$$\begin{bmatrix} a_{11} & \cdots & a_{13} \\ \vdots & & \vdots \\ a_{31} & \cdots & a_{33} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_3 \end{bmatrix}$$

$a_{11} x_1 + a_{12} x_2 + a_{13} x_3 = b_1$ — coordinates

Basis: a tuple of vector s.t. we can write every vector $\vec{b}$ as a linear combination of them

$$v = \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix} \quad w = \begin{bmatrix} w_1 \\ \vdots \\ w_m \end{bmatrix}$$

$$v \cdot w \equiv v^T w$$

$$v_1 \cdot w_1 + \dots + v_m \cdot w_m$$

$$v \cdot w = \|v\| \cdot \|w\| \cdot \cos\theta$$

orthogonal if $v \cdot w = 0$

Map $x \longrightarrow Ax$ geometric transformation: rotation, scaling, symmetric.

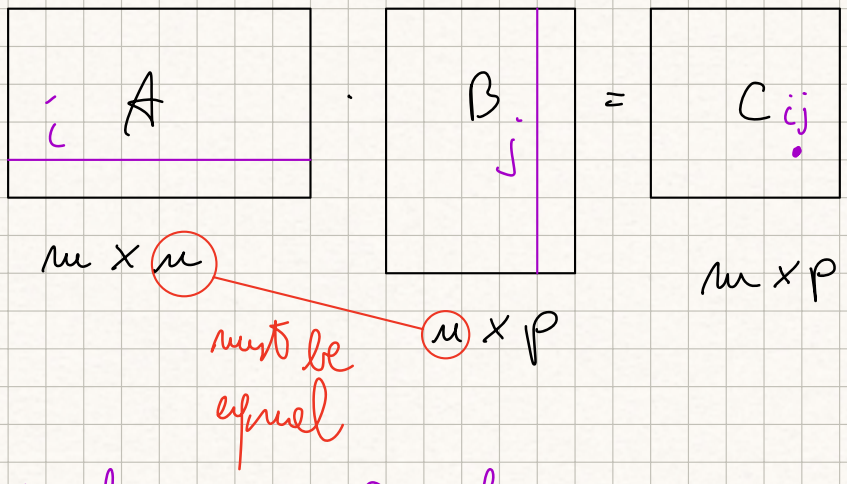
$$A \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 + x_2 \\ x_2 \end{bmatrix}$$

$I = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}$ $I \cdot v = v$ identity matrix

$v \longrightarrow A(Bv)$ composition is a transformation that corresponds to the matrix product AB

$$v \longrightarrow A(Bv) = (AB)v$$

Matrix product:



$$c_{ij} = a_{i1} \cdot b_{1j} + a_{i2} b_{2j} + \dots + a_{in} b_{nj}$$

vectors = $n \times 1$ matrices

Matrix algebra:

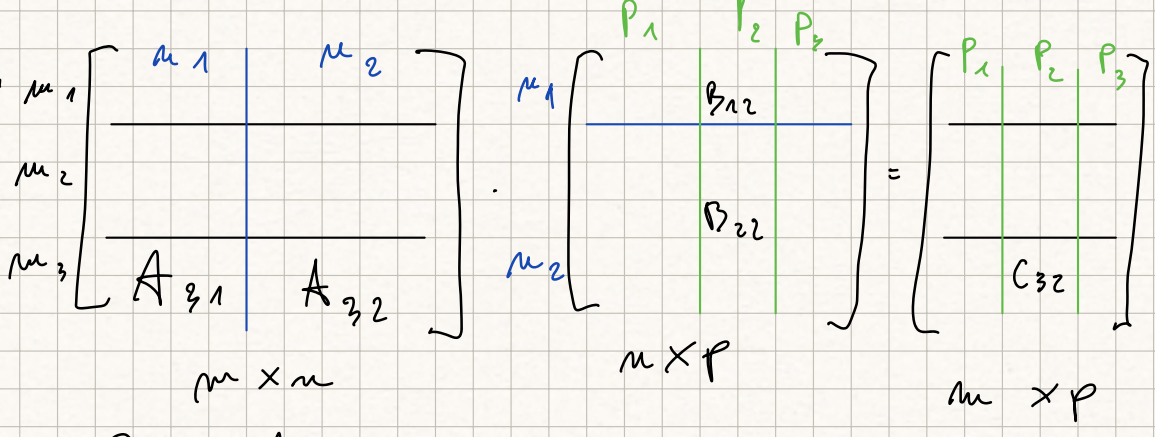
- $A(B+C) = AB + AC$
- $(AB)C = A(BC)$
- $(A+B)^2 = A^2 + B^2 + AB + BA$ $\quad AB \neq BA$

$AB = 0 \not\Rightarrow B = 0$ or $A = 0$

$AB = AC \not\Rightarrow B = C$

Rank of a matrix A: is the smallest number such that all the columns of A can be written as linear combinations of n vectors $v_1, v_2, \dots, v_n$

Block operation:



$$C_{32} = A_{31} \cdot B_{12} + A_{32} \cdot B_{22}$$

Linear systems: given $b \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$, which are the coordinates, I need to write b as a linear combination of the columns of A.

$$b = \begin{bmatrix} 4 \\ 4 \\ 0 \\ 0 \end{bmatrix} \quad A = \begin{bmatrix} 2 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad Ax = \begin{bmatrix} 2 \\ 0 \\ 0 \\ 0 \end{bmatrix} x_1 + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} x_2 + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} x_3 = b$$

If solution not available, we can get close to it solving

$$\min \|Ax - b\|$$

If A is square $(n \times n)$ and invertible: $x = A^{-1}b, \forall\, b \in \mathbb{R}^n$

$$A^{-1} \cdot A = A \cdot A^{-1} = I = \begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix}$$

Orthogonality: $U \in \mathbb{R}^{n \times n}$ is orthogonal if $U^T U = I$ / $U U^T = I$

$$U^{-1} = U^T$$

If U is orthogonal, then $\forall\, v \in \mathbb{R}^n \quad \|Uv\| = \|v\|$

$$\|v\|^2 = v^T v = v^T \cdot I \cdot v = v^T (U^T U)\, v = (Uv)^T (Uv) = \|Uv\|^2$$

$$\|x\|_A = x^T A x$$

Def: $\{u_1, u_2, \dots, u_n\}$ is orthonormal if $u_i^T u_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$

$$I = U^T U$$

$$\begin{bmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{bmatrix} = \begin{bmatrix} u_1^T \\ \hline u_2^T \\ \hline \vdots \\ \hline u_n^T \end{bmatrix} \cdot \left[\, u_1 \,\middle|\, u_2 \,\middle|\, \cdots \,\middle|\, u_n \,\right] = \begin{bmatrix} u_1^T u_1 & u_1^T u_2 & \cdots \\ & u_2^T u_2 & \\ & & \ddots \\ & & u_n^T u_n \end{bmatrix}$$

Def: U_1, U_2 orthogonal $\rightarrow$ $U_1 U_2$ is orthogonal

$$(U_1 \cdot U_2)^T (U_1 \cdot U_2) = U_2^T \cdot \underbrace{U_1^T \cdot U_1}_{I} \cdot U_2 = \underbrace{U_2^T \cdot U_2}_{I} = I$$

Eigenvalues / Eigenvectors

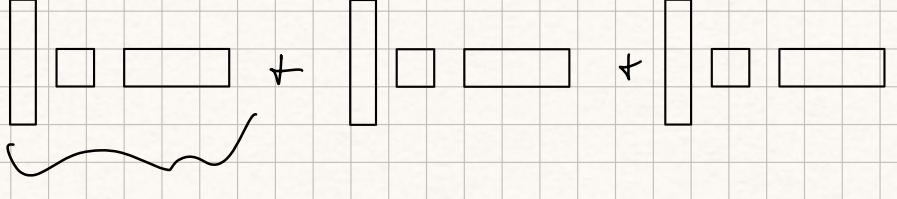
Given a square matrix $A \in \mathbb{R}^{m \times m}$ if $Av = v\lambda$ for a certain vector $v \in \mathbb{R}^m$, $\lambda \in \mathbb{R}$, then we say that v is an eigenvector and λ is an eigenvalue of A

For almost all matrices:

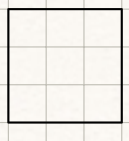
$$A = V \Lambda V^{-1} = \underbrace{\begin{bmatrix} v_1 & v_2 & \cdots & v_m \end{bmatrix}}_{\text{eigenvectors of } A} \underbrace{\begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_m \end{bmatrix}}_{\text{eigenvalues of } A} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix} =$$

$$A v_1 = v_1 \lambda_1, \quad A v_2 = v_2 \lambda_2, \ldots, \quad A v_m = v_m \lambda_m$$

$$= v_1 \lambda_1 w_1 + v_2 \lambda_2 w_2 + \cdots + v_m \lambda_m w_m$$



sum of m rank-1 terms



$[V, \Lambda] = \text{eig}(A)$ cost $O(m^3)$

Eigenvectors are even less unique: if v eigenvector, αv is again an eigenvector (with the same λ)

If v_1, v_2 eigenvectors with $\lambda_1 = \lambda_2 = \lambda$, then $\alpha v_1 + \beta v_2$ is again an eigenvector with eigenvalue λ

$$A(v_1\alpha + v_2\beta) = Av_1\alpha + Av_2\beta = v_1\lambda\alpha + v_2\lambda\beta = (v_1\alpha + v_2\beta)\lambda$$

Th (Spectral theorem) If $A = A^T$ (symmetric), then $A = V\Lambda V^{-1}$ always exists, the eigenvalues λ_i are real, V can be taken orthogonal

Th: Let Q be symmetric with minimum eigenvalue λ_{min} and maximum λ_{max}. Then, $\lambda_{min}\|x\|^2 \leq x^T Q x \leq \lambda_{max}\|x\|^2$

Proof: Let us first do the case $Q = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{bmatrix}$, then

$$\begin{bmatrix} x_1 & x_2 & \cdots & x_m \end{bmatrix} \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_m \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix} = \begin{bmatrix} x_1 & \cdots & x_m \end{bmatrix} \begin{bmatrix} \lambda_1 x_1 \\ \vdots \\ \lambda_m x_m \end{bmatrix}$$

$$\lambda_{min}\left(x_1^2 + \cdots + x_m^2\right) \leq \lambda_1 x_1^2 + \lambda_2 x_2^2 + \cdots + \lambda_m x_m^2 \leq \lambda_{max}\left(x_1^2 + \cdots + x_m^2\right)$$

Def: if $\lambda_i \geq 0$ for each eigenvalue λ_i of Q, then $x^T Q x \geq 0$. In this case we call Q positive semidefinite

Def: if $\lambda_i > 0$ for each eigenvalue λ_i of Q, then $x^T Q x > 0$ for each vector $x \neq 0$. In this case, we call Q positive definite.

$\downarrow$

Q is invertible

Th: Let $A \in \mathbb{R}^{m \times n}$. Then, $A^T A$ is symmetric, positive semidefinite matrix. It is also valid for $A \cdot A^T$

proof: $(A^T A)^T = A^T \cdot (A^T)^T = A^T A$, so the product is symm.

for each $x \in \mathbb{R}^n$

$$x^T (A^T A) x = (Ax)^T (Ax) = \|Ax\|^2 \geq 0$$

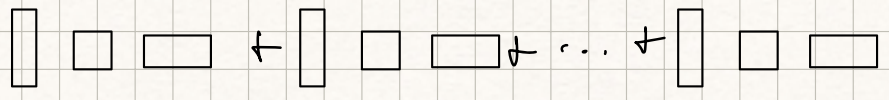
Singular value decomposition $A \in \mathbb{R}^{m \times m}$, each square matrix can be written as:

$$A = U \Sigma V^T = \left[u_1 \middle| u_2 \middle| \cdots \middle| u_m \right] \cdot \begin{bmatrix} \sigma_1 & & & 0 \\ & \sigma_2 & & \\ & & \ddots & \\ 0 & & & \sigma_m \end{bmatrix} \cdot \begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_m^T \end{bmatrix}$$

with U orthogonal, Σ diagonal, V orthogonal

and $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \dots \geq \sigma_m \geq 0$ singular values

$$= u_1 \sigma_1 v_1^T + u_2 \sigma_2 v_2^T + \dots + u_m \sigma_m v_m^T$$

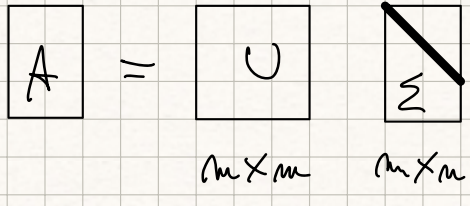


σ_i unique for each matrix, u_i and v_i may be not.

