M. Sombra, 2020

NLA V. The least square problem

Let A $m\times n$-matrix and b m-vector.
The least square problem (LSP): find x n-vector minimizing

$$\|Ax - b\|_2$$

If $m=n$ and A is nonsingular then x is the solution of $Ax=b$

If $m>n$ then typically $Ax=b$ has no solution.
The LSP gives the linear combination of the columns of A

$$A\underline{x} = x_1 \operatorname{col}_1(A) + \cdots + x_n \operatorname{col}_n(A) \in \mathbb{R}^m$$

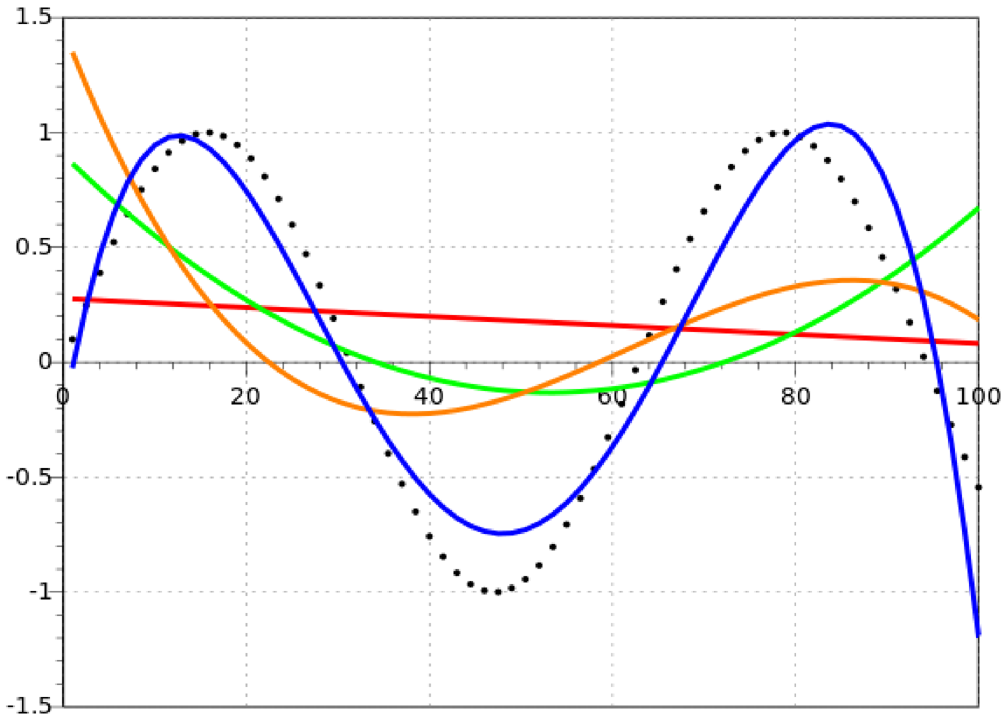
that best approaches (in the 2-norm) the vector b.

Example: curve fitting.

Suppose we have pairs $(y_i, b_i) \in \mathbb{R}^2$, $i=1,\dots,m$ and we want to find the "best" degree d polynomial fitting b_i as a function of y_i: find $x_0,\dots,x_d \in \mathbb{R}$ st

$$p(y) = \sum_{j=0}^{d} x_j y^j$$

minimizes the residual $p(y_i) - b_i$, $i=1,\dots,m$



Minimizing $\sum_{i=1}^{m} (p(y_i) - b_i)^2$ is a LSP for

$$A = \begin{pmatrix} 1 & y_1 & \cdots & y_1^d \\ \vdots & & & \\ 1 & y_m & \cdots & y_m^d \end{pmatrix} \in \mathbb{R}^{m \times (d+1)} \qquad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix} \in \mathbb{R}^m$$

There are three methods for solving the LSP:

