Lec 18: Overdetermined Linear Systems

– QR Factorization

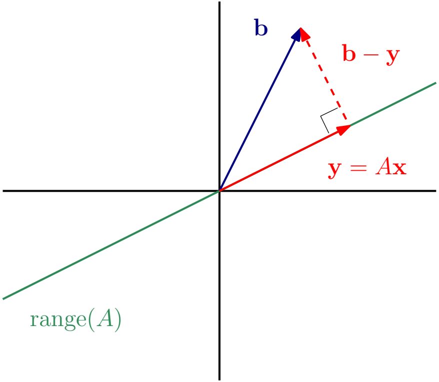
Preliminary: Orthogonality

Normal Equation Revisited

Alternate perspective on the "normal equation":

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

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



Orthogonal Vectors

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

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

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 $\mathbf{q}_i^{\mathrm{T}}\mathbf{q}_j = \delta_{i,j}$.

Notation. (Kronecker delta function)

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

Matrices with Orthogonal Columns

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

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

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.

Matrices with Orthonormal Columns

Theorem 2

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

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

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!

Properties of Orthogonal Matrices

Theorem 4

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

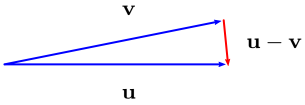
1. $Q^{-1} = Q^{\mathrm{T}}$;
2. Q^{T} *is also an orthogonal matrix;*
3. $\kappa_2(Q) = 1$;
4. *For any* $A \in \mathbb{R}^{n\times n}$, $\|AQ\|_2 = \|A\|_2$;
5. *if* $P \in \mathbb{R}^{n\times n}$ *is another orthogonal matrix, then* PQ *is also orthogonal.*

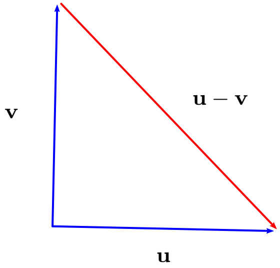
Why Do We Like Orthogonal Vectors?

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

$$\|\mathbf{u} \pm \mathbf{v}\|_2^2 =$$

- 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.





QR Factorization

The QR Factorization

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.

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

Thick VS Thin QR Factorization

- Here is the QR factorization again.

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

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

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

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**:

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

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