MCEN 3030

5 Mar 2024

HW #5 Released Tonight

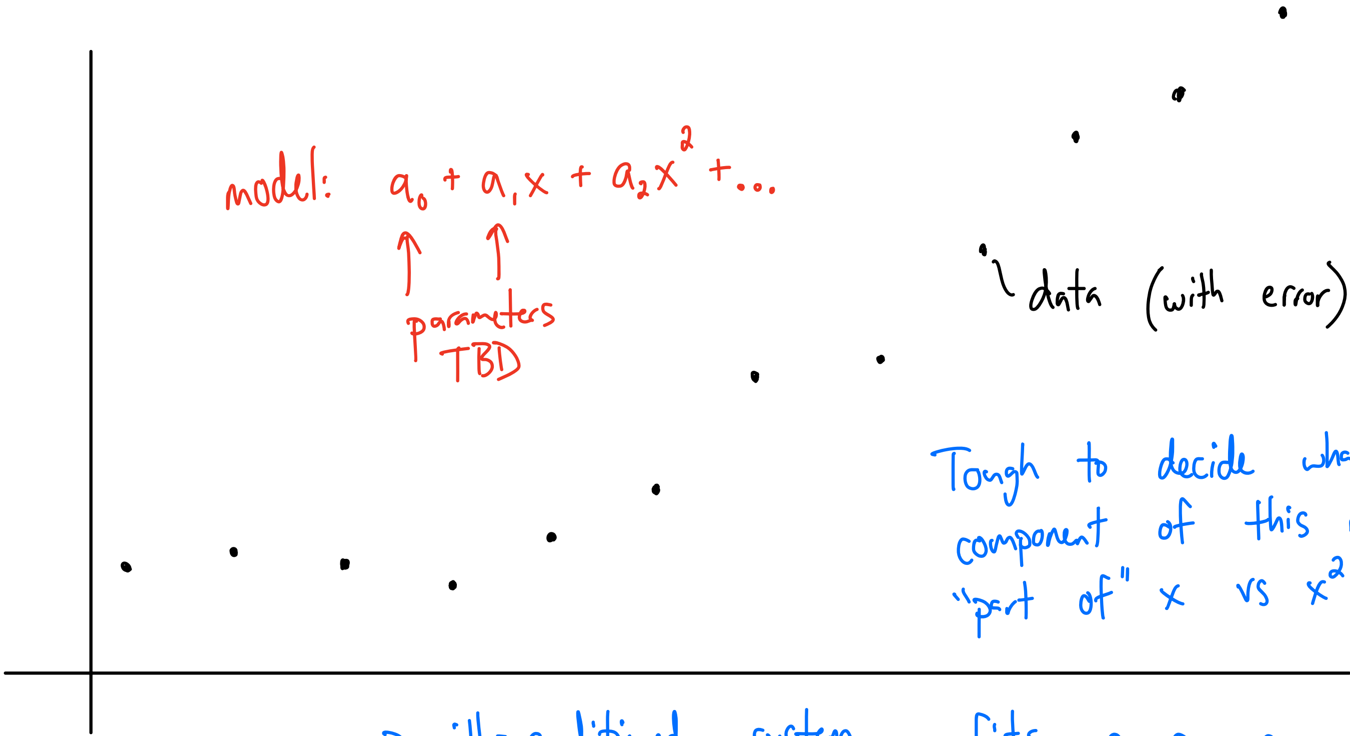
Previously: 2D Newton-Raphson
Modeling

Today: QR decomposition

how MATLAB does
A\b

A couple weeks ago

Modeling



model: $a_0 + a_1 x + a_2 x^2 + \ldots$

parameters TBD

data (with error)

Tough to decide what component of this is "part of" x vs x^2 etc

$\Longrightarrow$ ill-conditioned system, fits a_0, a_1, a_2 may be sensitive to error.