(1) normal equations

(2) QR factorization

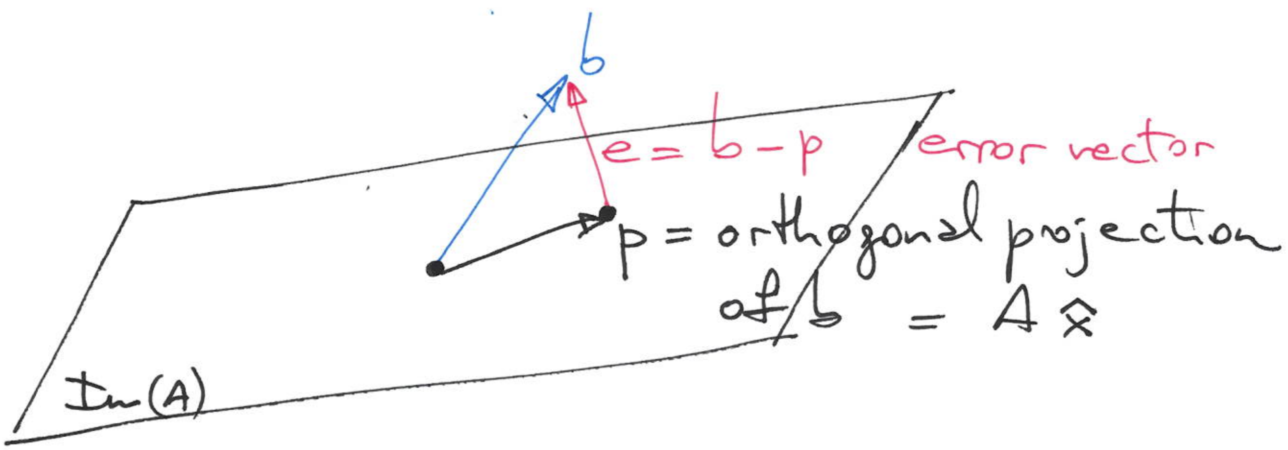
(3) SVD

We will consider the central case:

$m \geq n$ and $\operatorname{rank}(A) = n$

II.1 Normal equations [D, § 3.2.1]

The LSP and its solution $\hat{x}$:



e is perpendicular to the linear subspace $\mathrm{Im}(A)$
Hence for all $x \in \mathbb{R}^n$

$$0 = \langle Ax, e \rangle = (Ax)^T (b - A\hat{x}) = x^T A^T (b - A\hat{x})$$

Since this holds for all x

$$\Longrightarrow \quad A^T (b - A\hat{x})$$

The <u>normal equation</u> for $\hat{x}$:

$$\boxed{A^T A \hat{x} = A^T b} \qquad (*)$$

The $n \times n$-matrix $A^T A$ is <u>symmetric</u> and <u>positive definite</u> (check it!)

In particular, it is <u>nonsingular</u>, and $\hat{x}$ is the <u>only</u> solution of $(*)$

By Pythagoras, the 2-norm of the error is

$$\| e \| = \left(\|b\|^2 - \|Ax\|^2 \right)^{1/2}$$

We can apply <u>Cholesky factorization</u> of A^TA to solve the normal equation:

<u>Normal equation algorithm</u>:

1) Compute (the lower triangular part of) $C = A^TA$
2) Compute $d = A^Tb$
3) Compute the Cholesky factorization $C = GG^T$
4) Solve $Gy = d$ and $G^T\hat{x} = y$

It is <u>convenient</u> since relies on standard algorithms.
The complexity is $(m + \frac{n}{3})n^2 + O(n^2)$ flops
For $m \gg n$, the complexity

$$mn^2$$

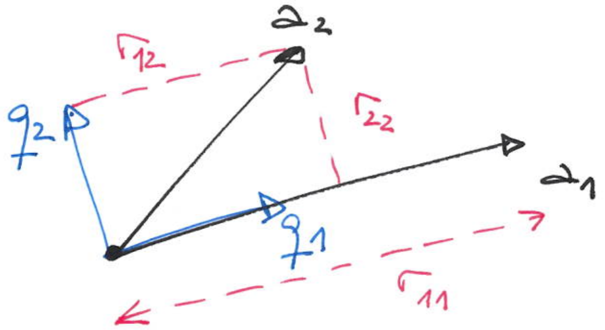
of <u>computing A^TA dominates.</u>
It is <u>reliable</u> when A is far from rank deficient, but unstable otherwise (more about this later).

