Normal Equations: Error Minimization

Let's look at fitting $f(x) = a + bx$ to a set of Data (x_i, f_i)

System we get:
$$\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \end{bmatrix}$$

$$\underline{A} \quad \underline{x} = \underline{b}$$

It was stated that the solution to

$$\underline{A}^T \underline{A} \, \underline{x} = \underline{A}^T \underline{b} \quad \text{that}$$

minimizes the error:

$$e = \sum \left(f_i - (a + bx_i) \right)^2 = \sum (f_i - a - bx_i)^2$$

To be a minimum the gradient w.r.t. a & b to be zero. $\Rightarrow \frac{\partial e}{\partial a} = 0$ & $\frac{\partial e}{\partial b} = 0$

$$\frac{\partial e}{\partial a} = \sum (f_i - a - bx_i) = 0$$

$$\frac{\partial e}{\partial b} = \sum x_i (f_i - a - bx_i) = 0$$

Write as a linear system:

$$\begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum f_i \\ \sum x_i f_i \end{bmatrix} \quad *$$

$n = \#$ of points (x_i, f_i)

Does $\underline{A}^T \underline{A}\, \underline{x} = \underline{A}^T \underline{b}$ give the same 2×2 system?

For simplicity, look at $n = 3$:

$(x_1, f_1) \qquad (x_2, f_2) \qquad (x_3, f_3)$

$$\underline{A} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \end{bmatrix} \qquad \underline{b} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} \qquad \underline{x} = \begin{bmatrix} a \\ b \end{bmatrix}$$

$$\underline{A}^T \underline{A} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \end{bmatrix} = \begin{bmatrix} 3 & x_1 + x_2 + x_3 \\ x_1 + x_2 + x_3 & x_1^2 + x_2^2 + x_3^2 \end{bmatrix}$$

$$= \begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix}$$

$$\underline{A}^T \underline{b} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} f_1 + f_2 + f_3 \\ x_1 f_1 + x_2 f_2 + x_3 f_3 \end{bmatrix}$$

$$= \begin{bmatrix} \Sigma f_i \\ \Sigma x_i f_i \end{bmatrix}$$

$\Rightarrow \underline{A}^T \underline{A}\, \underline{x} = \underline{A}^T \underline{b}$ results in the same system that minimizes $\underline{e}$,

Orthogonal & Orthonormal basis

Recall that vectors are orthogonal if

$$\underline{a}_0 \cdot \underline{a}_1 = 0$$

An orthogonal basis is one where the vectors of that basis are orthogonal to each other:

ex1) $B = \{ \underline{a}_0, \underline{a}_1, \underline{a}_2 \}$, $\underline{a}_i \cdot \underline{a}_j = 0 \quad i \neq j$

An orthonormal basis is one where we have

$$\underline{a}_i \cdot \underline{a}_j = 0 \quad i \neq j$$

$$\underline{a}_i \cdot \underline{a}_i = 1$$

Now look at a matrix w/ orthonormal columns:

$$\underline{Q} = \begin{bmatrix} | & | & | & & | \\ \underline{q}_1 & \underline{q}_2 & \underline{q}_3 & \cdots & \underline{q}_n \\ | & | & | & & | \end{bmatrix}$$

$\underline{q}_1 \cdot \underline{q}_1 = 1$, $\underline{q}_1 \cdot \underline{q}_2 = 0$, etc

Look at $\underline{Q}^T \underline{Q}$

$$\begin{bmatrix} — q_1^T — \\ — q_2^T — \\ \vdots \end{bmatrix} \begin{bmatrix} | & | & \\ q_1 & q_2 & \cdots \\ | & | & \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ & & & \ddots \end{bmatrix}$$

$$= \underline{I}$$

$$\Rightarrow \underline{Q}^T \underline{Q} = \underline{I}$$

If $\underline{Q}$ is square, then you can show that

$$\underline{Q}\,\underline{Q}^T = \underline{I}$$

$\Rightarrow \underline{Q}^T = \underline{Q}^{-1}$ if $\underline{Q}$ is square.

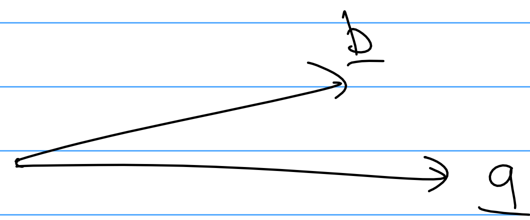
Called a Unitary Matrix

How can we construct $\underline{Q}$?