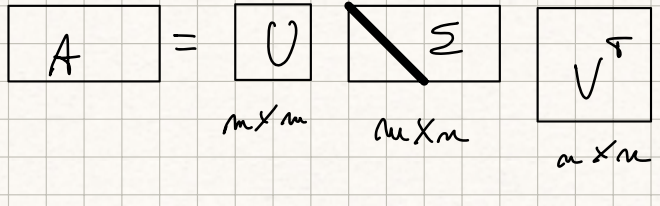
Th: If A rectangular $\in \mathbb{R}^{m \times n}$, it can always be decomposed as $A = U \Sigma V^T$ $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ orthogonal,

$\Sigma \in \mathbb{R}^{m \times n}$ diagonal (i.e. $\Sigma_{ij} = 0$ whenever $i \neq j$)

$m > n$ (A tall thin)



$n > m$ (A short fat)



$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \dots \sigma_{\min(m,n)} \geq 0$

$$A = \left[u_1 \middle| u_2 \middle| \dots \middle| u_q \right] \cdot \left[\begin{array}{ccc} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & \sigma_3 \\ \hline 0 & & \end{array} \right] \cdot \left[\begin{array}{c} v_1^T \\ \hline v_2^T \\ \hline v_3^T \end{array} \right]$$

if $m > n$, $u_{n+1} \dots u_m$ match up with zeros and do not

appear in the product. I can write

$$U = \underset{m \times m}{\left[\begin{array}{c|c} U_1 & U_2 \\ m & m-n \end{array}\right]} \qquad \Sigma = \left[\begin{array}{c} \Sigma \\ \hline 0 \end{array}\right]^{n}_{m-n} \qquad A = U\Sigma V^T = U_1 \Sigma_1 V_1^T$$

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

$$U\Sigma = U_1\Sigma_1 + \cancel{U_2 \cdot 0}$$

Properties of SVD

- $\text{rank}(A)$ = number of non-zero σ_i's:

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_{min} = 0$$

$$A = u_1 \sigma_1 v_1^T + \dots + u_r \sigma_r v_r^T + \cancel{u_{r+1} \sigma_{r+1} v_{r+1}^T} \dots$$

$\text{Image}(A) = \text{span}\{u_1, u_2, \dots, u_r\}$. Vectors are linearly independent column vector of A, and r is the rank of A, which is the dimension of the image.

$$\text{Ker}(A) = \{x \in \mathbb{R}^m \text{ s.t. } Ax = 0\}$$

When v_k, with $k > r$ $\quad Av_k = u_1 \sigma_1 \underbrace{v_1^T v_k}_{0} + \dots + u_r \sigma_r \underbrace{v_r^T v_k}_{0} = 0$

V is orthogonal

Hence $v_{r+1}, v_{r+2}, \dots v_n$ are in $\text{Ker}(A)$.

In fact, $\text{Ker}(A) = \text{span}\{v_{r+1}, v_{r+2}, \dots, v_n\}$

$A = U \Sigma V^T$ square invertible, SVD of A^{-1}:

$$A^{-1} = (V^T)^{-1} \Sigma^{-1} U^{-1} = V \begin{bmatrix} 1/\sigma_1 & & 0 \\ & \ddots & \\ 0 & & 1/\sigma_m \end{bmatrix} U^T = \begin{bmatrix} v_1 \mid v_2 \mid \dots \mid v_m \end{bmatrix} \Sigma^{-1} \begin{bmatrix} u_1^T \\ \hline \vdots \\ \hline u_m^T \end{bmatrix}$$

$$= v_1 \frac{1}{\sigma_1} u_1^T + v_2 \frac{1}{\sigma_2} u_2^T + \dots + v_m \frac{1}{\sigma_m} u_m^T$$

$$= \begin{bmatrix} v_m \mid v_{m-1} \mid \dots \mid v_1 \end{bmatrix} \begin{bmatrix} 1/\sigma_m & & \\ & 1/\sigma_{m-1} & \\ & & \ddots \; 1/\sigma_1 \end{bmatrix} \begin{bmatrix} u_m^T \\ \hline u_{m-1}^T \\ \hline \vdots \\ \hline u_1^T \end{bmatrix}$$

orth. diag. orth.

SVD of A^{-1}

Let $A = U \Sigma V^T$, eigenvalue decomposition of $A^T A$:

$$A^T A = (U \Sigma V^T)^T (U \Sigma V^T) = V \Sigma^T \underbrace{U^T U}_{I} \Sigma V^T$$

$$\Sigma^T \Sigma = \left[\begin{array}{ccc|c} \sigma_1 & & 0 & \\ & \ddots & & 0 \\ 0 & & \sigma_m & \end{array}\right] \left[\begin{array}{c} \sigma_1 \quad\quad 0 \\ \ddots \\ 0 \quad\quad \sigma_m \\ \hline 0 \end{array}\right] = \begin{bmatrix} \sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_m^2 \end{bmatrix}$$

$$A^T \cdot A = V \begin{bmatrix} \sigma_1^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_m^2 \end{bmatrix} V^T$$

Matrix norms: $\|x\| = (x^T x)^{1/2}$

def: given any norm $\|\cdot\|$ on vectors, we can define a norm on matrices as:

$$\|M\| = \max_{\substack{v \neq 0 \\ v \in \mathbb{R}^m}} \frac{\|Mv\|}{\|v\|} = \max_{\substack{u \in \mathbb{R}^m \\ \|u\|=1}} \|Mu\| \qquad v = \alpha u$$

scale, norm 1

- $\|Mv\| \leq \|M\| \cdot \|v\|$
- $\|A\| \geq 0$ equality only for $A = 0$
- $\|\alpha A\| = |\alpha| \cdot \|A\|$
- $\|A+B\| \leq \|A\| + \|B\|$
- $\|AB\| \leq \|A\| \cdot \|B\|$
- $\|Av\| \leq \|A\| \cdot \|v\|$

If we have $\|\cdot\| = \|\cdot\|_2$, then: $\|UA\|_2 = \|A\|_2$

$\|AU\|_2 = \|A\|$ with U orthogonal

If $A = U\Sigma V^T$, then $\|A\|_2 = \|U\Sigma V^T\|_2 = \|\Sigma\|_2 = \left\| \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_m \end{bmatrix} \right\|_2 =$

$\max(\sigma_1, \dots, \sigma_m) = \sigma_1$ why? $\left\| \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_m \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} \right\| = \left\| \begin{bmatrix} \sigma_1 u_1 \\ \vdots \\ \sigma_m u_m \end{bmatrix} \right\| =$

$$= \sqrt{\sigma_1^2 \mu_1^2 + \dots + \sigma_m^2 \mu_m^2} \leq \sqrt{\sigma_1^2 \underbrace{(\mu_1^2 + \dots + \mu_m^2)}_{1}} = \sigma_1$$

Frobenius norm: $\|A\|_F = \left(\sum_{i,j} a_{ij}^2 \right)^{1/2}$

satisfies all properties of norms, including $\|AU\|_F = \|UA\|_F = \|A\|_F$

but $\|I\|_2 = 1 \qquad \|I\|_F = \sqrt{m}$

Th: for both the Frobenius and the 2 norm, the matrix X that achieves $\min_{X \in \mathbb{R}^{m \times n}, \operatorname{rank} X \leq k} \|A - X\|$ is $X_k = \mu_1 \sigma_1 v_1^T + \dots + \mu_k \sigma_k v_k^T$

where $U, \Sigma, V = SVD(A)$

$$\rightarrow X_k = \left[\mu_1 \middle| \mu_2 \middle| \dots \middle| \mu_k \right] \begin{bmatrix} \sigma_1 & & 0 \\ & \sigma_2 & \\ & & \ddots \\ 0 & & \sigma_k \end{bmatrix} \begin{bmatrix} \text{---} \, v_1^T \, \text{---} \\ \vdots \\ \text{---} \, v_k^T \, \text{---} \end{bmatrix} =$$

$$= \left[\mu_1 \middle| \mu_2 \middle| \dots \middle| \mu_m \right] \begin{bmatrix} \sigma_1 & & & & 0 \\ & \ddots & & & \\ & & \sigma_k & & \\ 0 & & & \ddots & 0 \end{bmatrix} \begin{bmatrix} \text{---} \, v_1^T \, \text{---} \\ \vdots \\ \text{---} \, v_m^T \, \text{---} \end{bmatrix}$$

$$\|A - X\| = \| U \left(\begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_m \end{bmatrix} - \begin{bmatrix} \sigma_1 & & & & 0 \\ & \ddots & & & \\ & & \sigma_k & & \\ 0 & & & \ddots & 0 \end{bmatrix} \right) V^T \| = \left\| \begin{bmatrix} 0 & & & & 0 \\ & \ddots & & & \\ & & 0 & & \\ & & & \sigma_{k+1} & \\ 0 & & & & \sigma_m \end{bmatrix} \right\|$$

error of rank-1 approx $\|A - X_1\|_F^2 = \sigma_2^2 + \sigma_3^2 + \dots + \sigma_m^2$

if $\sigma_1 \gg \sigma_2, \sigma_3, \dots$ rank-1 approximation is a very good one

Principal component analysis:

$x_1, x_2, \dots, x_m \in \mathbb{R}^m$ data (length-m vectors of features)

$$A = \underbrace{\left[x_1 \mid x_2 \mid \dots \mid x_m \right]}_{\text{examples}} \Big\} \text{features}$$

(1) Removing the mean $\mu = \frac{1}{m}(x_1 + \dots + x_m)$ $\quad \hat{x}_i = x_i - \mu \quad i = 1 \dots m$

(2) Consider SVD of $\hat{A} = [\hat{x}_1 \mid \dots \mid \hat{x}_m]$ and interpret u_i, v_i as different features of the data

OR

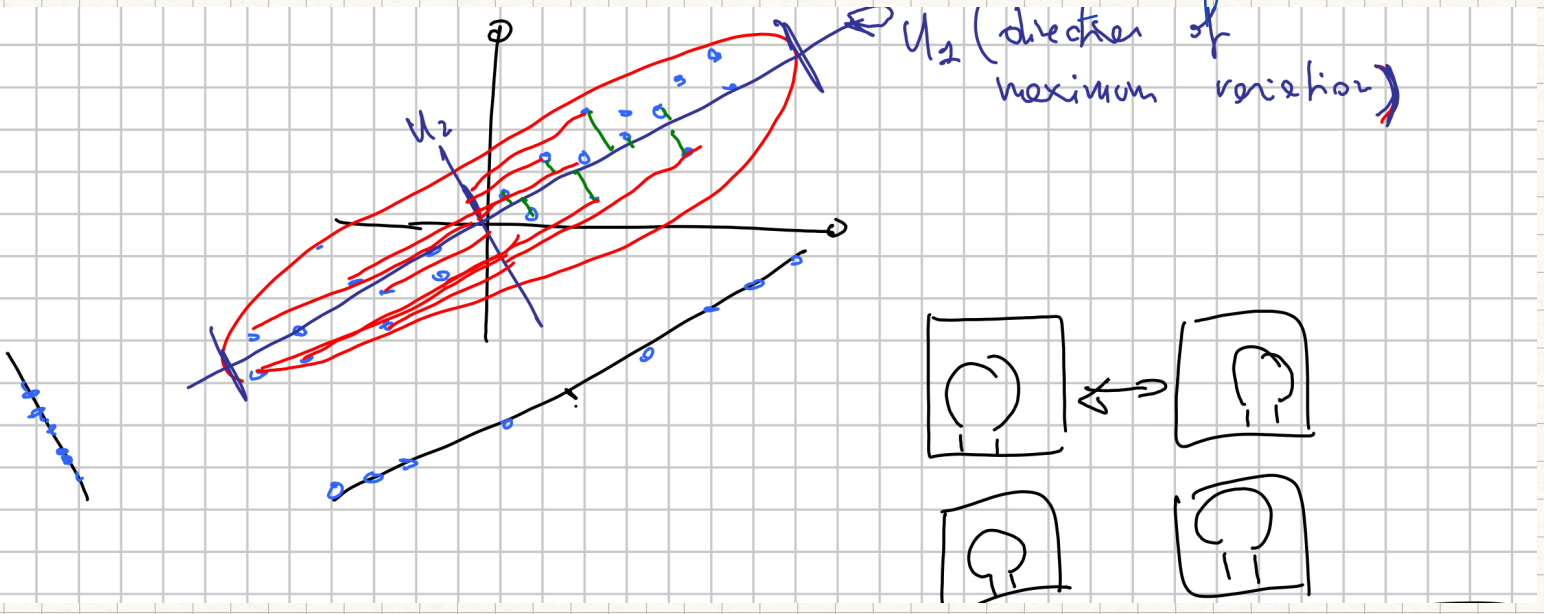
(2) as told by statistician: build covariance matrix

$$C = \frac{1}{m}\left(\hat{x}_1 \hat{x}_1^T + \hat{x}_2 \hat{x}_2^T + \dots + \hat{x}_m \hat{x}_m^T \right) \quad C \text{ is positive definite}$$

PCA builds a basis that lets us write:

$$\hat{x}_j = x_j - \mu = \mu_1 z_{1j} + \mu_2 z_{2j} + \dots + \mu_m z_{mj} \qquad j = 1, 2, \dots, m$$

where $\hat{x}_j$ vary the most in the direction u_1 ($\sum z_{1j}^2$ is max.)