V.2 QR factorization [D, § 3.2.2]

The columns of A are assumed to be independent ($\mathrm{rank}(A) = n$) but not to be orthogonal!

After orthogonalizing these columns, $\hat{x}$ is easy to find.

Orthogonalization is done by the Gram-Schmidt algorithm:



Let $a_j = \mathrm{col}_j(A) \in \mathbb{R}^m$ $\quad j = 1, \dots, n$

GS produces q_j, $j = 1, \dots, n$, orthonormal s.t

$$\mathrm{Vect}(q_1, \dots, q_j) = \mathrm{Vect}(a_1, \dots, a_j) \qquad j = 1, \dots, n \qquad (*)$$

generated linear subspace

$\boxed{\text{1st step:}}$ $\quad q_1 \leftarrow \frac{a_1}{\|a_1\|}$ unit vector

$\boxed{\text{GO step}}$: Orthogonalize $\quad \tilde{a}_2 \leftarrow a_2 - \langle a_2, q_1 \rangle q_1$

Normalize $\quad q_2 \leftarrow \frac{a_2}{\|a_2\|}$

Indeed $a_2 - \langle a_2, q_1 \rangle q_1$ is orthogonal to q_1

and q_2 unit vector orthogonal to q_1 $\Big\}$ prove it!

$\boxed{\text{GO step}}$ $$\tilde{a}_3 \leftarrow a_3 - \langle a_3, q_1\rangle q_1 - \langle a_3, q_2\rangle q_2$$

$$q_3 \leftarrow \frac{\tilde{a}_3}{\|\tilde{a}_3\|}$$

etc: the pseudocode organizing GO can be found in [D, page 107].

We can write the a's in terms of the q's:

$$a_1 = \|a_1\| \, q_1$$

$$a_2 = \langle a_2, q_1\rangle q_1 + \|\tilde{a}_2\| \, q_2$$

$$a_3 = \langle a_3, q_1\rangle q_1 + \langle a_3, q_2\rangle q_2 + \|\tilde{a}_3\| \, q_3$$

etc

Hence

$$\begin{pmatrix} a_1 & a_2 & a_3 & \cdots \end{pmatrix} = \begin{pmatrix} q_1 & q_2 & q_3 & \cdots \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & \cdots \\ 0 & r_{22} & r_{23} & \\ 0 & 0 & r_{33} & \\ \vdots & & & \ddots \end{pmatrix}$$

$$\boxed{A = Q \cdot R}$$

$n \times n$ upper triangular with positive diagonal entries

$m \times n$ - orthogonal: $Q^T \cdot Q = \mathbb{1}_n$

This factorization is unique (it is equivalent to the conditions (*) in page 5).

The r_{ij} can be computed in terms of scalar products from the a's and the q's:

$$r_{ij} = \langle q_i, a_j \rangle$$

The QR factorization solves the LSP: by the normal equation

$$\hat{x} = (A^T A)^{-1} A^T b$$
$$= ((QR)^T QR)^{-1} (QR)^T b$$
$$= (R^T R)^{-1} R^T Q b \qquad \text{(since } Q^T \cdot Q = \mathbb{1}_n\text{)}$$
$$\boxed{= R^{-1} Q^T b}$$

