Lec 18: Overdetermined Linear Systems

– QR Factorization

Preliminary: Orthogonality

Normal Equation Revisited

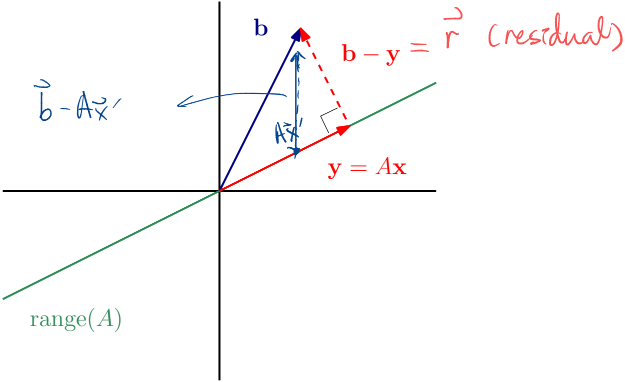
$\underbrace{A}_{\text{tall rectangle}}\vec{x}$ "=" $\vec{b}$ (LLS) $\longrightarrow$ $\underbrace{A^T A}_{\text{square}}\vec{x} = A^T\vec{b}$ [Normal Eqn.]

Alternate perspective on the "normal equation":

$$\underbrace{A^{\mathrm{T}}(\mathbf{b} - A\mathbf{x})}_{\in N(A^T)} = \mathbf{0} \iff \mathbf{z}^{\mathrm{T}}(\underbrace{\mathbf{b} - A\mathbf{x}}_{\text{residual} \,=\, \mathbf{r}}) = \underset{\text{(scalar)}}{\cancel{\mathbf{0}}\ 0} \quad \text{for all } \mathbf{z} \in \mathcal{R}(A),$$

i.e., $\mathbf{x}$ solves the normal equation if and only if the residual is orthogonal to the range of A.

$N(A^T) \perp R(A)$



Orthogonal Vectors

Recall that the angle θ between two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ satisfies

when $\theta = \pi/2$ (perp.)

$$\boxed{\cos(\theta) = \frac{\mathbf{u}^{\mathrm{T}}\mathbf{v}}{\|\mathbf{u}\|_2 \|\mathbf{v}\|_2}.}$$

$\left(\frac{\text{dot}}{\text{inner}}\right)$ product

$\cos(\theta) = 0$

$\Rightarrow \vec{u}^T \vec{v} = 0$

Definition 1

- Two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ are **orthogonal** if $\mathbf{u}^{\mathrm{T}}\mathbf{v} = 0$.
- Vectors $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k \in \mathbb{R}^n$ are **orthogonal** if $\mathbf{q}_i^{\mathrm{T}}\mathbf{q}_j = 0$ for all $i \neq j$.
- Vectors $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k \in \mathbb{R}^n$ are **orthonormal** if $\mathbf{q}_i^{\mathrm{T}}\mathbf{q}_j = \delta_{i,j}$.

$\perp$ unit length

Notation. (Kronecker delta function)

$$\delta_{i,j} = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}$$

- Each of $\vec{q}_1, \ldots, \vec{q}_k$ is a unit vector in 2-norm
- pairwise orthogonal.

Matrices with Orthogonal Columns

(i,j) entry $= \vec{q}_i^{\,T} \vec{q}_j$ (inner product) i.e. a scalar.

Let $Q = [\ \mathbf{q}_1 \mid \mathbf{q}_2 \mid \cdots \mid \mathbf{q}_k\ ] \in \mathbb{R}^{n \times k}$. Note that

In general

$$\underbrace{Q^{\mathrm{T}}Q}_{(k\times k)} = \begin{bmatrix} \mathbf{q}_1^{\mathrm{T}} \\ \hline \mathbf{q}_2^{\mathrm{T}} \\ \hline \vdots \\ \hline \mathbf{q}_k^{\mathrm{T}} \end{bmatrix} \left[\begin{array}{c|c|c|c} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_k \end{array}\right] = \begin{bmatrix} \mathbf{q}_1^{\mathrm{T}}\mathbf{q}_1 & \mathbf{q}_1^{\mathrm{T}}\mathbf{q}_2 & \cdots & \mathbf{q}_1^{\mathrm{T}}\mathbf{q}_k \\ \mathbf{q}_2^{\mathrm{T}}\mathbf{q}_1 & \mathbf{q}_2^{\mathrm{T}}\mathbf{q}_2 & \cdots & \mathbf{q}_2^{\mathrm{T}}\mathbf{q}_k \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{q}_k^{\mathrm{T}}\mathbf{q}_1 & \mathbf{q}_k^{\mathrm{T}}\mathbf{q}_2 & \cdots & \mathbf{q}_k^{\mathrm{T}}\mathbf{q}_k \end{bmatrix}.$$

Therefore,

- $\mathbf{q}_1, \ldots, \mathbf{q}_k$ are orthogonal. $\iff$ $Q^{\mathrm{T}}Q$ is a $k \times k$ diagonal matrix.
- $\mathbf{q}_1, \ldots, \mathbf{q}_k$ are orthonormal. $\iff$ $Q^{\mathrm{T}}Q$ is the $k \times k$ identity matrix.

