx\|_2}{\|x\|_2} = \max_{x\neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \|A\|_2$

$$\|AV\|_2 = \max_{x\neq 0} \frac{\|AVx\|_2}{\|x\|_2} \overset{y=Vx}{=} \max_{x\neq 0} \frac{\|Ay\|_2}{\|x\|_2} = \max_{y\neq 0} \frac{\|Ay\|_2}{\|y\|_2} = \|A\|_2$$

$$\|y\|_2 = \|Vx\|_2 = \|x\|_2$$

The matrix 2-norm

Then from the definition above, we get

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

Proof:

$A = U \Sigma V^T$

$\|A\|_2 = \|U \Sigma V^T\|_2 = \|\Sigma\|_2.$

$\Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \sigma_n \\ & & & 0 \end{bmatrix} \in \mathbb{R}^{m \times n}$

$\|\Sigma\|_2 = \max_{x \neq 0} \frac{\|\Sigma x\|_2}{\|x\|_2} = \max_{x \neq 0} \frac{\sqrt{\sigma_1^2 x_1^2 + \cdots + \sigma_n^2 x_n^2}}{\sqrt{x_1^2 + \cdots + x_n^2}} \leq \sigma_1$

$x \in \mathbb{R}^n$

$\leq \sigma_1 \sqrt{x_1^2 + x_2^2 \cdots + x_n^2}$

$$\|\Sigma\|_2 = \max_x \frac{\|\Sigma x\|_2}{\|x\|_2} \geq \frac{\|\Sigma e_1\|_2}{\|e_1\|_2} = \frac{\sigma_1}{1} = \sigma_1$$

$$\Sigma e_1 = \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_n \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \sigma_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

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:

$$A^+ = (A^TA)^{-1}A^T$$

$$A = U\Sigma V^T$$

$$A^TA = V\Sigma^T U^T U\Sigma V^T$$

$$= V\Sigma^T\Sigma V^T$$

$$= V\Lambda V^T$$

$$\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ 0 & & \end{bmatrix}$$

eigenvalue decomposition

$$\Lambda = \Sigma^T\Sigma = \text{diag}\begin{pmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_n^2 \end{pmatrix} = \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_n^2 \end{bmatrix}$$

of A^TA

$$(A^TA)^{-1} = (V\Lambda V^T)^{-1} = (V^T)^{-1}\Lambda^{-1}V^{-1} = V\Lambda^{-1}V^T$$

$$\Lambda^{-1} = \begin{bmatrix} 1/\sigma_1^2 & & \\ & \ddots & \\ & & 1/\sigma_n^2 \end{bmatrix}$$

the largest singular value of $A^TA = \sigma_1^2$

-- smallest --

$$(A^TA)^{-1}A^T = V\Delta^{-1}V^T(U\Sigma V)^T$$

$$= V\Delta^{-1}V^TV\Sigma^TU^T$$

$$= V\Delta^{-1}\Sigma^TU^T$$

$$\Sigma^+$$

$$\Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \sigma_n \\ & & 0 & \end{bmatrix}$$

$$\Sigma^T = \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & 0 \\ & & \sigma_n & \end{bmatrix}$$

$$\Sigma^+ = \Delta^{-1}\Sigma^T = \begin{bmatrix} 1/\sigma_1^2 & & \\ & \ddots & \\ & & 1/\sigma_n^2 \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \ddots & & 0 \\ & & \sigma_n & \end{bmatrix}$$

$$= \begin{bmatrix} 1/\sigma_1 & & & \\ & \ddots & & 0 \\ & & 1/\sigma_n & \end{bmatrix}$$

The condition number of a rectangular matrix

$A^\dagger = V \Sigma^\dagger U^T \qquad \|A^\dagger\|_2 = \|\Sigma^\dagger\|_2$

By the same argument, we get

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

By the same argument as before, the condition number of a rectangular matrix is given by $\kappa_2(A) = \|A\|_2 \|A^\dagger\|_2$

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

Proof:

↓ the ratio of the largest singular value to the smallest singular value.

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:

$$k_2(A^TA) \simeq \frac{\text{the largest singular value of } A^TA}{\text{smallest of } A^TA} \simeq \frac{\sigma_1^2}{\sigma_n^2} \simeq \left(\frac{\sigma_1}{\sigma_n}\right)^2$$

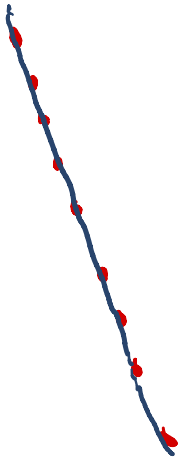
$$\simeq k_2(A)^2.$$

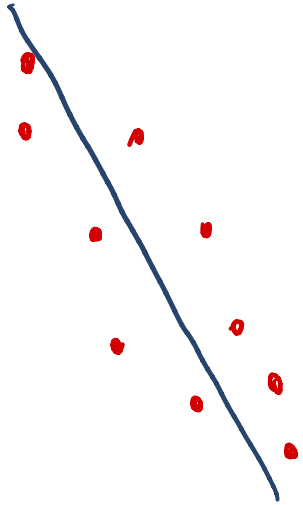
Sensitivity of the LSQ problem

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

$A \in \mathbb{R}^{m \times n}, \quad m > n$

if $\text{rank}(A) = n \Rightarrow$ full rank.

