Lecture 18

02/22/2023

Proof of the claim/derivation for normal equation

Proof using calculus:

Let $f(\underline{x}) := \| A\underline{x} - \underline{b} \|_2^2$

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

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

$$= \underline{x}^T A^T A \underline{x} \underbrace{- \underline{x}^T A^T \underline{b} - \underline{b}^T A^T \underline{x}} + \underline{b}^T \underline{b}$$

$$= \underline{x}^T A^T A \underline{x} - 2\, \underline{b}^T A \underline{x} + \underline{b}^T \underline{b}$$

The minimizer of $f(\underline{x})$ at $\underline{x} = \hat{\underline{x}}$ must satisfy:

$$\nabla_{\underline{x}} f(\underline{x}) \Big|_{\underline{x} = \hat{\underline{x}}} = \underline{0}$$

Uses calculus rules for taking derivative of scalar with respect to vector:

$$\frac{\partial}{\partial \underline{v}} \left(\underline{v}^T M \underline{v} \right) = \left(M + M^T \right) \underline{v}$$

$$\frac{\partial}{\partial \underline{v}} \left(\underline{n}^T \underline{v} \right) = \underline{n}$$

$$\Rightarrow 2A^T A \hat{\underline{x}} - 2A^T \underline{b} = \underline{0}$$

$$\Rightarrow \boxed{\left(A^T A \right) \hat{\underline{x}} = A^T \underline{b}}$$

the normal equation as claimed in Lec. 17, p. 6

(Proved.)

Proof without calculus:

Suppose $\underline{\hat{x}}$ solves the normal equation:

$$A^T A \underline{\hat{x}} = A^T \underline{b}$$

Choose any arbitrary $\underline{x} \neq \underline{\hat{x}}$

we will show: $\|A\underline{x} - \underline{b}\|_2^2 > \|A\underline{\hat{x}} - \underline{b}\|_2^2$

Now, $$\|A\underline{x} - \underline{b}\|_2^2 = \|\underbrace{A(\underline{x} - \underline{\hat{x}})}_{\underline{u}} + \underbrace{(A\underline{\hat{x}} - \underline{b})}_{\underline{v}}\|_2^2$$

$$= \|\underline{u} + \underline{v}\|_2^2$$

$$= \|\underline{u}\|_2^2 + \|\underline{v}\|_2^2 + \underbrace{2\underline{u}^T\underline{v}}_{\text{cross - term}}$$

But for the cross-term, notice that

$$\underline{u}^T \underline{v} = \left(A(\underline{x} - \hat{\underline{x}})\right)^T (A\hat{\underline{x}} - \underline{b})$$

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

$$= (\underline{x} - \hat{\underline{x}})^T \underbrace{(A^T A \hat{\underline{x}} - A^T \underline{b})}_{= 0 \text{ (thanks to normal equation)}}$$

$$= 0$$

$$\therefore \|A\underline{x} - \underline{b}\|_2^2 = \underbrace{\|A(\underline{x} - \hat{\underline{x}})\|_2^2}_{> 0 \text{ since } \underline{x} \neq \hat{\underline{x}}} + \|A\hat{\underline{x}} - \underline{b}\|_2^2$$

$$\therefore \|A\underline{x} - \underline{b}\|_2^2 > \|A\hat{\underline{x}} - \underline{b}\|_2^2 .$$

(Proved.)

Algorithm to compute the least sq. sol^{n}.

$$\hat{\underline{x}} = A^{+}\underline{b}$$

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

assuming A has linearly independent columns

If A has linearly indep. columns, then:

$$A = QR \longleftarrow \boxed{QR \text{ decomposition of } A}$$

square upper triangular matrix with >0 entries along main diagonal

has orthonormal columns

$$Q^TQ = I$$

Now,

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

$$= \left((QR)^T QR\right)^{-1} (QR)^T \underline{b}$$

$$= \left(R^T \underbrace{Q^T Q}_{I} R\right)^{-1} R^T Q^T \underline{b}$$

$$= (R^T R)^{-1} R^T Q^T \underline{b}$$

$$= R^{-1} \underbrace{R^{-T} R^T}_{I} Q^T \underline{b}$$

$$= R^{-1} Q^T \underline{b}$$

$$\Leftrightarrow \boxed{R \underline{\hat{x}} = Q^T \underline{b}}$$

← square linear system but different from the normal equation

Facts: QR factorization/decomposition complexity for any tall $\underset{\substack{m\times n\\ m>n}}{\underbrace{A}}$ is $O(mn^2)$

$$\gg [Q,R] = qr(A)$$

So to compute $\hat{\underline{x}}$, we can simply:

- do QR decomposition of A
- Then compute matrix-vector product: $\underline{c} = Q^T\underline{b}$
- Then solve the upper triangular square linear system: $R\hat{\underline{x}} = \underline{c} \iff \hat{\underline{x}} = R^{-1}\underline{c} = R^{-1}Q^T\underline{b}$

$\hookrightarrow$ complexity is $O(n^2)$

$\therefore$ Overall (worst-case) complexity for computing $\hat{\underline{x}}$:

$O(mn^2)$.