The QR Decomposition (more stable method of fitting data)

With model fitting we had something like this:

$$Z^T Z A = Z^T Y$$

$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}$ m data points (known)

$\begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}$ n parameters (unknown)

$Z = \begin{bmatrix} 1 & x_1 & \cdots \\ 1 & x_2 & \cdots \\ \vdots & \vdots & \ddots \\ 1 & x_m & \end{bmatrix}$ data (known)

Ostensibly: $A = (Z^T Z) \backslash (Z^T Y)$

this may be "ill-conditioned"

→ we bump up against num error

$a_0 + a_1 x + a_2 x^2 + \ldots$

Theorem: Every real $m \times n$ matrix can be decomposed into $Z = QR$ where Q is an $m \times m$ orthogonal matrix and R is an $m \times n$ upper triangular matrix.

$$\begin{bmatrix} \\ Z \\ \\ \end{bmatrix} = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ q_1 & q_2 & \cdots & q_m \\ \downarrow & \downarrow & & \downarrow \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & & \\ 0 & 0 & \ddots & \\ & & & r_{nn} \\ 0 & 0 & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & & \cdots & 0 \end{bmatrix}$$

Orthogonal Matrices have special properties: (orthonormal)

1) Their columns are orthogonal vectors

e.g. if one is $\langle 1, 0, 0 \rangle$,

another might be $\langle 0, 1, 0 \rangle$

$$Q^T Q = \begin{bmatrix} \leftarrow & q_1^T & \rightarrow \\ \leftarrow & q_2^T & \rightarrow \\ & \vdots & \\ \leftarrow & q_m^T & \rightarrow \end{bmatrix} \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ q_1 & q_2 & \cdots & q_m \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}$$

$$= \begin{bmatrix} q_1^T q_1 & q_1^T q_2 & \cdots & \\ q_2^T q_1 & \ddots & & \\ \vdots & & & \\ & & & q_m^T q_m \end{bmatrix} = I_{m \times m}$$

$\therefore \; Q^T = Q^{-1} \implies$ so easy to determine Q^{-1}

2) Q^T is also orthogonal

3) The condition number of Q is 1

The factorization process

(copied from above:)

$$\begin{bmatrix} \\ & Z & \\ \\ \end{bmatrix} = \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ q_1 & q_2 & \cdots & q_m \\ \downarrow & \downarrow & & \downarrow \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & & \\ 0 & 0 & \ddots & \\ & & & r_{nn} \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

the vectors $q_{n+1} \longrightarrow q_m$ end up not contributing to calculation of Z

$$\downarrow$$

$$= \begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ q_1 & q_2 & \cdots & q_n \\ \downarrow & \downarrow & & \downarrow \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & \cdots & r_{1n} \\ 0 & r_{22} & & & \cdots \\ 0 & 0 & \ddots & & \\ \vdots & \vdots & & \ddots & \\ \vdots & \vdots & & & r_{nn} \end{bmatrix}$$

$$Z = \hat{Q} \cdot \hat{R}$$

OK now into the least-squares regression equation

$$Z^T Z A = Z^T Y$$

$$(\hat{Q}\hat{R})^T (\hat{Q}\hat{R}) A = (\hat{Q}\hat{R})^T Y$$

$$\hat{R}^T \underbrace{\hat{Q}^T \hat{Q}}_{I} \hat{R} A = \hat{R}^T \hat{Q}^T Y$$

$$\hat{R}^T \hat{R} A = \hat{R}^T \hat{Q}^T Y \longrightarrow \hat{R} A = \hat{Q}^T Y$$

$\therefore$ Once we have $\hat{Q}^T$ generated (e.g. Gram-Schmidt or Householder), the best fits A can be determined from a back-sub algorithm (since $\hat{R}$ = upper triangular)

Q: Why couldn't we just do this?

$$Z^T Z A = Z^T Y \longrightarrow Z A = Y$$

$\Rightarrow$ Z^T is not square

Q: Why do we care?

We were using $Z^T Z A = Z^T Y$.

Z is often "ill-conditioned", which means Z^T is too. Ostensibly this means multiplying by $(Z^T Z)^{-1}$ leads to really big numbers, where we have errors associated with storing the full detail of the numbers. Then, we immediately undo some of this "bigness" with the multiplication by Z^T, but the damage has been done! And, as mentioned above,

$$Z^T Z A = Z^T Y \not\Rightarrow Z A = Y.$$

With QR decomposition, we write

$$\hat{R}^T \underbrace{\hat{Q}^T \hat{Q}}_{= I} \hat{R} A = \hat{R}^T \hat{Q}^T Y \implies \hat{R} A = \hat{Q}^T Y$$

If Z is ill-conditioned, $\hat{R}$ will be too. But! We were able to cancel out $\hat{R}^T$, because it is square. So there is only one ill-conditioned bit, not two. $(\text{ill-conditioned})^2 \to (\text{ill-conditioned})$, and so we are less inclined to hit numerical error limits.