How do we get the z_{ij}?

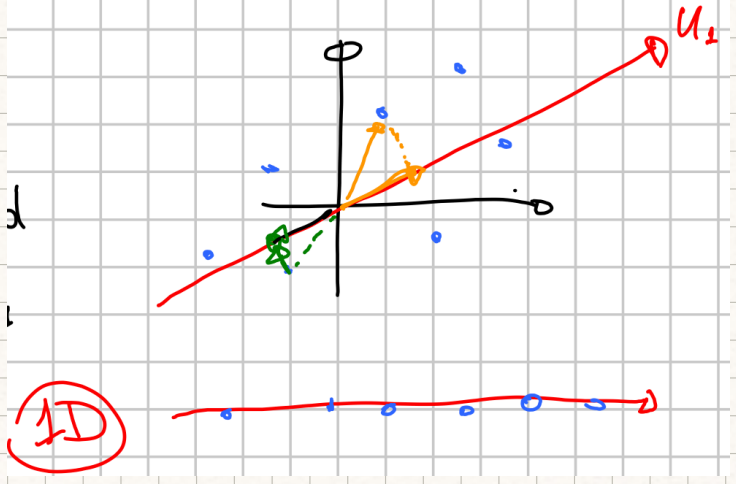
$$\hat{x}_j = \hat{A} \begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \leftarrow e_j = \underbrace{\sigma_1 u_1 v_1^T e_j}_{z_{1j}} + \underbrace{\sigma_2 u_2 v_2^T e_j}_{z_{2j}} + \dots + \underbrace{\sigma_m u_m v_m^T e_j}_{z_{mj}}$$

$$Z = \Sigma V^T = U^T \hat{A} \qquad \hat{x}_j = U z_j \qquad z_j = U^{-1} \hat{x}_j = U^T \hat{x}_j$$

Dimensionality reduction / plotting with PCA

idea: direction of maximum variation gives me a "better plot"

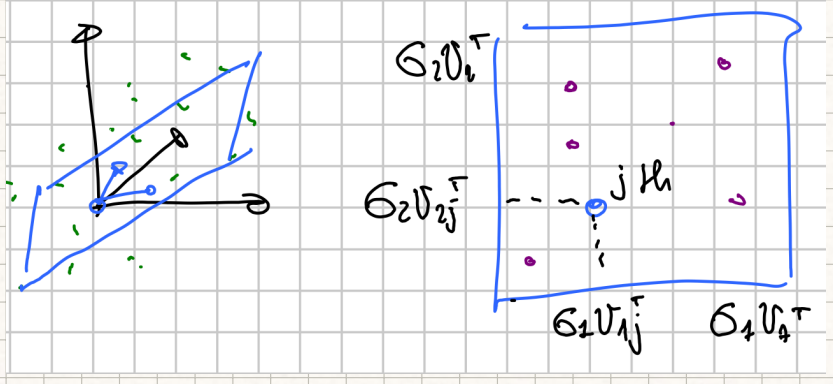
(1D) plotting $\longleftrightarrow$ rank-1 approximation: all of my data point are replaced by multiples of the same vector u_1



$$A \simeq X_1 = \mu_1 \sigma_1 v_1^T$$

(2D): plotting $\longleftrightarrow$ rank-2 approximation: all data point replaced by linear combination of 2 vectors

$$A \simeq X_2 = \mu_1 \sigma_1 v_1^T + \mu_2 \sigma_2 v_2^T$$



Limitations:

- Euclidean distance only
- no labeling of μ_i's.
- "flattening" the data: uses only matrix structure rather than the 4D-structure of my data
- We are lucky that rank-k approximation of a 2D-matrix $\min \|A - X\|$ has an easy solution in $\|\cdot\|_F, \|\cdot\|_2$

rank $X \leq k$

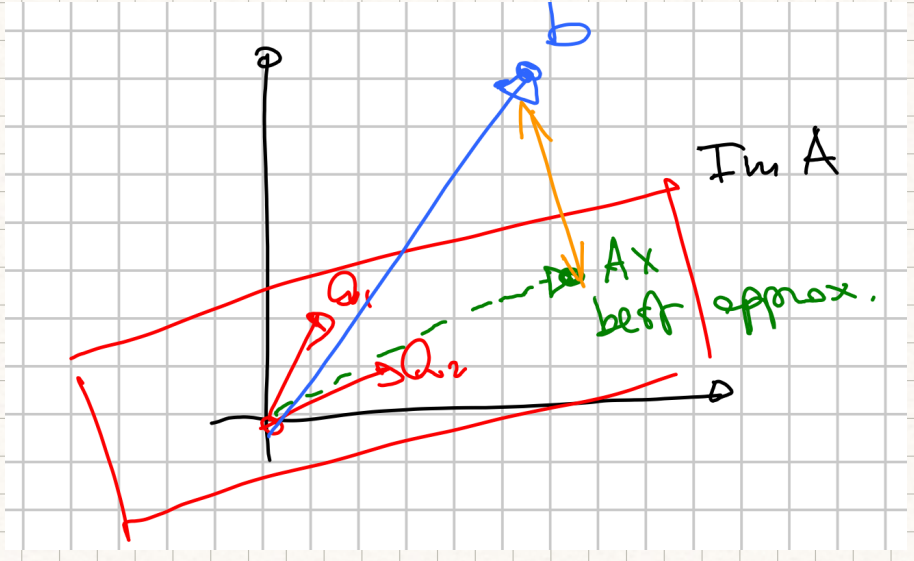
Least square problem: Best approximation problem: given vectors $a_1, a_2, \dots, a_n \in \mathbb{R}^m$ and a "target" vector $b \in \mathbb{R}^m$, find the coefficients that represent a linear combination of the a_i's that gets as close as possible to b

$$a_1 x_1 + a_2 x_2 + \dots + a_n x_n = \underbrace{A}_{\text{"}} x$$

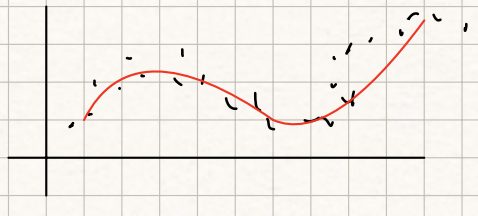
$$\left[a_1 | a_2 | \dots | a_n\right] \in \mathbb{R}^{m \times n} \qquad x \in \mathbb{R}^n$$

We know from linear algebra that if A square invertible then there is one solution that gets exactly b

But in general $m \neq n$ and no exact solution $\min \|Ax - b\|_2$



Polynomial fitting: given point $(x_1, y_1), (x_2, y_2) \dots (x_m, y_m)$



Let's find the degree-3 polynomial

$$y = p(x) = ax^3 + bx^2 + cx + d \qquad a, b, c, d \text{ unknown}$$

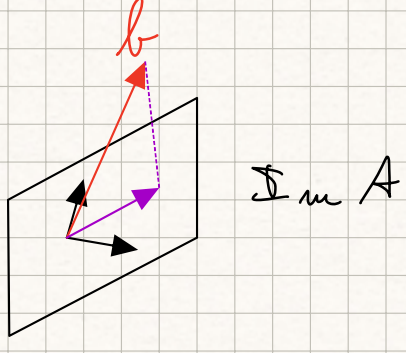
such that $p(x_i) \approx y_i$ $\quad \min \sum_{i=1}^{m} (p(x_i) - y_i)^2$

$$\min_{a,b,c,d} \left\| \begin{bmatrix} x_1^3 & x_1^2 & x_1 & 1 \\ x_2^3 & x_2^2 & x_2 & 1 \\ \vdots & & & \\ x_m^3 & x_m^2 & x_m & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} \right\|^2 = \left\| \begin{bmatrix} p(x_1) - y_1 \\ \vdots \\ p(x_m) - y_m \end{bmatrix} \right\|^2$$

A $\quad x$ $\quad b$

Least square problem

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2 \qquad A \in \mathbb{R}^{m \times n} \qquad b \in \mathbb{R}^m$$



$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad b = \begin{bmatrix} 4 \\ 4 \\ 0 \\ 1 \end{bmatrix}$$

We have multiple solutions:

$$x_1 = \begin{bmatrix} 4 \\ 2 \\ 0 \end{bmatrix} \qquad x_2 = \begin{bmatrix} 0 \\ 0 \\ 4 \end{bmatrix} \qquad x_3 = \begin{bmatrix} 2 \\ 1 \\ 2 \end{bmatrix}$$

$$Ax_1 = Ax_2 = Ax_3 = \begin{bmatrix} 4 \\ 4 \\ 0 \\ 0 \end{bmatrix}$$

this because exist $z \neq 0$ s.t. $Az = 0$

$z = \begin{bmatrix} 2 \\ 1 \\ -2 \end{bmatrix}$ and multiple $\ker A = \operatorname{span}(z)$

Def: $A \in \mathbb{R}^{m \times n}$ has full column rank if $\operatorname{rank}(A) = n$

(i.e. $\ker A = \{0\}$, there are no $z \neq 0$ s.t. $Az = 0$

the min $\|Ax - b\|$ has multiple solutions whenever A does not have full column rank.

Theorem: A has full column rank if and only if $A^T A$ is positive definite. (Recall: $A^T A$ is p. d. if $z^T A^T A z > 0, \forall z \neq 0$)

Proof: A has full cl. rank $\longleftrightarrow$ $Az \neq 0$ for all $z \neq 0$

$\longleftrightarrow \|Az\| \neq 0 \quad \forall z \neq 0 \longleftrightarrow (Az)^T(Az) = \|Az\|^2 \neq 0 \quad \forall z \neq 0 \longleftrightarrow$

$\longleftrightarrow z^T A^T A z \neq 0 \quad \forall z \neq 0 \longleftrightarrow A^T A$ positive defined $\square$

We assume that we have problems with full-column rank

$$\min_{x \in \mathbb{R}^n} \|Ax - b\| \longleftrightarrow \min_{x \in \mathbb{R}^n} \frac{1}{2}\|Ax - b\|^2$$

$$= \min_{x \in \mathbb{R}^n} \frac{1}{2}(Ax - b)^T(Ax - b) = \min_{x \in \mathbb{R}^n} \frac{1}{2}\left(x^T A^T A x - \underline{x^T A^T b} - \underline{b^T A x} + b^T b\right)$$

↖ equal ↗

because $(x^T A^T b)^T = b^T A x$

$$\min_{x \in \mathbb{R}^n} \frac{1}{2} x^T A^T A x - b^T A x + b^T b$$

$\frac{1}{2} x^T Q x + q^T x + c$ quadratic function

$\nabla f = A^T A x - A^T b$

$\nabla^2 f = A^T A$ positive definite matrix.

to find a minimum, set gradient to 0

$$x \text{ minimum} \longleftrightarrow A^T A x - A^T b = 0 \longleftrightarrow x = (A^T A)^{-1} A^T b$$

$$A^T A x = A^T b$$

the solution is $x = (A^T A)^{-1} A^T b$, whenever A has full column rank.

Remark: the solution is $x = A^+ b$, where $A^+ = (A^T A)^{-1} A^T$ is called pseudoinverse.