$\vec{q}_i^{\,T} \vec{q}_j = 0$ if $i \neq j$

Matrices with Orthonormal Columns

Theorem 2

Let $Q = [\, \mathbf{q}_1 \mid \mathbf{q}_2 \mid \cdots \mid \mathbf{q}_k \,] \in \mathbb{R}^{n\times k}$ and suppose that $\mathbf{q}_1, \dots, \mathbf{q}_k$ are orthonormal. Then

i.e., Q is a matrix w/ orthonormal columns.

1. $Q^{\mathrm{T}} Q = I \in \mathbb{R}^{k\times k}$;
2. $\|Q\mathbf{x}\|_2 = \|\mathbf{x}\|_2$ *for all* $\mathbf{x} \in \mathbb{R}^k$; ($Q$ preserves 2-norm)
3. $\|Q\|_2 = 1$.

Proof

① Explained in the previous page.

② $$\|Q\vec{x}\|_2^2 = (Q\vec{x})^T Q\vec{x} = \vec{x}^T \underbrace{Q^T Q}_{=I} \vec{x} = \vec{x}^T\vec{x} = \|\vec{x}\|_2^2$$

③ Exercise. Hint: Use the definition of induce matrix norm.

Orthogonal Matrices

Definition 3

We say that $Q \in \mathbb{R}^{n\times n}$ is an **orthogonal matrix** if $Q^{\mathrm{T}}Q = I \in \mathbb{R}^{n\times n}$.

- A square matrix with orthogonal columns is not, in general, an orthogonal matrix!

(unless they are orthonormal)

Ex $n=3$

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

$$\begin{array}{ccc} \uparrow & \uparrow & \uparrow \\ \vec{a}_1 & \vec{a}_2 & \vec{a}_3 \end{array}$$

- columns are orthogonal

$$\begin{cases} \cdot\ \vec{a}_1^{T}\vec{a}_2 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ -2 \end{bmatrix} = 1\cdot 0 + 0\cdot 0 + 0\cdot(-2) = 0 \\ \cdot\ \vec{a}_1^{T}\vec{a}_3 = \vec{a}_2^{T}\vec{a}_3 = 0 \end{cases}$$

- 2^{nd} and 3^{rd} columns are not unit vectors.

Thus, A is not an orthogonal matrix.

Properties of Orthogonal Matrices

Orthogonal matrices enjoy all properties listed in Thm 2.

Theorem 4

Let $Q \in \mathbb{R}^{n\times n}$ be orthogonal. Then

1. $Q^{-1} = Q^{\mathrm{T}}$;
2. Q^{T} *is also an orthogonal matrix;*
3. $\kappa_2(Q) = 1$;
4. *For any* $A \in \mathbb{R}^{n\times n}$, $\|AQ\|_2 = \|A\|_2$;
5. *if* $P \in \mathbb{R}^{n\times n}$ *is another orthogonal matrix, then* PQ *is also orthogonal.*

cf In general,

$\|AB\|_2 \le \|A\|_2 \|B\|_2$

Linear eqns involving Q can be solved stably on computer.

5 Exercise (Hint. Show that $(PQ)^T(PQ) = I$.)

3 $\kappa_2(Q) = \|Q\|_2 \|Q^{-1}\|_2$

$= 1 \cdot 1 = 1$ by Thm 2 3 and 1 + 2.