Example: Compute $\hat{\underline{x}}$ for

$$\underbrace{\begin{bmatrix} 3 & -6 \\ 4 & -8 \\ 0 & 1 \end{bmatrix}}_{A} \underbrace{\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}}_{\underline{x}} = \underbrace{\begin{pmatrix} -1 \\ 7 \\ 2 \end{pmatrix}}_{\underline{b}}$$

- QR factorization of A:

$$Q = \begin{bmatrix} 3/5 & 0 \\ 4/5 & 0 \\ 0 & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 5 & -10 \\ 0 & 1 \end{bmatrix}$$

- Compute $\underline{c} = Q^T \underline{b}$

$$= \begin{bmatrix} 3/5 & 4/5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} -1 \\ 7 \\ 2 \end{pmatrix}$$

$$= \begin{pmatrix} 5 \\ 2 \end{pmatrix}$$

- Solve $R\hat{\underline{x}} = \underline{c}$ via back substitution:

$$\begin{bmatrix} 5 & -10 \\ 0 & 1 \end{bmatrix} \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} = \begin{pmatrix} 5 \\ 2 \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \end{pmatrix} = \begin{pmatrix} 5 \\ 2 \end{pmatrix}.$$

Done.

• <u>Example</u>: (Round-off error in least sq. sol$^{\underline{n}}$)

$$A = \begin{bmatrix} 1 & -1 \\ 0 & 10^{-5} \\ 0 & 0 \end{bmatrix}, \quad \underline{b} = \begin{pmatrix} 0 \\ 10^{-5} \\ 1 \end{pmatrix}$$

Suppose we round-off to 8 significant digits :

<u>Approach 1</u> : Construct the Gram matrix $A^T A$ and directly solve the normal equation

$$A^T A = \begin{bmatrix} 1 & -1 \\ -1 & 1+10^{-10} \end{bmatrix} \rightsquigarrow \underbrace{\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}}_{\text{singular matrix !!}}$$

rounding error

Approach 2 : $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$, $R = \begin{bmatrix} 1 & -1 \\ 0 & 10^{-5} \end{bmatrix}$

unaffected by round-off error.

So, QR factorization is the standard approach for computing $\hat{x}$.

In MATLAB, compute $\hat{x}$ as :

$$\gg \underbrace{A \backslash b}_{}$$

same command as solving square linear system but different math & different algorithm

Least squares for solving regression / function approximation / model fitting:

Given dataset $(\underline{x}, y)$, we want to compute:

$$y \approx f(\underline{x})$$

output variable

feature / explanatory variable

$\Longleftrightarrow$ We want to approximate the truth "f" by a model "$\hat{f}$" based on data

Then the prediction from our model:

$$\hat{y} = \hat{f}(x)$$

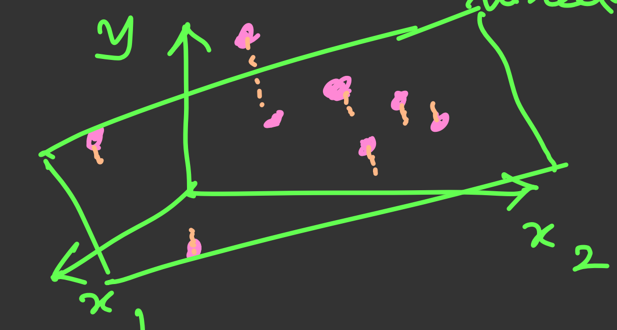
However, reality / truth : $y = f(x)$

$\therefore$ Prediction error $(r) := y - \hat{y}$

Example: (Linear regression)

Linear $\leftarrow$ $\hat{f}$ is linear

$\Leftrightarrow$ linear function approximation



n×1 vector   scalar

x	y
$\underline{x}_1$	y_1
$\underline{x}_2$	y_2
$\vdots$	
$\underline{x}_N$	y_N

$$\hat{f}(\underline{x}) = \underline{x}^T \underline{\beta} + v, \quad \underline{\beta} \in \mathbb{R}^n$$

$$v \in \mathbb{R}$$

parameters

Prediction error @ the i^{th} data sample: $(\underline{x}^{(i)}, y^{(i)})$

$$
\begin{aligned}
r^{(i)} &= y^{(i)} - \hat{f}(\underline{x}^{(i)}) \\
&= y^{(i)} - (\underline{x}^{(i)})^T \underline{\beta} - v
\end{aligned}
$$

Problem: Compute the parameters $\underline{\beta}$ and v such that the mean square error (MSE) is minimized:

$$
MSE = \frac{1}{N} \sum_{i=1}^{N} \left(r^{(i)}\right)^2,
$$

where N is the number of data samples