Computational cost of solving the problem with $x = (A^T A)^{-1} A^T b$:

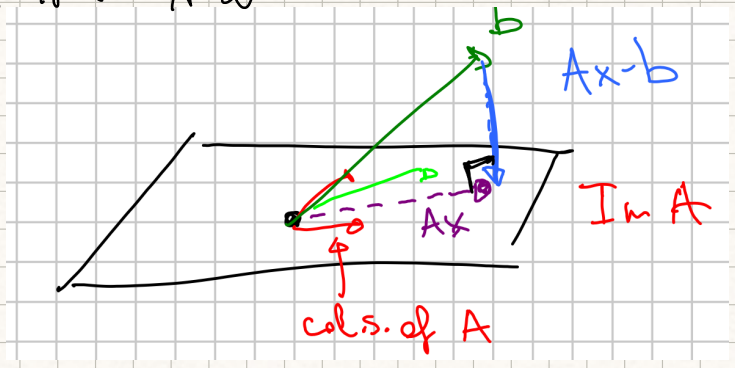
1 Compute $A^T \cdot A$ ($n \times m$, $m \times n$) ← $2mn^2$ flops

2 Compute $A^T \cdot b$ ($n \times m$, $m \times 1$) ← $2mn$

3 Solve $(A^T A) x = (A^T b)$ ($n \times n$, $n \times 1$) ← $\frac{2}{3} n^3$ (Gaussian elimination,

LU facet)

In summary: if you can't solve $Ax = b$, multiply both sides by A^T: $A^T A x = A^T b$



$Ax-b$ is orthogonal to all vectors in $\mathrm{Im}\, A$, in particular to the columns of A $A = [v_1 | v_2 | \dots | v_n]$

$$v_i^T (Ax - b) = 0 \quad \forall i$$

$$\begin{bmatrix} v_1^T \\ \hline \vdots \\ \hline v_n^T \end{bmatrix} (Ax - b) = \begin{bmatrix} 0 \\ \hline \vdots \\ \hline 0 \end{bmatrix} \qquad A^T (Ax - b) = 0$$

Method of normal equations: $x = (A^T A)^{-1} (A^T b)$

Pseudoinverse: $A^+ = (A^T A)^{-1} A^T$ $A \in \mathbb{R}^{m \times n}$ full column rank. $m \geq n$

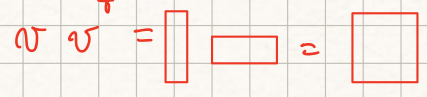
- Size (A^+) = size (A^T)
- If A orthogonal, $A^+ = A^T$, because $A^T A = I$
- $A^+ A = (A^T A)^{-1} A^T A = I$ ▭▯ = □

- $A \cdot A^T \neq I$

Lemma: for every vector $v \in \mathbb{R}^n$, the matrix

$$H = I_n - \frac{2}{v^T v} \cdot v v^T \qquad (*)$$

$v^T v = \|v\|^2$ scalar prod.

$v v^T =$ 

is orthogonal and symmetric.

Matrices of the form $(*)$ are called Householder reflectors.

Proof: $H^T = \left(I - \frac{2}{v^T v} v \cdot v^T\right)^T = I^T - \left(\frac{2}{v^T v} v v^T\right)^T = I - \frac{2}{v^T v} v v^T = H$

$$H^T H = H^2 = \left(I - \frac{2}{v^T v} v \cdot v^T\right)^2 = I - I \frac{2}{v^T v} v \cdot v^T - \frac{2}{v^T v} v \cdot v^T \cdot I + \frac{4}{(v^T v)^2} v \cancel{v^T v} v^T = I$$

Equivalent form: $I - \frac{2}{\|v\|^2} \cdot v v^T = I - 2 u u^T$, where $u = \frac{1}{\|v\|} \cdot v$

If H is a Householder reflector, and w is a vector, then we can compute Hw with cost $O(n)$ rather than $O(n^2)$.

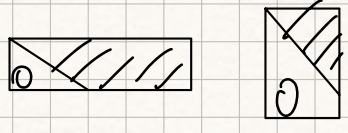
$$Hw = (I - 2 u u^T) w = w - 2u(u^T w)$$

1. compute scalar product $\alpha = u^T w$ $O(n)$ $2n$
2. compute linear combination $w - u(2\alpha)$ $O(n)$ $2n$

Lemma: let $x, y \in \mathbb{R}^m$ with $\|x\| = \|y\|$. If one chooses $v = x - y$, then $H = I_m - \frac{2}{\|v\|^2} v v^T$ is such that $Hx = y$

Th: for every $A \in \mathbb{R}^{m \times m}$, there exist $Q \in \mathbb{R}^{m \times m}$ orthogonal, $R \in \mathbb{R}^{m \times m}$ upper triangular such that $A = Q \cdot R$

upper triangular: $R_{ij} = 0$ if $i > j$



We have already shown this for $m = 1$; given any $A \in \mathbb{R}^{m \times 1}$ there is H Householder reflector s.t. $H \cdot A = \begin{bmatrix} s \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

$A = H \begin{bmatrix} s \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ $(H^{-1} = H^T = H$, as $H^2 = I)$

QR decomposition algorithm:

Step 1: take H_1, Householder reflector such that

$$H \cdot \begin{bmatrix} A_{11} \\ \vdots \\ A_{m1} \end{bmatrix} = \begin{bmatrix} s \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad H_1 A = \begin{bmatrix} s_1 & \star & \cdots & \star \\ 0 & \star & \cdots & \star \\ \vdots & \vdots & & \vdots \\ 0 & \star & \cdots & \star \end{bmatrix}$$

first column of A

Step 2: find $H_2 H_1 A = \begin{bmatrix} \star & s_2 & \star & \cdots & \star \\ \vdots & 0 & \vdots & & \vdots \\ \star & 0 & \star & \cdots & \star \end{bmatrix}$, as it would obstruct zeros in the first column

so...

$$Q_2 = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & H_2 \end{array}\right] \quad (m-1) \qquad Q_2 = \begin{bmatrix} v_1 \\ \hline v_2 \\ \vdots \\ \hline v_m \end{bmatrix} = \begin{bmatrix} v_1 \\ \hline H_2 \cdot \begin{bmatrix} v_2 \\ \vdots \\ v_m \end{bmatrix} \end{bmatrix} \qquad Q_2 (H_1 A) = \begin{bmatrix} s_1 & * & \cdots & * \\ 0 & s_2 & * & * \\ \vdots & 0 & \vdots & \vdots \\ 0 & 0 & * & * \end{bmatrix}$$

$H_2 \in \mathbb{R}^{(m-1)\times(m-1)}$ where H_2 is chosen s.t. $(H_1 A)$ is mapped to $\begin{bmatrix} s_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

Step 3: $Q_1 = H_1$

$$Q_2 Q_1 A = \begin{bmatrix} s_1 & x & x & \cdots & x \\ 0 & s_2 & x & & \vdots \\ \vdots & 0 & x & & \vdots \\ 0 & 0 & x & \cdots & x \end{bmatrix}$$

$H_3 \in \mathbb{R}^{(m-2)\times(m-2)}$ that maps $(Q_2 Q_1 A)_{3:end,3}$ to $\begin{bmatrix} s_3 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

$$Q_3 = \left[\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 1 & \\ \hline 0 & 0 & H_3 \end{array}\right] \text{ so that}$$

$$Q_3 (Q_2 Q_1 A) = \begin{bmatrix} s_1 & x & \cdots & x \\ 0 & s_2 & s_3 & x \\ \vdots & 0 & 0 & \ddots \\ 0 & 0 & 0 & x \end{bmatrix} \begin{matrix} \} \text{ unchanged} \\ \} \text{ multiplied} \\ \text{each column by } H_3 \end{matrix}$$

We can continue until the end, when (after n steps)

$$\underbrace{Q_n Q_{n-1} \cdots Q_2 Q_1}_{\text{orthogonal}} A = \begin{bmatrix} x & \cdots & x \\ 0 & \ddots & x \\ \vdots & \ddots & \ddots \\ 0 & \cdots & x \end{bmatrix} = R \quad \text{upper triangular}$$

$$Q_k = \begin{bmatrix} I_{k-1} & 0 \\ 0 & H_k \end{bmatrix} \begin{matrix} k-1 \\ m-k+1 \end{matrix}$$

$\qquad\qquad\quad k-1 \qquad m-k+1$

Multiply each side on the left by $Q_1 Q_2 Q_3 \dots Q_n$,

use that $Q_k \cdot Q_k = I$ to get $Q_1 Q_2 \dots Q_n Q_n Q_{n-1} \dots Q_2 Q_1 A = Q_1 Q_2 \dots Q_n R$

orthogonal triangular

□

Problem: cubic cost at each iteration $O(m^2 n)$

↓

total cost is a fourth power $O(m^2 n^2)$

We need to use fast householder arithmetic:

$$HM = (I - 2uu^T)M = M - (2u)(u^T M) \leftarrow \text{huge optimization}$$

$p \times p \quad p \times q$

$O(p^2 q) \qquad\qquad O(pq)$

$\qquad\qquad\qquad\qquad\qquad O(pq)$

bring the cost down from quadratic to cubic

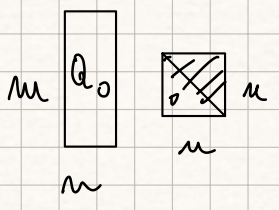
Minor optimization: $H_i A = \begin{bmatrix} s_1 & \times & \cdots & \times \\ 0 & \vdots & & \vdots \\ 0 & \times & \cdots & \times \end{bmatrix}$

↳ we know that the first column of $H_k \cdot R(k{:}end, k{:}end)$ is equal to $\begin{bmatrix} s_k \\ 0 \\ \vdots \end{bmatrix}$, we do not need to recompute it.

2 If $m = n$, then Q_m does very little, I can drop it.

for tall thin A

$$A = QR = \begin{bmatrix} Q_0 & | & Q_c \\ n & & m-n \end{bmatrix} \begin{bmatrix} R_0 \\ \hline 0 \end{bmatrix}_{m-n}^{n} = Q_0 R_0$$



So we can write $A = Q_0 R_0$ with thinner matrices $Q_0 \in \mathbb{R}^{m \times n}$, $R_0 \in \mathbb{R}^{n \times n}$

thin QR factorization: returns only Q_0, R_0, and has cheaper cost $O(mn^2)$. How to return only Q_0 without forming the whole Q?

$$Q = \begin{bmatrix} I - 2u_1 u_1^T \end{bmatrix} \cdot \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & I - 2u_2 u_2^T \end{array}\right] \cdot \left[\begin{array}{c|c} I_2 & \\ \hline & I - 2u_3 u_3^T \end{array}\right] \cdots \left[\begin{array}{c|c} I_{n-1} & \\ \hline & I - 2u_n u_n^T \end{array}\right]$$

(*)

Can be represented only via $u_1, u_2, \dots, u_n \rightarrow O(mn)$

If you want to compute e.g. products $Q \cdot w$, $Q^T w$, then you can use (*) plus fast Householder products to compute products with the same time complexity.

↪ Don't return Q or Q_0, just return $u_1, u_2, \dots u_n$

- You can use this trick also to compute $Q_0 = Q \begin{bmatrix} I_n \\ 0 \end{bmatrix}_n^m$

$$Q_0 = \left[I - 2 u_1 u_1^T \right] \cdots \left[\begin{array}{c|c} I_{n-1} & 0 \\ \hline 0 & I - 2 u_n u_n^T \end{array} \right] \begin{bmatrix} I_n \\ 0 \end{bmatrix}_n^m$$

Compute associating to the right

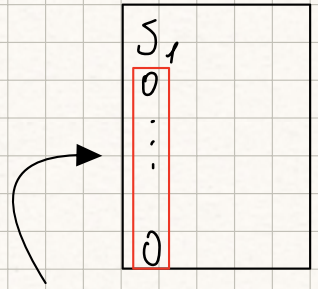
Computational cost:

thin QR: $O(mn^2) \longrightarrow 2mn^2 - \frac{2}{3}n^3 + O(mn)$

→ square: $\frac{4}{3}n^3$ as twice as much as LU factorization

→ tall-thin $m \gg n$ $2mn^2$.