How to Construct an Orthonormal basis

Given a set of vectors that span a subspace, find an orthonormal basis that also spans that subspace.

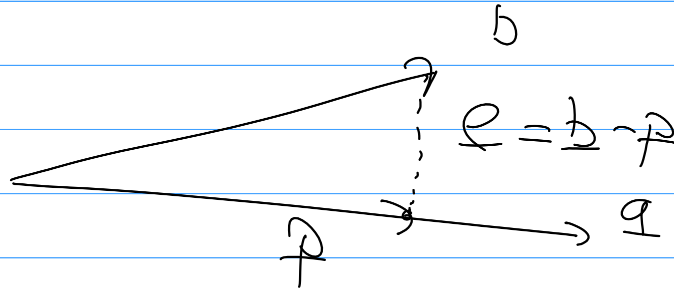
ex1) In 2D: let $\underline{a}$ & $\underline{b}$ be non-parallel vectors in 2D:



They span $\mathbb{R}^2$

But are not orthogonal.

Recall the projection of $\underline{b}$ onto $\underline{a}$



$\underline{e} = \underline{b} - \underline{p} = \underline{b} - \dfrac{\underline{a}^T\underline{b}}{\underline{a}^T\underline{a}}\,\underline{a}$ $\rightarrow$ is perpendicular to $\underline{a}$.

Thus, an orthonormal basis is

$$q_1 = \frac{\underline{a}}{\|\underline{a}\|_2} \qquad q_2 = \frac{\underline{e}}{\|\underline{e}\|_2} = \frac{\underline{b} - (\underline{a}^T\underline{b}/\underline{a}^T\underline{a})\,\underline{a}}{\|\underline{e}\|_2}$$

Are q_1 & q_2 unique? No!

You could project $\underline{a}$ onto $\underline{b}$.

Now look at a matrix.

Find an orthonormal basis to Column Space.

$$\underline{A} = \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \\ a_4 & b_3 & c_4 \end{bmatrix} = [\,\underline{a} \quad \underline{b} \quad \underline{c}\,]$$

Step 1: Set $\underline{t}_1 = \underline{a}$

Step 2: Project onto $\underline{\perp}$ space of $\underline{a}$!

$$\underline{t}_2 = \underline{b} - \frac{\underline{t}_1^T \underline{b}}{\underline{t}_1^T \underline{t}_1} \underline{t}_1$$

$\Rightarrow \underline{t}_1 \cdot \underline{t}_2 = 0$

Step 3: Project onto $\underline{\perp}$ space of $\underline{t}_1$ & $\underline{t}_2$

$$\underline{t}_3 = \underline{c} - \frac{\underline{t}_1^T \underline{c}}{\underline{t}_1^T \underline{t}_1} \underline{t}_1 - \frac{\underline{t}_2^T \underline{c}}{\underline{t}_2^T \underline{t}_2} \underline{t}_2$$

$\Rightarrow \underline{t}_2 \cdot \underline{t}_3 = 0$ & $\underline{t}_1 \cdot \underline{t}_3 = 0$

Step 4: Normalize

$$q_1 = \frac{\underline{t}_1}{\|\underline{t}_1\|} \qquad q_2 = \frac{\underline{t}_2}{\|\underline{t}_2\|} \qquad q_3 = \frac{\underline{t}_3}{\|\underline{t}_3\|}$$

This is called Gram-Schmidt (G-S)

This results in an orthonormal basis to $C(\underline{A})$

How do $\underline{Q}$ & $\underline{A}$ relate?

