CS 450: Numerical Anlaysis

Lecture 6
Chapter 3 – Linear Least Squares
QR Factorization

Edgar Solomonik

Department of Computer Science
University of Illinois at Urbana-Champaign

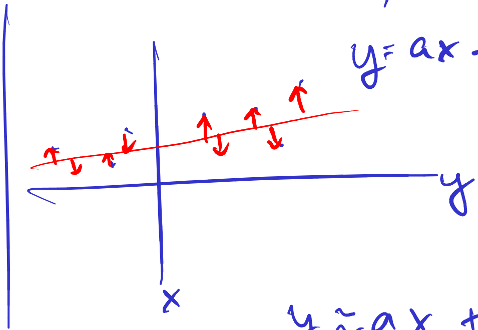
Linear Least Squares Motivation

$Ax \cong b$ overdetermined, minimize $\|Ax - b\|_2$

(x_1, y_1)

$\vdots$

(x_m, y_m)



$y = ax + b$

$y_1 \approx a x_1 + b$

$$\begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & \vdots \\ 1 & \vdots \\ 1 & x_m \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} \cong \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

Linear Least Squares

residual

- Find $\boldsymbol{x}^\star = \operatorname{argmin}_{\boldsymbol{x}\in\mathbb{R}^n} ||\boldsymbol{A}\boldsymbol{x}-\boldsymbol{b}||_2$ where $\boldsymbol{A}\in\mathbb{R}^{m\times n}$:

$$x^* = \operatorname{argmin}_{x\in\mathbb{R}^n} \underbrace{\langle Ax-b,\ Ax-b\rangle}_{||Ax-b||^2}$$

$$\Sigma = \begin{bmatrix} \sigma_1 \\ 0 \end{bmatrix} \qquad \Sigma^\dagger = \begin{bmatrix} \sigma_1^{-1} \\ 0 \end{bmatrix}$$

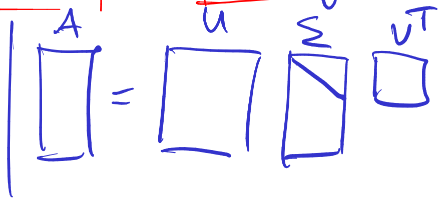
pseudoinverse

- Given the SVD $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ we have $\boldsymbol{x}^\star = \boxed{\boldsymbol{V}\boldsymbol{\Sigma}^\dagger\boldsymbol{U}^T}\boldsymbol{b}$, where $\boldsymbol{\Sigma}^\dagger$ contains the reciprocal of all nonzeros in $\boldsymbol{\Sigma}$:

$A\in\mathbb{R}^{m\times n}$

$$\boxed{U\Sigma \underbrace{V^T x}_{y} \cong b} \iff \boxed{U\Sigma y \cong b} \iff \boxed{\Sigma y \cong \underbrace{U^T b}_{c}}$$

$$\Sigma y = c$$



$$\underset{\Sigma}{\begin{bmatrix} \diagdown \\ 0 \end{bmatrix}} \begin{bmatrix} y \end{bmatrix} = \underset{c}{\begin{bmatrix} c_1 \\ c_2 \end{bmatrix}}$$

$\Sigma_1 y = c_1$

$y = \Sigma_1^{-1} c_1$

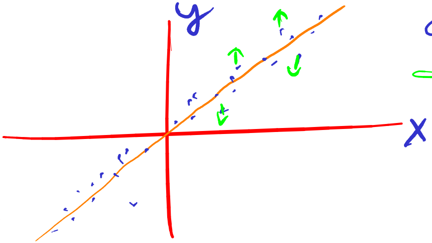
$$\begin{bmatrix}\sigma_1 & & & \\ & \sigma_2 & & \\ & & \sigma_3 & \\ & & & \sigma_4 \\ & & & \\ & & & \\ & & & \end{bmatrix}\begin{bmatrix}y_1\\y_2\\y_3\\y_4\end{bmatrix} = \underbrace{\begin{bmatrix}c_1\\c_2\\c_3\\c_4\\0\\0\\0\end{bmatrix} \approx \begin{bmatrix}c_1\\c_2\\c_3\\c_4\\c_5\\c_6\\c_7\end{bmatrix}}_{\text{diff. is residual}}$$