Why Do We Like Orthogonal Vectors?

theoretical benefit

- If $\mathbf{u}$ and $\mathbf{v}$ are orthogonal, then

$\vec{u}^T\vec{v} = 0$

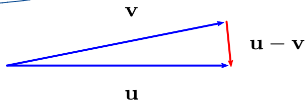
$$\|\mathbf{u} \pm \mathbf{v}\|_2^2 = (\vec{u} \pm \vec{v})^T(\vec{u} \pm \vec{v}) = \vec{u}^T\vec{u} \pm \cancel{2\vec{u}^T\vec{v}} + \vec{v}^T\vec{v} = \|\vec{u}\|_2^2 + \|\vec{v}\|_2^2$$

Pythagorean identity

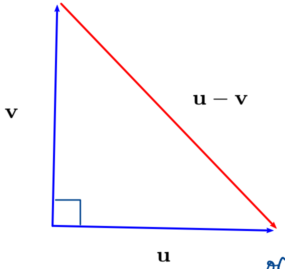
computational or Numerical

- Without orthogonality, it is possible that $\|\mathbf{u} - \mathbf{v}\|_2$ is much smaller than $\|\mathbf{u}\|_2$ and $\|\mathbf{v}\|_2$.
- The addition and subtraction of orthogonal vectors are guaranteed to be well-conditioned.

Catastrophic cancellation



nonorthogonal



orthogonal.

QR Factorization

The QR Factorization

- $A\vec{x} = \vec{b}$ (square) $\longrightarrow$ LU factorization (Gaussian elim.)
- $A\vec{x}$ "=" $\vec{b}$ (rectangle) $\longrightarrow$ QR factorization (Gram-Schmidt)

The following matrix factorization plays a role in solving linear least squares problems similar to that of LU factorization in solving linear systems.

Theorem 5

Let $A \in \mathbb{R}^{m\times n}$ where $m \geqslant n$. Then $A = QR$ where $Q \in \mathbb{R}^{m\times m}$ is orthogonal and $R \in \mathbb{R}^{m\times n}$ is upper triangular.

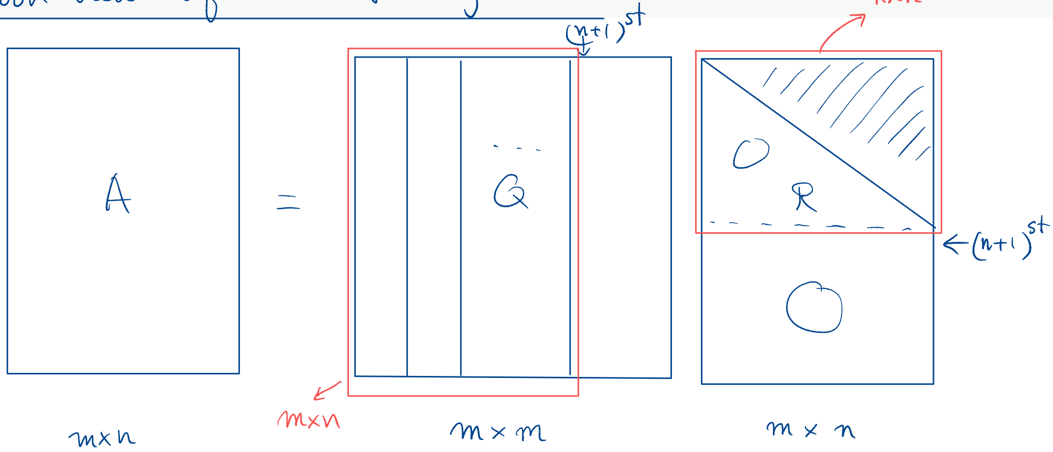
tall rectangular

orthonormal columns

$$\underbrace{\begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_m \end{bmatrix}}_{Q} \underbrace{\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}}_{R}$$

$(m\times n)$ $\qquad$ $(m\times m)$ $\qquad$ $(m\times n)$

Cartoon view of QR factorization



$m \geq n$

Thick VS Thin QR Factorization

$$\begin{cases} A = QR & (\text{Thick}) \\ A = \hat{Q}\hat{R} & (\text{Thin}) \end{cases}$$

- Here is the QR factorization again.

$$A = \underbrace{\left[\begin{array}{c|c|c|c} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_m \end{array}\right]}_{Q} \underbrace{\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}}_{R} \quad \text{(thick) (full)}$$

When A is a very thin & tall matrix

- When m is much larger than n, it is much more efficient to use the *thin* or *compressed* QR factorization.

$$A = \underbrace{\left[\begin{array}{c|c|c|c} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{array}\right]}_{\hat{Q}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \end{bmatrix}}_{\hat{R}} \quad \text{(thin) (compressed)}$$

QR Factorization in MATLAB

Either type of QR factorization is computed by `qr` command.

- Thick/Full QR factorization

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

- Thin/Compressed QR factorization

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

Test the orthogonality of `Q` by calculating the norm of $Q^{\mathrm{T}}Q - I$ where I is the identity matrix with *suitable* dimensions.

```
norm(Q'*Q - eye(m))        % full QR
norm(Q'*Q - eye(n))        % thin QR
```

Least Squares and QR Factorization

Substitute the thin factorization $A = \widehat{Q}\widehat{R}$ into the normal equation $A^{\mathrm{T}}A\mathbf{x} = A^{\mathrm{T}}\mathbf{b}$ and simplify.