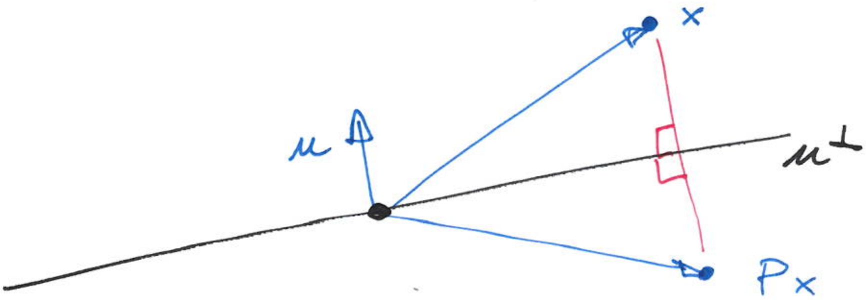
The GO algorithm is not stable if the columns of A are close to linearly dependent (i.e. if A is close to rank deficient).

VII Householder reflections [D, § 3.4.1]

A Householder reflection is

$$P = \mathbb{1}_m - 2 u u^T \in \mathbb{R}^{m \times m} \quad \text{for } u \in \mathbb{R}^m \text{ unit vector}$$

It is the reflexion with respect to the hyperplane $u^\perp$



P is symmetric ($P^T = P$) and orthogonal ($P^T P = \mathbb{1}_m$). prove it!

Given $y \in \mathbb{R}^m$ there is a reflexion that <u>zeroes</u> <u>all but the first entry</u>

$$Py = \begin{pmatrix} c \\ 0 \\ \vdots \\ 0 \end{pmatrix} = c \cdot e_1 \in \mathbb{R}^m$$

1st vector in the standard basis

Since P is orthogonal:

$$|c| = \|Py\| = \|y\|$$

To compute u:

$$Py = (\mathbb{1}_m - 2uu^T)y = y - 2\langle u, y\rangle u = \pm \|y\| e_1$$

Then $2\langle u, y \rangle u = y \pm \|y\| e_1$

choose $\mathrm{sign}(x_1)$ to avoid cancellation

then u is a scalar multiple of

$$\tilde{u} = y \pm \|y\| e_1 = \begin{pmatrix} y_1 + \mathrm{sign}(y_1)\|y\| \\ y_2 \\ \vdots \\ y_m \end{pmatrix}$$

and so $u = \mathrm{House}(y) = \dfrac{\tilde{u}}{\|\tilde{u}\|}$

We show how to compute the QR factorization using Householder reflexions when $m=4$ and $n=3$:

$$A = \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{pmatrix}$$

1) Choose P_1 s.t. $\quad A_1 \leftarrow P_1 \cdot A = \begin{pmatrix} * & & \\ 0 & & \\ 0 & & \\ 0 & & \end{pmatrix}$

2) Choose $P_2 = \left(\begin{array}{c|c} 1 & \underline{0} \\ \hline \underline{0} & P_2' \end{array}\right)$ s.t. $\quad A_2 \leftarrow P_2 A_1 = \begin{pmatrix} * & * & * \\ 0 & * & * \\ 0 & 0 & * \\ 0 & 0 & * \end{pmatrix}$

3) Choose $P_3 = \left(\begin{array}{cc|c} 1 & & 0 \\ & 1 & \\ \hline 0 & & P_3' \end{array}\right)$ s.t. $\quad A_3 \leftarrow \begin{pmatrix} * & * & * \\ 0 & * & * \\ 0 & 0 & * \\ 0 & 0 & 0 \end{pmatrix}$

Then

$$P_3 P_2 P_1 A = \tilde{R} \; (= A_4) \quad \text{upper triangular}$$

Hence

$$A = P_1^T P_2^T P_3^T \tilde{R} = Q \cdot R$$

first three rows of $\tilde{R}$

first three columns of $P_1^T P_2^T P_3^T$

Example Let $m = 3$, $n = 2$

$$A = \begin{pmatrix} 1 & -3 \\ 0 & 2 \\ -1 & -1 \end{pmatrix}$$

Set $\tilde{u}_1 = \begin{pmatrix} 1+\sqrt{2} \\ 0 \\ -1 \end{pmatrix}$ and $u_1 = \text{House}\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} = \frac{\tilde{u}_1}{\|\tilde{u}_1\|} = \begin{pmatrix} 0.92 \\ 0 \\ -0.38 \end{pmatrix}$