Recall that G-S stated that

$$q_1 = \frac{\underline{a}}{\|\underline{a}\|} = \frac{\underline{a}}{r_{11}} \Rightarrow \underline{a} = r_{11} \underline{q}_1$$

$$q_2 = \frac{\underline{b} - r_{21} \underline{q}_1}{\|\underline{b} - r_{21} \underline{q}_1\|} = \frac{\underline{b} - r_{21} \underline{q}_1}{r_{22}} = \frac{1}{r_{22}} \underline{b} - \frac{r_{21}}{r_{22}} q_1$$

$\Rightarrow \underline{b} = r_{21} \underline{q}_1 + r_{22} q_2$

Similarly $\Rightarrow \underline{c} = r_{31} q_1 + r_{32} q_2 + r_{33} q_3$

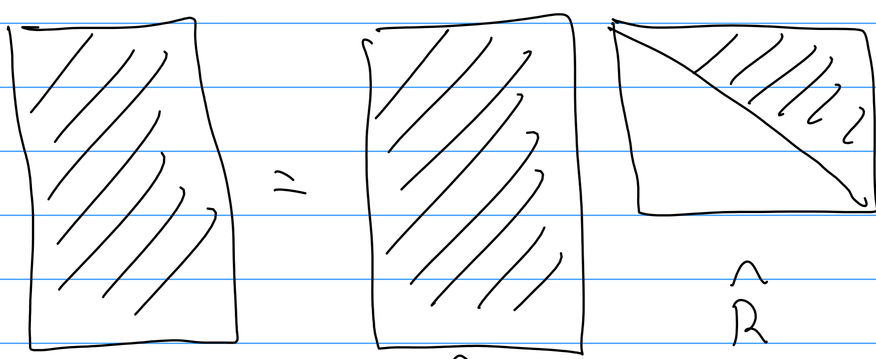
$$\underline{A} = [\underline{a} \;\; \underline{b} \;\; \underline{c}] = \begin{bmatrix} | & | & | \\ q_1 & q_2 & q_3 \\ | & | & | \end{bmatrix} \begin{bmatrix} r_{11} & r_{21} & r_{31} \\ 0 & r_{22} & r_{32} \\ 0 & 0 & r_{33} \end{bmatrix}$$

$$\underbrace{\underline{A} = \underset{\substack{\uparrow \\ \text{Orthonormal} \\ \text{Matrix}}}{\underline{Q}} \; \underset{\substack{\nwarrow \\ \text{Upper triangular}}}{\underline{R}}}{} \quad \leftarrow \text{QR Decomposition}$$

What we just found is actually called the

Reduced QR Decomposition,

Typically written as $\underline{A} = \underline{\hat{Q}} \, \underline{\hat{R}}$



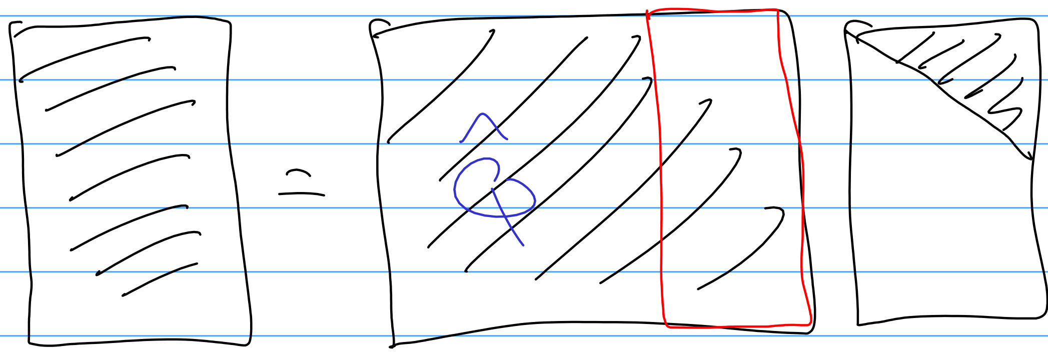
$\hat{R}$

$n \times n$

$\underline{A} = \hat{Q}$

$m \times n$ $\quad$ $m \times n$

You can also determine the Full QR Factorization by appending $\hat{Q}$ to make it $m \times m$, (typically $m \geq n$)



$$\underset{m \times n}{\underline{A}} = \underset{m \times m}{Q} \; \underset{m \times n}{R}$$

The columns q_j for $j > n$ must be orthogonal to the range$(\underline{A})$.

If $\text{rank}(\underline{A}) = n$, then these columns are the orthonormal basis to $\text{null}(\underline{A}^T)$

Why is this useful?