Least Squares and QR Factorization (cont')

Summary: Algorithm for LLS Approximation

If A has rank n, the normal equation $A^{\mathrm{T}}A\mathbf{x} = A^{\mathrm{T}}\mathbf{b}$ is consistent and is equivalent to $\widehat{R}\mathbf{x} = \widehat{Q}^{\mathrm{T}}\mathbf{b}$.

1. Factor $A = \widehat{Q}\widehat{R}$.
2. Let $\mathbf{z} = \widehat{Q}^{\mathrm{T}}\mathbf{b}$.
3. Solve $\widehat{R}\mathbf{x} = \mathbf{z}$ for $\mathbf{x}$ using backward substitution.

Least Squares and QR Factorization (cont')

```
function x = lsqrfact(A,b)
% LSQRFACT x = lsqrfact(A,b)
% Sove linear least squares by QR factorization
% Input:
%   A    coefficient matrix (m-by-n, m>n)
%   b    right-hand side (m-by-1)
% Output:
%   x    minimizer of || b - Ax || (2-norm)
    [Q,R] = qr(A,0);            % thin QR fact.
    z = Q'*b;
    x = backsub(R,c);
end
```

Appendix: Gram-Schmidt Orthogonalization

The Gram–Schmidt Procedure

Problem: Orthogonalization

Given $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^m$, construct orthonormal vectors $\mathbf{q}_1, \ldots, \mathbf{q}_n \in \mathbb{R}^m$ such that

$$\operatorname{span}\{\mathbf{a}_1, \ldots, \mathbf{a}_k\} = \operatorname{span}\{\mathbf{q}_1, \ldots, \mathbf{q}_k\}, \quad \text{for any } k \in \mathbb{N}[1, n].$$

- **Strategy.** At the jth step, find a unit vector $\mathbf{q}_j \in \operatorname{span}\{\mathbf{a}_1, \ldots, \mathbf{a}_j\}$ that is orthogonal to $\mathbf{q}_1, \ldots, \mathbf{q}_{j-1}$.
- **Key Observation.** The vector $\mathbf{v}_j$ defined by

$$\mathbf{v}_j = \mathbf{a}_j - \left(\mathbf{q}_1^{\mathrm{T}} \mathbf{a}_j\right) \mathbf{q}_1 - \left(\mathbf{q}_2^{\mathrm{T}} \mathbf{a}_j\right) \mathbf{q}_2 - \cdots - \left(\mathbf{q}_{j-1}^{\mathrm{T}} \mathbf{a}_j\right) \mathbf{q}_{j-1}$$

satisfies the required properties.

GS Algorithm

The Gram–Schmidt iteration is outlined below:

$$
\begin{aligned}
\mathbf{q}_1 &= \frac{\mathbf{a}_1}{r_{11}}, \\
\mathbf{q}_2 &= \frac{\mathbf{a}_2 - r_{12}\mathbf{q}_1}{r_{22}}, \\
\mathbf{q}_3 &= \frac{\mathbf{a}_3 - r_{13}\mathbf{q}_1 - r_{23}\mathbf{q}_2}{r_{33}}, \\
&\vdots \\
\mathbf{q}_n &= \frac{\mathbf{a}_n - \sum_{i=1}^{n-1} r_{in}\mathbf{q}_i}{r_{nn}},
\end{aligned}
$$

where

$$
r_{ij} = \begin{cases} \mathbf{q}_i^{\mathrm{T}}\mathbf{a}_j, & \text{if } i \neq j \\ \pm\left\|\mathbf{a}_j - \sum_{k=1}^{j-1} r_{kj}\mathbf{q}_k\right\|_2, & \text{if } i = j \end{cases} .
$$

GS Procedure and Thin QR Factorization

- The GS algorithm, written as a matrix equation, yields a **thin QR factorization**:

$$A = \underbrace{\begin{bmatrix} \vert & & \vert \\ \mathbf{a}_1 & \cdots & \mathbf{a}_n \\ \vert & & \vert \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} \vert & & \vert \\ \mathbf{q}_1 & \cdots & \mathbf{q}_n \\ \vert & & \vert \end{bmatrix}}_{\hat{Q}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \end{bmatrix}}_{\hat{R}} = \widehat{Q}\widehat{R}$$

- While it is an important tool for theoretical work, the GS algorithm is numerically unstable.