$$\underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix}_{m\times n}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix}_{n\times 1}}_{x} = \underbrace{\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}_{m\times 1}}_{b}$$

CX1104: **Linear Algebra for Computing**

Chap. No : **7.1.5**

Lecture : **Least Squares**

Topic : **Least Squares**

Concept : **Summary Least Squares Solution**

Instructor: **A/P Chng Eng Siong**

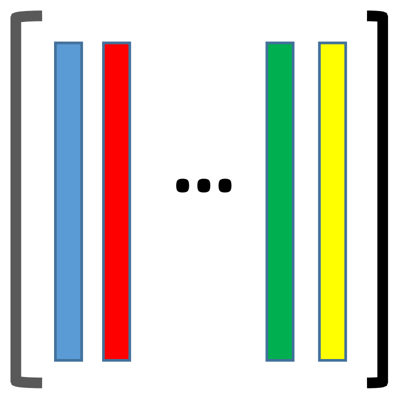
TAs: **Zhang Su, Vishal Choudhari**

Rev: 2nd July 2020

Solving Least Squares using QR Factorisation and MATLAB

Solving $Ax = b$:

Case 1

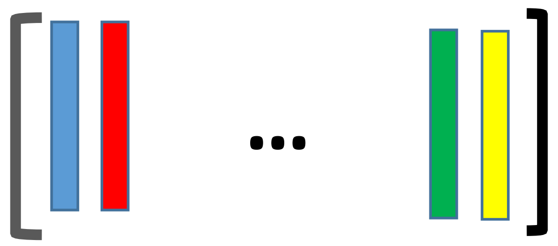


$$M \approx N$$

Say,

- A is square and invertible (full rank)
- then, b is in column space of A

Case 2

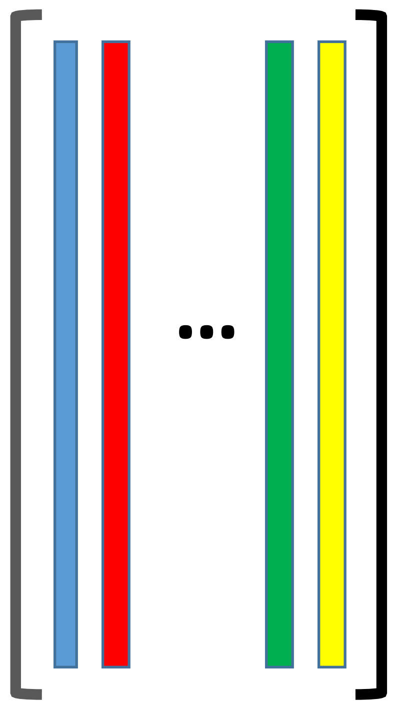


$$M \ll N$$

Under-determined

- As there are more unknowns than equations, infinitely many solutions exist.
- Hence, the goal then becomes to solve for x, such that, $||x||$ is minimised!

Case 3



$$M \gg N$$

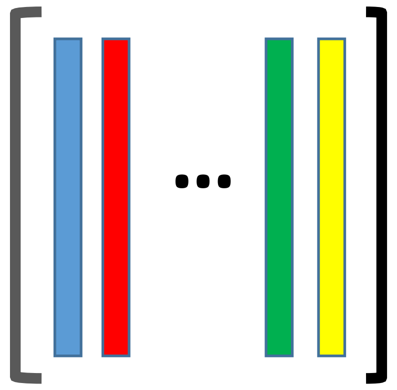
Over-determined

- b may not be in col space of A
- Hence, $b = Ax + \epsilon$
- ϵ models error/noise

Solving Least Squares using QR Factorisation and MATLAB

Solving $Ax = b$:

Case 1



$M \approx N$

Solution:

As b in column space of A, unique solution exists.

$\mathbf{x} = \mathbf{A^{-1}b}$, since A is square and has full rank.

Say,

- A is square and invertible (full rank)
- then, b is in column space of A

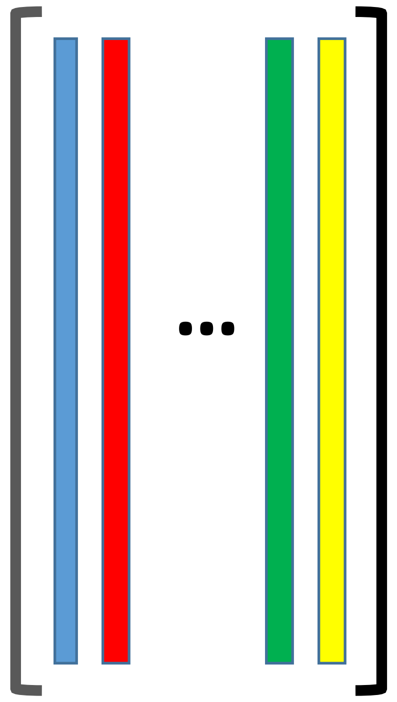
Solving Least Squares using QR Factorisation and MATLAB

For case 3, b is not in col. space of A.

Hence, an estimate of x, denoted by $\hat{x}$, such that least squares error ($||b - Ax||$) is minimised can be found.

Solving $Ax = b$:

Case 3



$M \gg N$

Over-determined

- b may not be in col space of A
- Hence, $b = Ax + \epsilon$
- ϵ models error/noise

Three ways to solve for case 3!

Way 1
Pseudoinverse

$\hat{x} = pinv(A) \times b$

Way 2
Inverting Normal Equation

$\hat{x} = (A^T A)^{-1} A^T b$

Way 3
QR

Let $A = QR$, where $Q^T Q = I$.

Hence, $Ax = b$ can be rewritten as:
$Q^T QRx = Q^T b$ (or)
$Rx = Q^T b$

Therefore, $\hat{x} = R^{-1} Q^T b$

Revisiting QR Factorisation and Solving $Ax = b$

Consider solving the system of equations: $Ax = b$

Through QR factorisation, A can be written as:

$$A = QR$$

where,

The Q Factor:

- Q is $m \times n$ with orthonormal columns ($Q^T Q = I$)
- If A is square $(m = n)$, then Q is orthogonal, i.e, $Q^T Q = QQ^T = I$

The R Factor:

- R is $n \times n$ upper triangular, with nonzero diagonal elements
- R is nonsingular (diagonal elements are nonzero)

So, $Ax = b$ can be rewritten as: $QRx = b$.

Multiplying both sides by Q^T yields:

$$Q^T QRx = Q^T b \text{ (or)}$$
$$Rx = Q^T b$$

Since R is an upper triangular matrix, x can be solved by:

1. Back-substitution
2. On MATLAB: $x = R^{-1} Q^T b$

Algorithm Complexity

1. compute QR factorization $A = QR$ ($2mn^2$ flops if A is $m \times n$)
2. matrix-vector product $d = Q^T b$ ($2mn$ flops)
3. solve $Rx = d$ by back substitution (n^2 flops)

complexity: $2mn^2$ flops

Ref: “Why use QR to solve Ax=b?” by Dr. Peyam
https://www.youtube.com/watch?v=J41Ypt6Mftc