$$y_i = c_i/\sigma_i$$

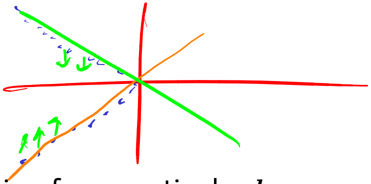
Conditioning of Linear Least Squares

- Consider fitting a line to a collection of points, then perturbing the points:

$$\begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix} \begin{bmatrix} c \end{bmatrix} \cong \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$



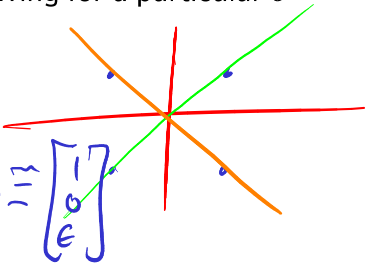
conditioning w.r.t. y



- LLS is ill-posed for any A, unless we consider solving for a particular b

for all possible b in $Ax \cong b$

LLS is ill posed



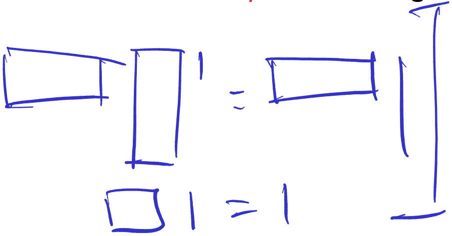
$$\begin{bmatrix} \epsilon, \\ \cancel{0} & \cancel{0} \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \cong \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \qquad x^* = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\Longrightarrow \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ \epsilon \end{bmatrix} \cong \begin{bmatrix} 1 \\ 0 \\ \epsilon \end{bmatrix}$$

Normal Equations

$A = U\Sigma V^T$

- *Normal equations* are given by solving $\boldsymbol{A}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{A}^T\boldsymbol{b}$:



$$V\Sigma^T \underbrace{U^T U}_{I} \Sigma V^T x = V\Sigma^T U^T b$$

$$\cancel{V}\Sigma^T \Sigma V^T x = \cancel{V}\Sigma^T U^T b$$

- However, solving the normal equations is a more ill-conditioned problem then the original least squares algorithm

$$A^T A = V\Sigma^T\Sigma V^T = \underbrace{[\quad]}_{\text{ortho}} \begin{bmatrix} \sigma_{max}^2 & & \\ & \ddots & \\ & & \sigma_{min}^2 \end{bmatrix} [\quad]$$

$$\kappa(A^T A) = \kappa(A)^2 = \frac{\sigma_{max}^2}{\sigma_{min}^2}$$

A is $n+1 \times n$

$$\begin{bmatrix} A_1 \\ 0 \end{bmatrix} [\quad] = \underbrace{\begin{bmatrix} \vdots \\ 0 \end{bmatrix}}_{\text{ortho}}$$

Solving the Normal Equations

- If $\boldsymbol{A}$ is full-rank, then $\boldsymbol{A}^T\boldsymbol{A}$ is symmetric positive definite (SPD):

e.g. positive

$$A^TA = V\Sigma^2V^T$$

rows eigenvectors

eigenvalue decomposition

eigenvalues

$$(A^TA)^T = A^TA$$

$$\underbrace{A^TA}_{V\Sigma^2V^T}V = V\Sigma^2$$

- Since $\boldsymbol{A}^T\boldsymbol{A}$ is SPD we can use Cholesky factorization, to factorize it and solve linear systems:

$$A^TA = LL^T$$

not unit-diagonal

QR Factorization

- If A is full-rank there exists an orthogonal matrix Q and a unique upper-triangular matrix R with a positive diagonal such that $A = QR$

