## **QR** Factorization

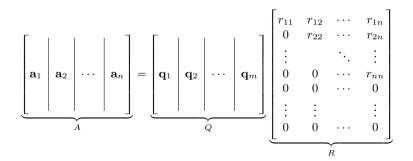
# **QR** Factorization

#### 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.

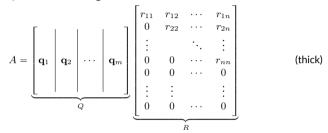
#### Theorem 1

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



#### Thick VS Thin QR Factorization

· Here is the QR factorization again.



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

$$A = \left[\begin{array}{c|ccc} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{array}\right] \underbrace{\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \end{array}}_{\widehat{R}}$$
 (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  ${\mathcal Q}$  by calculating the norm of  $Q^{\rm T}Q-I$  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

#### Moore-Penrose Pseudoinverse

Let  $A \in \mathbb{R}^{m \times n}$  with  $m \ge n$  and suppose that columns of A are linearly independent.

- The least square problem  $A\mathbf{x}$  "="  $\mathbf{b}$  is equivalent to the normal equation  $A^{\mathrm{T}}A\mathbf{x} = A^{\mathrm{T}}\mathbf{b}$ , which is a square matrix equation.
- The solution can be written as

$$\mathbf{x} = \left(A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}}\mathbf{b}.$$

The matrix

$$A^{+} = \left(A^{\mathrm{T}}A\right)^{-1}A^{\mathrm{T}} \in \mathbb{R}^{n \times m},$$

is called the (Moore-Penrose) pseudoinverse.

#### Moore-Penrose Pseudoinverse (cont')

- MATLAB's backslash is mathematically equivalent to left-multiplication by the inverse or pseudoinverse of a matrix.
- MATLAB's pinv calculates the pseudoinverse, but it is rarely used in practice, just as inv.
- $A^+$  can be calculated by using the thin QR factorization  $A=\hat{Q}\hat{R}$ .

$$A^+ = \widehat{R}^{-1}\widehat{Q}^{\mathrm{T}}.$$

It can be done using the thick QR factorization as seen on p. 1624 of the text.

#### **Least Squares Using QR Factorization**

We now reveal the connection between QR factorization and the LLS approximation.

Substitute the thin factorization  $A=\hat{Q}\hat{R}$  into the normal equation  $A^{\rm T}A{\bf x}=A^{\rm T}{\bf b}$  and simplify.

#### Least Squares Using 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  $\hat{R}\mathbf{x}=\hat{Q}^{\mathrm{T}}\mathbf{b}$ .

- 2 Let  $\mathbf{z} = \hat{Q}^{\mathrm{T}} \mathbf{b}$ .
- **3** Solve  $\hat{R}\mathbf{x} = \mathbf{z}$  for  $\mathbf{x}$  using backward substitution.

#### Least Squares Using QR Factorization (cont')

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

# Appendix: More on Pseudoinverse and Normal Equation

## **Analytical Properties of Pseudoinverse**

The matrix  $A^{\rm T}A$  appearing in the definition of  $A^+$  satisfies the following properties.

#### Theorem 2

For any  $A \in \mathbb{R}^{m \times n}$  with  $m \geqslant n$ , the following are true:

- **1**  $A^{\mathrm{T}}A$  is symmetric.
- **2**  $A^{\mathrm{T}}A$  is singular if and only if  $\operatorname{rank}(A) < n$ .
- **3** If  $A^{T}A$  is nonsingular, then it is positive definite.

A symmetric positive definite (SPD) matrix S such as  $A^{\rm T}A$  permits so-called the **Cholesky factorization** 

$$S = R^{\mathrm{T}}R$$

where R is an upper triangular matrix.

#### Least Squares Using Cholesky Factorization

One can solve the LLS problem  $A\mathbf{x}$  "="  $\mathbf{b}$  by solving the normal equation  $A^{\mathrm{T}}A\mathbf{x} = A^{\mathrm{T}}\mathbf{b}$  directly as below.

- **1** Compute  $N = A^{T}A$ .
- **2** Compute  $\mathbf{z} = A^{\mathrm{T}}\mathbf{b}$ .
- 3 Solve the square linear system  $N\mathbf{x} = \mathbf{z}$  for  $\mathbf{x}$ .

Step 3 is done using chol which implements the Cholesky factorization.

#### **MATLAB** Implementarion.

### **Conditioning of Normal Equations**

- Recall that the condition number of solving a square linear system  $A\mathbf{x} = \mathbf{b}$  is  $\kappa(A) = \|A\| \|A^{-1}\|$ .
- Provided that the residual norm at the least square solution is relatively small, the conditioning of LLS problem is similar:

$$\kappa(A) = ||A|| ||A^+||.$$

- If A is rank-deficient (columns are linearly dependent), then  $\kappa(A) = \infty$ .
- If an LLS problem is solved solving the normal equation, it can be shown that the condition number is

$$\kappa(A^{\mathrm{T}}A) = \kappa(A)^2.$$