Optimal storage:



recycle this position to store v_1

$u_1 \in \mathbb{R}^{m \times 1}$ $\quad u_1 = \dfrac{v_1}{\|v_1\|}$

up to normalization

$A = QR = Q_0 R_0$ $\quad Q = \underset{m}{\overset{m}{\left[Q_0 \mid Q_c \right]}} \in \mathbb{R}^{m \times m}$ orthogonal

$R = \begin{bmatrix} R_0 \\ 0 \end{bmatrix}_{m-n}^{n}$ R_0 upper triangular

How to use the QR factorization to solve L.S. prob.:

Find $x \in \mathbb{R}^{n \times m}$ that minimizes $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$

$$\|Ax - b\| = \|QRx - b\| = \|Q^T(QRx - b)\| = \|Rx - Q^T b\| =$$

↑ orth. matrix preserve the 2-norm

$$= \left\| \begin{bmatrix} R_0 \\ 0 \end{bmatrix} x - \begin{bmatrix} Q_0^T \\ Q_c^T \end{bmatrix} b \right\| = \left\| \begin{bmatrix} R_0 x \\ 0 \end{bmatrix} - \begin{bmatrix} Q_0^T b \\ Q_c^T b \end{bmatrix} \right\| = \left\| \begin{bmatrix} R_0 x - Q_0^T b \\ Q_c^T b \end{bmatrix} \right\|$$

We make the norm as small as possible choosing x as:

Second block: does not contain x, its entries are always going to be in the sum of squares.

First block: becomes 0 if x solves $R_0 x - Q_0^T b$ (square linear system with matrix R_0). If R_0 is invertible, this is always possible.

$\min_x \|Ax - b\|$ equals to $\|Q_c^T b\|$, and is achieved by

$$x = R_0^{-1} Q_0^T b$$

Where R_0 is invertible → Lemma: R_0 is invertible i.f.f. A has full column rank.

Proof: A has full col. rank ⟷ $A^T A$ is invertible ⟷

$\longleftrightarrow$ $R^T Q^T Q R$ is invertible $\longleftrightarrow$ $[R_0^T \; 0] \begin{bmatrix} R_0 \\ 0 \end{bmatrix} = R_0^T R_0$ is invertible $\longleftrightarrow$ R_0 is invertible.

Second algorithm to solve o.l.s. prob. $\min_x \|Ax - b\|$, with $A \in \mathbb{R}^{m \times n}$ with full column rank:

1 Compute thin QR factorization of A $\quad 2mn^2$ if $m \gg n$

2 form $c = Q_0^T b$ $\quad 2mn$

$n \times m \quad m \times 1$

3 Solve lin. system $R_0 x = c$ $\quad n^2$

$n \times n$

$$\begin{bmatrix} R_{11} & \cdots & R_{1n} \\ & \ddots & \vdots \\ 0 & & R_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}$$

Solving least squares problem with SVD

$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2 \quad A \in \mathbb{R}^{m \times n} \quad m \geq n \quad b \in \mathbb{R}^m$

$A = U S V^T$

$U = \left[\underset{m}{U_0} \;\middle|\; \underset{m-n}{U_1} \right]_m \quad S = \begin{bmatrix} S_0 \\ \hline 0 \end{bmatrix}_{m-n}^{n} \quad V = \mathbb{R}^{n \times n}$

$U \in \mathbb{R}^{m \times m}$

$S = \boxed{} \in \mathbb{R}^{m \times n}$

$A = U_0 S_0 V^T$

$V^T x = y$

$$\|Ax - b\| = \|USV^T x - b\| = \|U^T(USV^T x - b)\| = \|Sy - U^T b\| =$$

$$= \left\| \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ 0 & & \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} - \begin{bmatrix} u_1^T b \\ \vdots \\ u_n^T b \end{bmatrix} \right\| = \left\| \begin{bmatrix} \sigma_1 y_1 \\ \vdots \\ \sigma_n y_n \\ 0 \end{bmatrix} - \begin{bmatrix} u_1^T b \\ \vdots \\ u_n^T b \end{bmatrix} \right\|$$

We can choose y s.t. the first n rows are equal to 0

$$y_i = \frac{u_i^T b}{\sigma_i}$$

$$x = Vy = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix} \begin{bmatrix} \frac{u_1^T b}{\sigma_1} \\ \vdots \\ \frac{u_n^T b}{\sigma_n} \end{bmatrix} = v_1 \frac{u_1^T b}{\sigma_1} + v_2 \frac{u_2^T b}{\sigma_2} + \dots + v_n \frac{u_n^T b}{\sigma_n}$$

small singular value count more

$$x = V \begin{bmatrix} 1/\sigma_1 & & 0 \\ & \ddots & \\ 0 & & 1/\sigma_n \end{bmatrix} \begin{bmatrix} u_1^T b \\ \vdots \\ u_n^T b \end{bmatrix} = \underbrace{V S_0^{-1} U_0^T}_{A^+} b$$

The value of the minimum is $\left\| \begin{bmatrix} 0 \\ \vdots \\ 0 \\ u_{n+1}^T b \\ \vdots \\ u_m^T b \end{bmatrix} \right\| = \| U_0^T b \|$

$\sigma_i \neq 0$ for all $i = 1, 2, \dots n$ i.f.f. A has full column rank. Indeed,

A full col. rank $\longleftrightarrow$ $A^T A$ is non singular

$$A^T A = (USV^T)(USV^T) = VS^T U^T USV^T = V \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_n^2 \end{bmatrix} V^T$$

which is invertible i.f.f. $\sigma_i \neq 0 \ \forall i$.

- Also SVD reveal how far A is from a matrix without full column rank.

X doesn't have a full col. rank $\longleftrightarrow$ rank X in $n-1$ or lower

$\min \|Ax\| = \sigma_n$ the minimum value is obtained with truncated SVD.

σ_n is the smallest singular value, so A has full col. rank $\longleftrightarrow \sigma_n \neq 0$.

With SVD we can solve problem in which A doesn't have a full rank. $\|Ax - b\| =$

$$\left\| \begin{bmatrix} \sigma_1 y - u_1^T b \\ \vdots \\ \sigma_n y - u_n^T b \\ -u_{n+1}^T b \\ \vdots \\ -u_m^T b \end{bmatrix} \right\|$$

If $\sigma_{z+1} = \sigma_{z+2} = \sigma_n = 0$ i.e. the rank of A is z, the $y_{z+1} \dots y_n$ do not affect the value of the norm. We can choose them arbitrarily $\longrightarrow \infty$ solutions

Among the infinite solutions, one can identify the one with the smallest norm:

$$\min_{x \text{ solves } \min\|Ax-b\|} \|x\| = \min\|Vx\| = \min\|y\|$$

the optimum is $\begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} \frac{u_1^T b}{\sigma_1} \\ \frac{u_2^T b}{\sigma_2} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$ } proof so that y is a solution
} to get minimum norm

Solution of rank-deficient least square problem

$$x = Vy = v_1 \frac{u_1^T b}{\sigma_1} + \dots + v_2 \frac{u_2^T b}{\sigma_2}$$

$$= \begin{bmatrix} v_1 & v_2 & \dots & v_m \end{bmatrix} \begin{bmatrix} 1/\sigma_1 & & & \\ & \ddots & & \\ & & 1/\sigma_2 & \\ & & & 0 \end{bmatrix} \begin{bmatrix} u_1^T b \\ \vdots \\ u_m^T b \end{bmatrix}$$

$$= \overbrace{V \begin{bmatrix} 1/\sigma_1 & & & \\ & \ddots & & \\ & & 1/\sigma_2 & \\ & & & 0 \end{bmatrix} U_0^T}^{A^+} b$$

Def: pseudoinverse of a matrix A without full col. rank is

$$A^+ = V \begin{bmatrix} 1/\sigma_1 & & & \\ & \ddots & & \\ & & 1/\sigma_2 & \\ & & & 0 \end{bmatrix} U_0^T$$

with this definition, the vector $x = A^+ b$ is always a

solution of $\min \|Ax - b\|$, and if there is an infinite number of them this formula gives the one with the smallest norm.

Truncating: in SVD many singular values (and their u_i, v_i) can be set to 0, as they are responsible for smaller changes in norm.

Ridge regression is alternative to truncation

$$\min_x \|Ax - b\|^2 + \lambda^2 \|x\|^2 \qquad \lambda \geq 0$$

Trick:

$$\|Ax - b\|^2 + \lambda^2 \|x\|^2 = \left\| \begin{bmatrix} A \\ \lambda I_n \end{bmatrix}_{m+n} \cdot x - \begin{bmatrix} b \\ 0 \end{bmatrix}_{m+n} \right\|^2 = \left\| \begin{bmatrix} Ax - b \\ \lambda x \end{bmatrix} \right\|^2$$

solution of the ridge-regression problem:

$$x_{ridge} = \begin{bmatrix} A \\ \lambda I_n \end{bmatrix}^+ \begin{bmatrix} b \\ 0 \end{bmatrix} = \left(\begin{bmatrix} A^T & \lambda I \end{bmatrix} \begin{bmatrix} A \\ \lambda I \end{bmatrix} \right)^{-1} \begin{bmatrix} A^T & \lambda I \end{bmatrix} \begin{bmatrix} b \\ 0 \end{bmatrix}$$

$$= (A^T A + \lambda^2 I)^{-1} (A^T b)$$

additional term with respect to $x = (A^T A)^{-1} A^T b$

always positive definite

Alternative expression with SVD $A = U S V^T$

$$x_{ridge} = V \begin{bmatrix} \frac{\sigma_1}{\sigma_1^2+\lambda^2} & & \\ & \ddots & \\ & & \frac{\sigma_n}{\sigma_n^2+\lambda^2} \end{bmatrix} U^T b$$

the quantity $\frac{1}{\sigma_i}$, which appeared in the analogous expression for x is replaced by $\frac{\sigma_i}{\sigma_i^2+\lambda^2}$

$$\frac{\sigma_i}{\sigma_i^2+\lambda^2} \begin{cases} \frac{1}{\sigma_i} \longrightarrow \text{if } \sigma_i \gg \lambda \\ 0 \longrightarrow \text{if } \sigma_i \ll \lambda \end{cases}$$

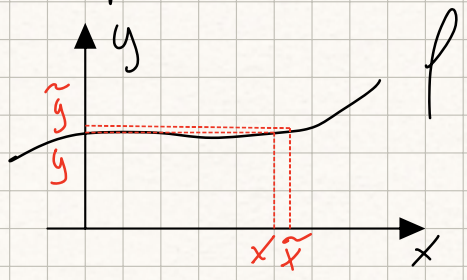
this approximation truncating singular values that are smaller than $\approx \lambda$

Recap: 3 algorithms for LS problems:

Normal equations	QR	SVD
$x = (A^T A)^{-1} A^T b$	$x = R_1^{-1} Q_1^T b$	$x = V S_0^{-1} U_0^T b$
		Pseudoinverse A^+
$m \gg n$ $\quad mn^2$	$\approx 2mn^2$	$2mn^2$
$m \approx n$ $\quad \approx 4/3\, n^3$	$\approx 4/3\, n^3$	$13 n^3$

Condition number: a way to measure the sensitivity of a problem for small perturbations of the inputs

given $f: \mathbb{R} \to \mathbb{R}$ $\quad y = f(x)$ $\quad \tilde{y} = f(\tilde{x})$



$$f(\tilde{x}) = f(x) + f'(x)(\tilde{x} - x) + O\left((\tilde{x} - x)^2\right)$$

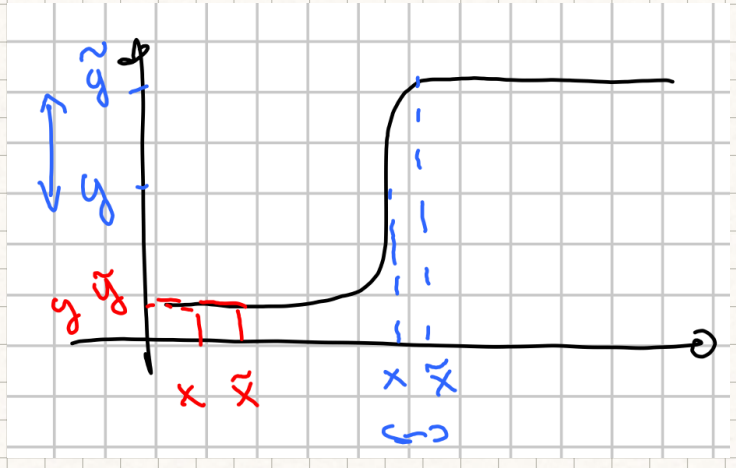
$f(\tilde{x}) = \tilde{y}$, $f(x) = y$

Relative error difference:

$$\frac{|\tilde{y} - y|}{|y|} = \frac{|f'(x)(\tilde{x} - x)|}{|y|} + O\left((\tilde{x} - x)^2\right) = \frac{|f'(x)| \cdot |x|}{|f(x)|} \cdot \frac{|\tilde{x} - x|}{|x|} + O(\cdot)$$