$$\boxed{A^T A = LL^T} = R_1^T R_1$$

$$AR_1^{-1} = AL^{-T} = Q_1$$

$$\underbrace{L^{-1}A^T}_{Q^T}\underbrace{AL^{-T}}_{Q} = I$$

$A = QR$

orth. upper tri full

$$[A] = [Q_1 \mid \;]\begin{bmatrix} R_1 \\ 0 \end{bmatrix} = [Q_1][R_1]$$

- A reduced QR factorization (unique part of general QR) is defined so that $Q \in \mathbb{R}^{m\times n}$ has orthonormal columns and R is square and upper-triangular

orthonormal columns

Cholesky-QR

Gram-Schmidt Orthogonalization

- **Classical Gram-Schmidt process for QR:**

$$r_1 = \|a_1\|$$
$$r_i = \|b_i\|$$

$$A = \begin{bmatrix} a_1 \cdots a_n \end{bmatrix} = \begin{bmatrix} q_1 \cdots q_n \end{bmatrix} \begin{bmatrix} R \end{bmatrix}$$

$$q_1 = \frac{a_1}{\|a_1\|}, \quad q_i = \frac{b_i}{\|b_i\|}, \quad b_i = a_i - \sum_{j=1}^{i-1} \underbrace{\langle q_j, a_i \rangle}_{r_{ji}} q_j$$

- **Modified Gram-Schmidt process for QR:**

$$b_i = MGS(a_i, i-1), \quad MGS(x, j) = MGS(x - \langle q_j, x \rangle q_j, j-1)$$

$$MGS(x, 0) = x$$

$$b_i = b_i^2 - \left\langle q_3, \underbrace{\left(b_i^1 - \left\langle q_2, \underbrace{\left(a_i - \langle q_1, a_i \rangle q_1 \right)}_{b_i^1} \right\rangle q_2 \right)}_{b_i^2} \right\rangle q_3$$

b_4

$$q_i = \frac{b_i^1}{\|b_i\|}$$

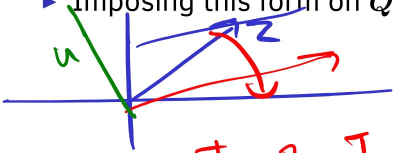
Householder QR Factorization

- **A Householder transformation $Q = I - 2uu^T$ is an orthogonal matrix defined to annihilate entries of a given vector z, so $||z||_2 Q e_1 = z$:**

$$Q^T z = \begin{bmatrix} \|z\|_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = e_1 \|z\|_2$$

$$\begin{bmatrix} a_1 & \cdots \end{bmatrix} \xRightarrow{Q^T} \begin{bmatrix} r_{11} & \cdots \end{bmatrix}$$

- Imposing this form on Q leaves exactly two choices for u given z,



$$Q = I - 2uu^T$$

$$u = \frac{z \pm ||z||_2 e_1}{||z \pm ||z||_2 e_1||_2}$$

$$\begin{bmatrix} Q^T a_2 & Q^T a_3 & \cdots \end{bmatrix}$$

Applying Householder Transformations

- The product $\boldsymbol{x} = \boldsymbol{Q}\boldsymbol{w}$ can be computed using $O(n)$ operations if $\boldsymbol{Q}$ is a Householder transformation

$$Q = I - 2uu^T$$

$$Q^T v = (I - 2uu^T)v = v - 2\langle u, v\rangle u \qquad O(n)$$

- Householder transformations are also called *reflectors* because their application reflects a vector along a hyperplane (changes sign of component of $\boldsymbol{w}$ that is parallel to $\boldsymbol{u}$)

$$z - \underbrace{\langle q_1, z\rangle}_{q_1^T z} q_1 = (I - \underbrace{q_1 q_1^T}) z$$

projection (G-S)

$z - w$

$$(I - 2uu^T)z = z - 2u\langle u, z\rangle$$

$$\underbrace{uu^T z}_{w}$$

Hh. $z - 2w$