$\text{rank}(A) < n \Rightarrow$ rank deficient.

- 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

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

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 $\mathbf{Ax} = \mathbf{b}$,

upper triangular matrix

$$\mathbf{QRx} = \mathbf{b} \Rightarrow \mathbf{Q}^T\mathbf{QRx} = \mathbf{Q}^T\mathbf{b} \Rightarrow \mathbf{Rx} = \mathbf{Q}^T\mathbf{b}$$

so solve a triangular system for $\mathbf{x}$ in $\mathrm{O}(n^2)$ 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**.

Givens' transform, Household transformation

Gram – Schmidt process

$\{\underset{\sim}{a}_1, \underset{\sim}{a}_2, \cdots \underset{\sim}{a}_n\}$ linearly independent in $\mathbb{R}^m$

$\underset{\sim}{q}_1 = b_{11}\underset{\sim}{a}_1 \qquad \underset{\sim}{q}_1 \cdot \underset{\sim}{q}_1 = 1 \Rightarrow b_{11}$

$\underset{\sim}{q}_2 = b_{12}\underset{\sim}{a}_1 + b_{22}\underset{\sim}{a}_2 \qquad \underset{\sim}{q}_2 \cdot \underset{\sim}{q}_1 = 0,\ \underset{\sim}{q}_2 \cdot \underset{\sim}{q}_2 = 1 \Rightarrow b_{12}, b_{22}$

$\vdots \quad \underset{\sim}{q}_1 \cdots \underset{\sim}{q}_{k-1}$

$\underset{\sim}{q}_k = b_{1k}\underset{\sim}{a}_1 + b_{2k}\underset{\sim}{a}_2 + \cdots + b_{k-1,k}\underset{\sim}{q}_{k-1} + b_{kk}\underset{\sim}{a}_k$

$\underset{\sim}{q}_n = b_{1n}\underset{\sim}{a}_1 + \cdots + b_{n-1,n}\underset{\sim}{q}_{n-1} + b_{nn}\underset{\sim}{a}_n$

$\underset{\sim}{q}_n \cdot \underset{\sim}{q}_1 = 0, \cdots \underset{\sim}{q}_n \cdot \underset{\sim}{q}_{n-1} = 0 \Rightarrow b_{1n}, b_{2n}, \cdots b_{n-1,n}, b_{nn}$

$\Rightarrow$

$\underset{\sim}{a}_1 = r_{11}\underset{\sim}{q}_1$

$\underset{\sim}{a}_2 = r_{12}\underset{\sim}{q}_1 + r_{22}\underset{\sim}{q}_2$

$\vdots$

$\underset{\sim}{a}_n = r_{1n}\underset{\sim}{q}_1 + r_{2n}\underset{\sim}{q}_2 + \cdots + r_{n-1,n}\underset{\sim}{q}_{n-1} + r_{nn}\underset{\sim}{q}_n$

$$\begin{bmatrix} \underset{\sim}{a}_1 & \underset{\sim}{a}_2 & \cdots & \underset{\sim}{a}_n \end{bmatrix} = \begin{bmatrix} \underset{\sim}{q}_1 & \underset{\sim}{q}_2 & \cdots & \underset{\sim}{q}_n \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & r_{22} & & r_{2n} \\ & & & \\ & & & r_{nn} \end{bmatrix}$$

A	Q	R
$\mathbb{R}^{m\times n}$	$\mathbb{R}^{m\times n}$	$\mathbb{R}^{n\times n}$

Orthogonal transformations

$$x_{LS} = \arg\min_x \|r\|_2 = \|b - Ax\|$$

$$Ax = b \rightarrow QAx = Qb$$

$$r \rightarrow Qr$$

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

$\|Qv\|_2 = \|v\|_2$

Proofs:

$k_2(U) = 1.$

$\|U\|_2 = 1 \Rightarrow \|U\|_2 = \max_{x \neq 0} \frac{\|Ux\|_2}{\|x\|_2} \geq \max_{x \neq 0} \frac{\|x\|_2}{\|x\|_2} \geq 1$

$k_2(QA) = \|QA\|_2 \, \|(QA)^\dagger\|_2$

$(QA)^\dagger = A^\dagger Q^\dagger = A^\dagger Q^T$

$= \|A\|_2 \, \|A^\dagger Q^T\|_2$

$= \|A\|_2 \, \|A^\dagger\|_2 = k_2(A)$

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.

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

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)

A has full rank

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

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

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