$\frac{|f'(x)| \cdot |x|}{|f(x)|}$: condition number of f in point x

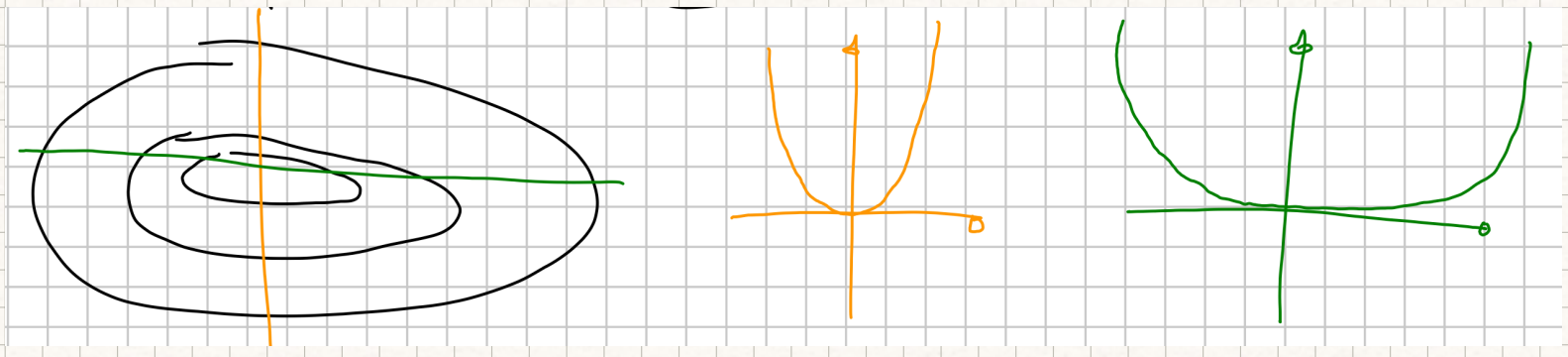
$\frac{|\tilde{x} - x|}{|x|}$: rel. error on input



Formal definition: condition number of $f: \mathbb{R} \to \mathbb{R}$

$$\lim_{|\tilde{x} - x| \to 0} \frac{\frac{|f(\tilde{x}) - f(x)|}{|f(x)|}}{\frac{|\tilde{x} - x|}{|x|}}$$

For function with more inputs & outputs, the answer may also depend on the direction:



We can define a maximum over all directions

Def:

$$K_{f,x} = \lim_{\|\tilde{x} - x\| \to 0} \frac{\frac{\|f(\tilde{x}) - f(x)\|}{\|f(x)\|}}{\frac{\|\tilde{x} - x\|}{\|x\|}}$$

In terms of the Jacobian (matrix of derivatives)

$$K_{f,x} = \frac{\|\nabla f_x\| \cdot \|x\|}{\|f(x)\|}$$

The condition number is a warning sign: it tells you whether your problem is very sensitive to input perturbations.

Conditioning of linear algebra problems:

solving linear system $A \in \mathbb{R}^{n \times n}$ invertible, $b \in \mathbb{R}^n$, $Ax = b$

let $\tilde{b}$ a perturbation of b

$A\tilde{x} = \tilde{b}$ perturbed output

$Ax = b$ exact output

$$\frac{\|\tilde{x} - x\|}{\|x\|} \leq \kappa \frac{\|\tilde{b} - b\|}{\|b\|}$$

Theorem: this condition number is $\kappa(A) = \|A\| \cdot \|A^{-1}\|$

Proof: $\|\tilde{x} - x\| = \|A^{-1}\tilde{b} - A^{-1}b\| = \|A^{-1} \cdot (\tilde{b} - b)\| \leq \|A^{-1}\| \cdot \|\tilde{b} - b\|$

$$\|b\| = \|Ax\| \leq \|A\| \cdot \|x\| \longrightarrow \|x\| \geq \frac{\|b\|}{\|A\|}$$

$$\frac{\|\tilde{x} - x\|}{\|x\|} \leq \frac{\|A^{-1}\| \|\tilde{b} - b\|}{\frac{\|b\|}{\|A\|}} = \boxed{\|A^{-1}\| \cdot \|A\|} \cdot \frac{\|\tilde{b} - b\|}{\|b\|}$$

conditioned number of solving linear equations w.r.t. b

Def: $\kappa(A) = \|A\| \cdot \|A^{-1}\|$ is called the condition number of A

Condition number and singular value

$$\|A\| = \sigma_1$$

$$\|A^{-1}\| = \|(U S V^T)^{-1}\| = \|(V^T)^{-1} S^{-1} U^{-1}\| = \|V \underbrace{\begin{bmatrix} 1/\sigma_1 & & \\ & \ddots & \\ & & 1/\sigma_n \end{bmatrix} U^T}_{\text{almost an SVD}}\|$$

$$= \|\underbrace{\begin{bmatrix} v_n & \cdots & v_1 \end{bmatrix} \begin{bmatrix} 1/\sigma_n & & \\ & \ddots & \\ & & 1/\sigma_1 \end{bmatrix} \begin{bmatrix} u_n & \cdots & u_1 \end{bmatrix}^T}\| = \frac{1}{\sigma_n}$$

SVD of A^{-1}

$\kappa(A) = \|A\| \cdot \|A^{-1}\| = \frac{\sigma_1}{\sigma_n}$ formula for $\kappa(A)$ in terms of SVD

Condition number and distance to singularity:

$$\min_{X \text{ singular}} \|A - X\| = \sigma_n \qquad \min_{X \text{ singular}} \frac{\|A - X\|}{\|A\|} = \frac{\sigma_n}{\sigma_1} = \frac{1}{\kappa(A)}$$

Condition number of solving linear system w.r.t. perturbations of A: $Ax = b$ $\tilde{A}\tilde{x} = b$

$$\frac{\|\tilde{x} - x\|}{\|x\|} \leq \underbrace{\|A\| \cdot \|A^{-1}\|}_{\kappa(A)} \cdot \frac{\|\tilde{A} - A\|}{\|A\|} + O\left(\|\tilde{A} - A\|^2\right)$$

Condition number of least square problems.

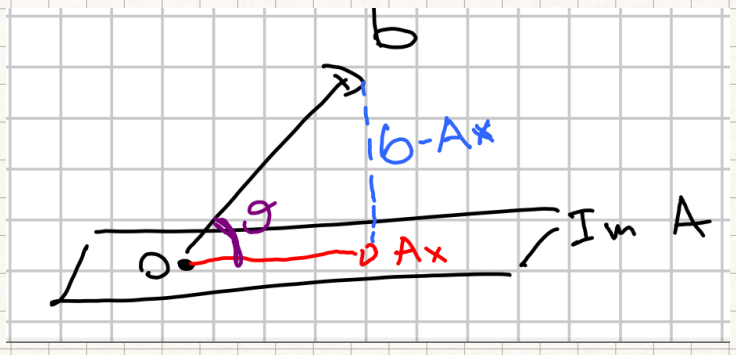
Consider the L.S. problem $\min \|Ax - b\|$, $A \in \mathbb{R}^{m \times n}$ full column rank $b \in \mathbb{R}^m$. The condition number w.r.t. input b is:

$$\kappa_{b \to x} \leq \frac{\kappa(A)}{\cos\theta}$$

And w.r.t. input A

$$\kappa_{A \to x} \leq \kappa(A) + \kappa(A)^2 \tan\theta$$

where $\kappa(A) = \frac{\sigma_1}{\sigma_n}$, θ is the angle s.t. $\cos\theta = \frac{\|Ax\|}{\|b\|}$



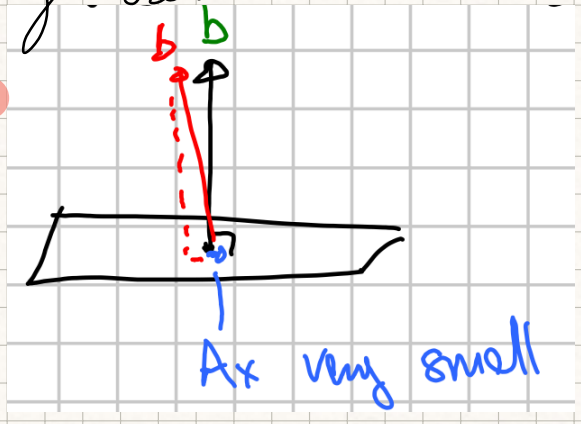
θ = angle between $\mathrm{Im}\, A$ and b

$$\|b - Ax\| = \|U_c^T b\| = \|U_c^T b\|$$

$$\|Ax\| = \|U_0^T b\| = \|U_0^T b\|$$

Interesting case:

$\theta \approx 90°$

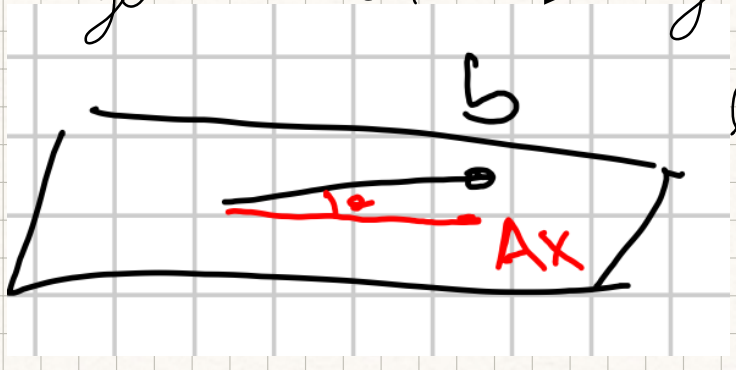


$$\kappa_{b \to x} = \frac{\kappa(A)}{\cos\theta}$$ very large

$$\kappa_{A \to x} = \kappa(A) + \tan\theta\, \kappa(A)^2$$ very large

small change in b or A ⟶ large change in x

$\theta \approx 0°$



$b - Ax$ very small

$$\kappa_{b \to x} = \kappa(A)$$

$$\kappa_{A \to x} = \kappa(A)$$

behave like a linear system, $\kappa(A)$

$0° \leq \theta < 90°$ $\quad \kappa_{b \to x} = \frac{\kappa_i(A)}{\cos(\theta)}$ $\quad \kappa_{A \to x} = \kappa(A) + \tan\theta \cdot \kappa(A)^2$

more sensitive to perturbation

of A then a linear system

Note that $\frac{1}{\kappa(A)}$ = rel. distance between A and the closest matrix without full column rank.

Sensitivity: how does the solution change w.r.t. small input changes ⟶ condition number

Stability: how well does an algorithm compute the solution to the problem

Stability of numerical algorithms

if your input is not exact floating point numbers, then you cannot compute the exact solution.