Thm: Every $\underline{A} \in \mathbb{R}^{m \times n}$ with $m \geq n$ has a full QR factorization & a reduced QR factorization

Thm: Each $\underline{A} \in \mathbb{R}^{m \times n}$ with $m \geq n$ of full rank ($\text{rank}(\underline{A}) = n$) has a unique reduced QR with $r_{jj} > 0$

↑ All diagonals of $\underline{R}$ are positive,

$\Rightarrow \underline{R}^{-1}$ exists so does $\underline{R}^{-T}$

Now look at $\underline{A}\,\underline{x} = \underline{b}$, $\underline{A}$ is full rank,

Solve via $\underline{\hat{Q}}\,\underline{\hat{R}}$

1. Decompose: $\underline{A} = \underline{\hat{Q}}\,\underline{\hat{R}}$ ← expensive
2. $\underline{\hat{Q}}\,\underline{\hat{R}}\,\underline{x} = \underline{b} \Rightarrow \underline{\hat{Q}}^T \underline{\hat{Q}}\,\underline{\hat{R}}\,\underline{x} = \underline{\hat{Q}}^T \underline{b}$
3. $\underline{\hat{R}}\,\underline{x} = \underline{\hat{Q}}^T \underline{b}$ ← Solve (cheap)

If $\underline{A}$ does not change, but $\underline{b}$ does

→ Cheap to solve new $\underline{b}$'s.

Note: Drop ^

Now look at $\underline{A}^T \underline{A}\, \underline{x} = \underline{A}^T \underline{b}$

$$\underline{A}^T \underline{A}\, \underline{x} = \underline{A}^T \underline{b} \qquad \underline{A} = \underline{Q}\, \underline{R}$$

$$(\underline{Q}\,\underline{R})^T (\underline{Q}\,\underline{R})\, \underline{x} = (\underline{Q}\,\underline{R})^T \underline{b}$$

$$\underline{R}^T \underbrace{\cancel{\underline{Q}^T \underline{Q}}}_{= I}\, \underline{R}\, \underline{x} = \underline{R}^T \underline{Q}^T \underline{b}$$

$$\underline{R}^T \underline{R}\, \underline{x} = \underline{R}^T \underline{Q}^T \underline{b} \qquad \underline{R}^{-T} \text{ exists}$$

$$\Rightarrow \underline{R}\, \underline{x} = \underline{Q}^T \underline{b}$$

If $m > n$ for $\underline{A} \in \mathbb{R}^{m \times n}$, then

Solving $\underline{R}\, \underline{x} = \underline{Q}^T \underline{b}$ is the solution

that minimizes the error.

Other advantage: Solving $\underline{R}\, \underline{x} = \underline{Q}^T \underline{b}$

is <u>much</u> more stable than

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

Classical Gram-Schmidt

One algorithm for reduced QR of $\underline{A}$.

Let $\underline{A} = [\underline{a}_1 \ \underline{a}_2 \ \cdots \ \underline{a}_n]$

Recall that $q_1 = \dfrac{\underline{a}_1}{\|\underline{a}_1\|}$

$$q_2 = \frac{\underline{a}_2 - r_{12} q_1}{r_{22}}$$

etc.,

where $r_{ij} = q_i^T \underline{a}_j$ for $i \neq j$

and $|r_{jj}| = \|\underline{a}_j - \sum_{i=1}^{j-1} r_{ij} q_i\|_2$

r_{jj} can be either + or −, choose +

Algorithm: Classical G-S

$[\underline{a}_1 \ \underline{a}_2 \ \cdots]$

for $j = 1:n$
  $\underline{v}_j = \underline{a}_j$
  for $i = 1:j-1$
    $r_{ij} = q_i^T \underline{a}_j$
    $\underline{v}_j = \underline{v}_j - r_{ij} q_i$
  $r_{jj} = \|\underline{v}_j\|_2$
  $q_j = \underline{v}_j / r_{jj}$

It turns out that Classical G-S is not numerically stable

$\Rightarrow$ Round-off errors cause issues.

$\rightarrow$ Will not prove, This would require complicated stability & error analysis,

$\rightarrow$ You will have a HW problem showing this,

We need a better method!

Modified Gram-Schmidt

Op Count for G-S

Most expensive operation is $r_{ij} = q_i^T \underline{v}_j$

& $\underline{v}_j = \underline{v}_j - r_{ij} q_i$

$$\Rightarrow \sum_{i=1}^{n} \sum_{j=i+1}^{n} 4m \sim \sum_{i=1}^{n} i\,4m \sim 2mn^2$$