Real input x — F.P. approximation $\tilde{x}$ — $\frac{|\tilde{x}-x|}{|x|} \leq u$ $u \approx 10^{-16}$

or $\frac{\|\tilde{x}-x\|}{\|x\|} \leq O(m,n)\,u$ for vectors/matrices

You can't compute $f(x)$, only $f(\tilde{x})$

$$\frac{\|f(\tilde{x}) - f(x)\|}{\|f(x)\|} \leq \kappa_f \times \frac{\|\tilde{x}-x\|}{\|x\|}$$ "intrinsic error"

To asses error of an algorithm, the only resource is keeping track of each floating point error

machine precision

$$a \oplus b \longrightarrow (a+b)(1+\varepsilon) \qquad |\varepsilon| \leq u$$

Trick: backward stability analysis; for some algorithms you can write the result $\tilde{y}$ as an exact $f(\hat{a}, \hat{b})$ with $\hat{a}, \hat{b}$ "close" to a, b.

$$\tilde{y} \doteq \underbrace{a_1 b_1 + a_2 b_2 + a_3 b_3}_{y} + a_1 b_1 (\varepsilon_1 + \varepsilon_4 + \varepsilon_5) + a_2 b_2 (\varepsilon_2 + \varepsilon_4 + \varepsilon_5) + a_3 b_3 (\varepsilon_3 + \varepsilon_5)$$

$$= a_1 \hat{b}_1 + a_2 \hat{b}_2 + a_3 \hat{b}_3$$

$$\hat{b}_1 = b_1 (1 + \varepsilon_1 + \varepsilon_4 + \varepsilon_5)$$
$$\hat{b}_2 = b_2 (1 + \varepsilon_2 + \varepsilon_4 + \varepsilon_5)$$
$$\hat{b}_3 = b_3 (1 + \varepsilon_3 + \varepsilon_5)$$

$$\|\hat{b} - b\|_2 = \left\| \begin{bmatrix} b_1(\varepsilon_1 + \varepsilon_4 + \varepsilon_5) \\ b_2(\varepsilon_2 + \varepsilon_4 + \varepsilon_5) \\ b_3(\varepsilon_3 + \varepsilon_5) \end{bmatrix} \right\| \leq 3u \|b\|_2$$

This tells us immediately an accuracy bound on y:

$$\frac{|\hat{y} - y|}{|y|} \leq \kappa_{f,b} \frac{\|\hat{b} - b\|}{\|b\|} = \kappa_{f,b} \cdot 3u$$

$\Rightarrow$ the result obtained is just a vector 3 away from the (unavoidable) error coming from machine-precision approximation of u

$$\frac{|\tilde{y} - y|}{|y|} \leq \kappa_{f,b} \underbrace{\frac{\|\tilde{b} - b\|}{\|b\|}}_{u}$$

We say that an algorithm (or a computation) is backward stable if the computed output $\tilde{y} = f(\tilde{x})$ is exactly equal to $f(\hat{x})$, where $\frac{\|\hat{x} - x\|}{\|x\|} = O(u)$. In this case, the algorithm is as accurate as it could get given the intrinsic error.

algorithmic error $\simeq$ intrinsic error

⚠ Here $O(u)$ can include small-degree polynomials of the dimensions, because typically these are overestimates, and because they are typically much smaller than possible condition numbers.

this trick doesn't work for every algorithm.

Sometimes it's possible to prove stability "a posteriori" after you have computed a solution, you can show it solves a nearby problem.

es: residual test for square linear systems

$A \in \mathbb{R}^{m \times m}$, $b \in \mathbb{R}^m$ find $x \in \mathbb{R}^m$ that solves $Ax = b$

Suppose your algorithm computes $\tilde{x}$ (approx. solution)

residual: $z = A\tilde{x} - b$

Th: the error satisfies

$$\frac{\|\tilde{x} - x\|}{\|x\|} \leq \kappa(A) \cdot \frac{\|r\|}{\|b\|}$$

$$\kappa(A) = \|A\| \cdot \|A^{-1}\|$$

Proof: $r = A\tilde{x} - b \longleftrightarrow A\tilde{x} = \underbrace{b + r}_{\hat{b}}$

$\tilde{x}$ is the exact solution of modified input $A, \hat{b}$. Then, the cond. number bound tells us that:

$$\frac{\|\tilde{x} - x\|}{\|x\|} \leq \kappa(A) \cdot \frac{\|\hat{b} - b\|}{\|b\|} = \kappa(A) \cdot \frac{\|r\|}{\|b\|}$$

→ solution with small residual are accurate (up to $\kappa(A)$)

problem $r = Ax - b$ is affected by an error itself. then error is bounded by $\|A\| \cdot \|x\| \cdot O(u)$

→ r is typically accurate at least as order of magnitude.

Conversely, backward stable solution → small residual:

if $\tilde{x}$ solves $A\tilde{x} = \hat{b}$ with $\frac{\|\hat{b} - b\|}{\|b\|} = O(u)$

then $A\tilde{x} - b = \hat{b} - b$ small w.r.t. $\|b\|$

$\hat{A} = A + E \qquad \frac{\|E\|}{\|A\|} = O(u)$

If $\tilde{x}$ solves $\hat{A}\tilde{x} = \hat{b}$

$\hat{b} = b + f \qquad \frac{\|f\|}{\|b\|} = O(u)$

↓

then $(A+E)\tilde{x} = b + f \longleftrightarrow A\tilde{x} - b = f - E\tilde{x}$

$$\|A\tilde{x} - b\| \le \|f\| + \|E\| \cdot \|\tilde{x}\| = O(u)\left(\|b\| + \|A\| \cdot \|\tilde{x}\|\right)$$

QR factorization: $A = QR$ if $\|A - \tilde{Q}\tilde{R}\|$ is small, then you have computed an exact solution of $qr(A+E)$ with $E = \tilde{Q}\tilde{R} - A$, hence

$$\frac{\|\tilde{E} - E\|}{\|E\|} \le K_{qr,R} \cdot \frac{\|E\|}{\|A\|}$$

small residual implies algorithm as accurate as possible

Residual tests for LS problems: $r = A\tilde{x} - b$ is not necessary small

$\min \frac{1}{2}\|Ax - b\|^2 =$

$\min \frac{1}{2} x^T A^T A x - b^T A x + \text{const} \longrightarrow$ small gradient

$\nabla f(\tilde{x}) = A^T A\tilde{x} - A^T b$ (residual of normal equations)

$A^T A x = A^T b$

Residual bound on normal equations:

$$\frac{\|\tilde{x} - x\|}{\|x\|} \leq \kappa(A^TA) \cdot \frac{\|A^TA\tilde{x} - A^Tb\|}{\|A^Tb\|}$$

$A^TA = V \begin{bmatrix} \sigma_1^2 & & \\ & \sigma_2^2 & \\ & & \ddots \\ & & & \sigma_n^2 \end{bmatrix} V^T$ ← SVD of A^TA $\quad \kappa(A^TA) = \frac{\sigma_1^2}{\sigma_n^2} = \kappa(A)^2$

Even if $\frac{\|A^TA\tilde{x} - A^Tb\|}{\|A^Tb\|} = O(u)$, we get an error of the order of

$\frac{\|\tilde{x} - x\|}{\|x\|} \leq \kappa(A^TA) \cdot O(u)$, but $\kappa_{LS,A}$ and $\kappa_{LS,b}$ can be better smaller

than $\kappa(A^TA) = \kappa(A)^2$

Th: Suppose $A = Q_0 R_0$ is a thin QR factorization exactly computed!

Let $r_0 = Q_0^T(A\tilde{x} - b)$. Then $\tilde{x}$ is the exact solution of

$$\min \|A\tilde{x} - (b + Q_0 r_0)\|$$

$$\frac{\|\tilde{x} - x\|}{\|x\|} \leq \kappa_{LS,b} \cdot \frac{\|Q_0 r_0\|}{\|b\|} = \kappa_{LS,b} \cdot \frac{\|r_0\|}{\|b\|}$$

orthogonal

true cond. number of the problem

For good (backward stable) algorithms, relative residuals ≈ machine precision and this let you prove that

$$\frac{\|\tilde{x} - x\|}{\|x\|} \leq \kappa_{prob} \cdot O(u)$$

best one can do because of intrinsic error.

"A priori" backward stability for alg. that are based on orthogonal transformations

Stability of matrix multiplication

$$y = a^T b \qquad \tilde{y} = a^T \odot b$$

$$|\tilde{y} - y| \leq O(n) \cdot |a| \cdot |b|$$

$$C = A \cdot B \qquad A, B, C \in \mathbb{R}^{n \times n} \qquad \tilde{C} = A \odot B$$

$$|\tilde{c}_{ij} - c_{ij}| \leq O(n) \cdot |A_{i:}| \cdot |B_{:j}|$$

This gives rise to a normwise bound:

$$\|\tilde{C} - C\| \leq O(n) \|A\| \cdot \|B\|$$

↑ depends on the chosen norm

If one of the matrices is orthogonal, then:

$$\|\tilde{C} - C\| \leq O(n) \|A\| \cdot \|B\| \leq O(n) \cdot \|B\|$$

$$\tilde{C} = C + E \qquad \|E\| \leq O(n) \cdot \|B\| \leftarrow \text{forward stability bound}$$

$\|C\| = \|A \cdot B\| = \|B\|$ if A orthogonal

$$\frac{\|\Theta\|}{\|C\|} \leq O(u)$$

Backward stability of:

$$\hat{C} = C + \Theta = A\hat{B} \qquad \hat{B} = B + F \qquad \frac{\|F\|}{\|B\|} = O(u)$$

$$\hat{C} = C + \Theta = AB + \Theta = A(B + \underbrace{A^T\Theta}_{F})$$

$$\|F\| = \|A^T\Theta\| = \|\Theta\| \leq O(u) \cdot \|B\|$$

(A^T: orthogonal)

What we proved: multiplication, with the standard alg., by an orthogonal matrix is backward stable.

(Remark: if A were not orthogonal, the same computation gives

$$\hat{C} = A \odot B = A(B + F) \qquad \|F\| \leq O(u) \cdot \|B\| \cdot \kappa(A)$$

If H is a Householder reflector, then the product is backward stable

$$\hat{C} = H \odot B = H(B + F) \qquad H = I - 2uu^T$$

$$\|F\| = O(u)\|B\| \qquad HB = B - 2u(u^TB)$$

Proof wie $\hat{C} = C + \Theta$, $\|\Theta\| = \|B\| O(u)$ by keeping track of $(1+\varepsilon_i)$ errors

Th: $[\tilde{Q}\ \tilde{R}] = qr(A)$ computed via Householder reflectors returns matrices such that $\tilde{Q}\tilde{R} = A + F$, $\|F\| = O(u)\|A\|$.

($\tilde{Q}$ represented implicitly via reflectors $v_1, v_2, \dots, v_m$)

Proof: exact qr steps

$$\begin{bmatrix} I_{k-1} & \\ & H(u_k) \end{bmatrix} \begin{bmatrix} \times & \cdots & \times \\ 0 & \ddots & \vdots \\ \vdots & \ddots & \times \\ 0 & 0 & \times \end{bmatrix} = \begin{bmatrix} \times & \cdots & \times \\ 0 & \times & \vdots \\ \vdots & 0 & \times \\ 0 & 0 & 0 \end{bmatrix}$$

Q_k ; $k-1$; A_{k-1} ; A_k

$$\begin{bmatrix} I_{k-1} & \\ & H(\tilde{u}_k) \end{bmatrix} \odot \tilde{A}_{k-1} = \tilde{A}_k$$

$\tilde{Q}_k$ is exactly orthogonal

$\tilde{Q}_k$

$$\tilde{A}_k = \tilde{Q}_k \tilde{A}_{k-1} + \underset{k-1}{F} = \tilde{Q}_k (\tilde{A}_{k-1} + \underset{k-1}{F})$$

$$\|F_{k-1}\| = O(u)\|\tilde{A}_{k-1}\| = O(u)\|A\|$$

$$\tilde{A}_1 = \tilde{Q}_1 (A + F_0)$$

$$\tilde{A}_2 = \tilde{Q}_2 (\tilde{A}_1 + F_1) = \tilde{Q}_2 \tilde{Q}_1 (A + F_0 + \tilde{Q}_1^T F_1)$$

$$\tilde{A}_3 = \tilde{Q}_3 (\tilde{A}_2 + F_2) = \tilde{Q}_3 \tilde{Q}_2 \tilde{Q}_1 (A + F_0 + \tilde{Q}_1^T F + \tilde{Q}_2^T \tilde{Q}_1 F_2)$$

all errors written as modification of the matrix A

$$\tilde{R} = \tilde{Q}_n \tilde{Q}_{n-1} \dots \tilde{Q}_1 (A + F_0 + \dots + \tilde{Q}_n \dots Q_1 F_{n-1})$$

n errors terms, each of norm $O(u)\,\|A\|$

$$\tilde{R} = \tilde{Q}_n \dots \tilde{Q}_1 (A+F) \qquad \|F\| = O(u)\,\|A\|$$

→ exact computed $\tilde{R}$ (and $\tilde{Q}$) are the exact result of $qr(\hat{A})$ $\hat{A} = A + F$ $\|F\| = O(u)\,\|A\|$ ■

In addition, one can also prove that backward substitution

$$\begin{bmatrix} x & \cdots & x \\ 0 & \ddots & \vdots \\ 0 & \cdots & x \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

is backward stable : $\tilde{x}$ satisfies $\underbrace{(R+F)}_{\hat{R}} \tilde{x} = b$

All steps to compute $x = qr$ are backward stable

QR computes $\tilde{x}$ that solves exactly $\tilde{x} = \min \|\hat{A}\tilde{x} - \hat{b}\|$ and will always produce small $r_0 = Q^T (A\tilde{x} - b)$, and is as accurate

as it could be → $\frac{\|\tilde{x} - x\|}{\|x\|} = O(u) \cdot (\kappa_{LS,A} + \kappa_{LS,b})$.

The same results holds for x-svd:

will have backward errors of order $\|A\| = \|S\|$ times $O(u)$

→ $\tilde{x}$ solves $\min_{\tilde{x}} \|\hat{A}\tilde{x} - \hat{b}\|$ exactly

This does not hold for normal equations:

$M = A^T A$ — non-backward stable steps → larger error

$c = A^T b$ — the error $\frac{\|\tilde{x} - x\|}{\|x\|}$ is always going to be of

$x_{NE} = M \backslash c$ — magnitude at least $\kappa(A)\, O(u) = \kappa(A)^2 O(u)$

$\kappa(A)^2$ can be much larger than $\kappa_{LS,A} + \kappa_{LS,b}$

	NE	QR	SVD
$m \approx n$	$4/3\, n^3$	$4/3\, n^3$	$\approx 13\, n^3$
$m \gg n$	mn^2	$2mn^2$	$2mn^2$
	↑ unstable, esp. when θ is small	↑ backward stable	↑ gives info on distance from instability, works better on almost-singular problems

Matlab:

$x = A \backslash b$ uses QR.

LU factorization : it's a factorization

$A = LU \qquad A \in \mathbb{R}^{m \times m}$

- L: lower triangular with 1 on the diagonal
- U: upper triangular

Idea: follow the plan of QR factorization, but use lower triangular matrices instead of Householder reflector.

This produces essentially Gaussian elimination:

$$A = \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}$$

$$\text{row } i = \text{row } i - \frac{A_{i1}}{A_{11}} (\text{row } 1)$$

eliminate entries by summing to row $i = 2, 3, \dots, m$ a multiple of row 1

$$A_2 = L_1 A_1$$

$$\begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -a_2 & 1 & 0 \\ -a_3 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}$$

$$a_i = \frac{A_{i1}}{A_{11}}$$

multiplying on the left by L_1 $\longleftrightarrow$ one step of G.E.

second step of G.E.

$$\begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -b_3 & 1 \end{bmatrix} \cdot \begin{bmatrix} x & x & x \\ 0 & x & x \\ 0 & x & x \end{bmatrix}$$

$$b_i = \frac{(A_2)_{i2}}{(A_2)_{22}} \qquad A_2$$

Successive steps: multiplying on the left by matrices of the form

$$L_k = \begin{bmatrix} 1 & & & \\ & \ddots & & \\ & x & \ddots & \\ & x & & 1 \end{bmatrix}$$

$\uparrow$ non zero entries only in the k-th column

$$U = L_3 A_3 = L_3 L_2 A_2 = \underbrace{L_3 L_2 L_1 A}$$

lower tr. with ones on diagonal

$\downarrow$ invertible

$$L_1^{-1} L_2^{-1} L_3^{-1} U = A$$

lower triangular matrix — upper triangular

Update at each step:

$$U_{k+1:end,\,k+1:end} = U_{k+1:end,\,k+1:end} - \begin{bmatrix} L_{k+1,k} \\ \vdots \\ L_{m,k} \end{bmatrix} \cdot \left[U_{k,k+1} \,|\, \dots \,|\, U_{k,m} \right]$$

$(m-k) \times (m-k)$

$(m-k)^2$ subtractions

$(m-k)^2$ products

rank-1 matrix

Total cost: $2(m-1)^2 + 2(m-2)^2 + \dots + 2\cdot 2^2 + 2\cdot 1^2 = 2\sum_{k=1}^{m-1} k^2 = \frac{2}{3} m^3 + O(m^2)$

$\frac{2}{3}$: half as much as QR

If we never encounter divisions by zero, this produces a factorization $A = LU$ for every $A \in \mathbb{R}^{m \times m}$.

We can use this factorization to solve linear systems:

$$Ax = b \longrightarrow LUx = b \longrightarrow \begin{cases} Ux = y \\ Ly = b \end{cases}$$

(y = Ux) this form can be solved

With substitutions:

① Compute $A = LU$ — $\frac{2}{3} m^3$

② solve $Ly = b$ for y $\quad m^2$

③ solve $Ux = y$ for x $\quad m^2$

equivalent to writing

$$x = A^{-1}b = (LU)^{-1}b = U^{-1}(L^{-1}b)$$

LU factorization is NOT stable

$$\begin{bmatrix} 10^{-30} & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & -1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 10^{30} & 1 & 0 \\ 10^{30} & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 10^{-30} & 1 & 1 \\ 0 & 1-10^{30} & 1-10^{30} \\ 0 & 1-10^{30} & -1-10^{30} \end{bmatrix}$$

becomes exactly singular when you approximate it with floating point numbers

Stability analysis:

computed $\tilde{L}$, $\tilde{U}$ are s.t. $\tilde{L}\tilde{U} = A + \delta$ $\quad \|\delta\| = O(u)\|L\| \cdot \|U\|$

may both be much larger than $\|A\|$

Fix: pivoting: exchange rows at each time step to ensure pivots are as much large as possible

Step 1: swap rows to bring the largest value in the 1° column to (1,1)

$A = \begin{bmatrix} \times & \times & \times \\ \times & \times & \times \\ \times & \times & \times \end{bmatrix}$, then proceed with step 1 of G.E. LU factorization with pivoting is stable most of the time!

Pivoting ensures $|l_{ij}| \leq 1$

However, $\|U\| / \|A\|$ can still grows exponentially $\sim 2^m$

LU with pivoting is still the go-to algorithm to solve square linear problems. With QR, instead, $\|Q\| = 1$, $\frac{\|R\|}{\|A\|} = 1$ but it costs 2x times.

Note: one cannot use LU for least-squares problems because multiplying by 2 does change norms.

LU factorization + sparse matrices:

fill in: entries in A that were zero get filled in.

Direct algorithms for symmetric linear systems

Symmetric matrix: can we take advantage of symmetry

$Mx = b$, $M = M^T$

$L_1 M$ $\quad L_1 = \begin{bmatrix} 1 & & \\ x & \ddots & \\ x & & 1 \end{bmatrix}$

trick: to keep symmetry, instead of multiplying $L_1 M$, compute

$L_1 M L_1^T$ $\quad (L_1 M L_1^T) = L_1 M L_1^T =$

$\begin{bmatrix} x & 0 & 0 & 0 \\ 0 & x & & x \\ \vdots & x & \ddots & \vdots \\ 0 & x & \cdots & x \end{bmatrix}$

by symmetry, these are zeros

Similarly, after more steps,

$L_2 (L_1 M L_1^T) L_2^T = \begin{bmatrix} x & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & x & \cdots & x \\ \vdots & & \vdots & & \vdots \\ 0 & \cdots & x & \cdots & x \end{bmatrix}$

$L_{n-1} L_{n-2} \cdots L_2 L_1 M L_1^T L_2^T \cdots L_{n-2}^T L_{n-1}^T = D$ → diagonal

$M = \underbrace{L_1^{-1} L_2^{-1} \cdots L_{n-1}^{-1}}_{\begin{bmatrix} 1 & & 0 \\ x & \ddots & \\ x & x & 1 \end{bmatrix}} D \underbrace{(L_{n-1}^T)^{-1} (L_{n-2}^T)^{-1} \cdots (L_1^T)^{-1}}_{L^T} = L D L^T$

L: lower tr. with 1 on diagonal; D: diagonal

Updates to get the next matrix:

$$(M_k)_{k:m,\,k:m} = (M_{k-1})_{k:m,\,k:m} - (M_{k-1})_{k:m,\,k-1} \frac{1}{(M_{k-1})_{k-1,\,k-1}} \cdot (M_{k-1})_{k,\,k:m}$$

3 symmetric matrices

$$M = \begin{bmatrix} M_{11} & M_{12} & M_{13} & M_{14} \\ M_{21} & \multicolumn{3}{l}{\multirow{3}{*}{$M_{2:4,2:4}$}} \\ M_{31} & & & \\ M_{41} & & & \end{bmatrix}$$

$$M^{(2)}_{2:4,2:4} \leftarrow M^{(1)}_{2:4,2:4} - \begin{bmatrix} \frac{M^{(1)}_{21}}{M^{(1)}_{11}} \\ \frac{M^{(1)}_{31}}{M^{(1)}_{11}} \\ \frac{M^{(1)}_{41}}{M^{(1)}_{11}} \end{bmatrix} \begin{bmatrix} M^{(1)}_{12}, & M^{(1)}_{13} & M^{(1)}_{14} \end{bmatrix}$$

Since all matrices are symmetric, one can compute the lower triangle, and get the upper triangle from symmetry. Computational cost halved! $\frac{2}{3} n^3$ LU

$\frac{1}{3} n^3$ LDL^T

We need again pivoting for stability. In order to keep symmetry one must swap columns accordingly:

swap row 1-3. → × × ×

× × ×

and column 1-3. → × × ×

Th: if M is positive definite, then

- no zero pivots
- factorization is stable

To prove it, we need two easy properties of P.D. matrices:

1) $M > 0 \longrightarrow M_{kk} > 0 \quad \forall k = 1, 2, \dots, n$

$$M_{kk} = [0 \dots 0 \underset{k}{1} 0 \dots 0] \, M \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = e_k^T M e_k > 0$$

Recap: $M > 0$ means $x^T M x > 0 \quad \forall x \neq 0$

2) $M > 0 \longrightarrow B M B^T > 0 \quad \forall B$ invertible $M, B \in \mathbb{R}^{n \times n}$

$x \neq 0 \qquad x^T (B M B^T) x = y^T M y > 0$

$y = B^T x \neq 0$

Properties 1 + 2 ensure that all matrices encountered during LDL are P.D., and all pivots are > 0.

In particular, $D_{kk} > 0 \ \forall k$. The LDL^T factorization for a P.D. matrix can be written in a different format:

$$D = \begin{bmatrix} d_{11} & & \\ & d_{22} & \\ & & \ddots \end{bmatrix} \quad \begin{bmatrix} \sqrt{d_{11}} & & \\ & \sqrt{d_{22}} & \\ & & \ddots \end{bmatrix} \quad \begin{bmatrix} \sqrt{d_{11}} & & \\ & \sqrt{d_{22}} & \\ & & \ddots \end{bmatrix}$$

$$\begin{bmatrix} \ddots & & \\ & d_{mm} \end{bmatrix} = \begin{bmatrix} \ddots & & \\ & \sqrt{d_{mm}} \end{bmatrix} \cdot \begin{bmatrix} \sqrt{d_{22}} & & \\ & \ddots & \\ & & \sqrt{d_{mm}} \end{bmatrix}$$

$$M = LDL^T = \underbrace{LD^{1/2}}_{R^T} \underbrace{D^{1/2}L^T}_{R} = R^T R \qquad R \text{ upper triangular}$$

Th: Every P.D. $M \in \mathbb{R}^{m \times m}$ can be written as $M = R^T R$, where $R^T \in \mathbb{R}^{m \times m}$ is upper triangular. **Cholesky factorization**

Recall: from LS problem: $A \in \mathbb{R}^{m \times n}$, $A = Q_0 R_0 \longrightarrow A^T A = R_0^T R_0$

$$\sigma_i(R) = \sigma_i(A^T A)^{1/2} \longrightarrow \|R\| = \|M\|^{1/2}$$ norm of R doesn't grow!

We can use LDL^T and $R^T R$ to solve linear systems:

$$Mx = b \qquad M = LDL^T$$

$$x = M^{-1} b = (L^T)^{-1} D^{-1} L^{-1} b$$

$$\begin{cases} y = L^{-1} b \longrightarrow Ly = b \\ z = D^{-1} y \longrightarrow Dz = y \\ x = (L^T)^{-1} z \longrightarrow L^T x = z \end{cases}$$

$1/3\, n^3$ for the factorization + $O(n^2)$ for the substitution

How to choose an algorithm to solve $Ax = b$?

A pos. def → use Cholesky (LDL^T)

A symmetric → use LDL^T + pivoting

} $\frac{1}{3} n^3$

A non symmetric → use LU

} $\frac{2}{3} n^3$

If A is sparse, one can use sparse versions of these 3 algorithms