Then $P_1 = \mathbb{1}_3 - 2\, u_1 u_1^T = \begin{pmatrix} -0.71 & 0 & 0.71 \\ 0 & 1 & 0 \\ 0.71 & 0 & 0.71 \end{pmatrix}$

and $A_1 = P_1 \cdot A = \begin{pmatrix} -1.41 & 1.41 \\ 0 & 2 \\ 0 & -2.83 \end{pmatrix}$

Set $\tilde{u}_2 = \begin{pmatrix} 2 + (2^2 + (-2.83)^2)^{1/2} \\ -2.83 \end{pmatrix}$ and $u_2 = \text{House}\begin{pmatrix} 2 \\ -2.83 \end{pmatrix} = \frac{\tilde{u}_2}{\|\tilde{u}_2\|} = \begin{pmatrix} 0.89 \\ -0.46 \end{pmatrix}$

Then $P_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \multicolumn{2}{c}{\underbrace{P_2'}_{\mathbb{1}_2 - 2\, u_2 u_2^T}} \\ 0 & & \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -0.58 & 0.82 \\ 0 & 0.82 & 0.58 \end{pmatrix}$

and

$$P_2 \cdot A_1 = \underbrace{\begin{pmatrix} -1.41 & 1.41 \\ 0 & -3.49 \\ 0 & 0 \end{pmatrix}}_{\tilde{R}}$$

We have that

$$A = P_1^T P_2^T \cdot \tilde{R} = \overbrace{\begin{pmatrix} -0.71 & 0.58 & 0.41 \\ 0 & -0.58 & 0.82 \\ 0.71 & 0.58 & 0.41 \end{pmatrix}}^{\tilde{Q}} \overbrace{\begin{pmatrix} -1.41 & 1.41 \\ 0 & -3.47 \\ 0 & 0 \end{pmatrix}}^{\tilde{R}}$$

$$= \underbrace{\begin{pmatrix} -0.71 & 0.58 \\ 0 & -0.58 \\ 0.71 & 0.58 \end{pmatrix}}_{Q} \underbrace{\begin{pmatrix} -1.41 & 1.41 \\ 0 & -3.49 \end{pmatrix}}_{R}$$

QR factorization with Householder reflections

for $i = 1$ to $\min(m-1, n)$

$u_i \leftarrow \text{House}\ A(i:m, i)$

$P_i \leftarrow \mathbb{1}_{m-i+1} - 2 u_i u_i^T$

$A_i(i:m, i:n) \leftarrow P_i' A(i:m, i:n)$

end for

To implement it: we dont really need P_i' explicitly but just the multiplication:

$$(\mathbb{1}_{m-i+1} - 2 u_i u_i^T)\, A(i:m, i:n)$$

$$= A(i:m, i:n) - 2u_i\left(u_i^T (A\, i:m, i:n)\right)$$

The complexity of this algorithm is

$$2n^2 m - \frac{2}{3} n^2 \text{ flops} \qquad (*)$$

about twice the complexity of solving the normal equations via Cholesky algorithm

for $m \gg n$

(*) if the product $Q = P_1^T P_2^T \cdots$ is not required.

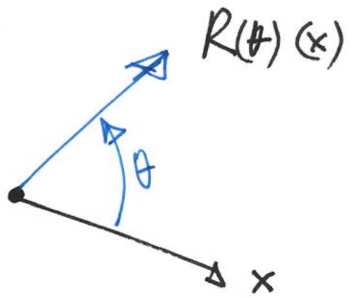
V.3 Givens rotations [D, § 3.4.2]

A rotation on the plane with angle θ is a linear map

$$R(\theta): \mathbb{R}^2 \longrightarrow \mathbb{R}^2$$

given by the orthogonal matrix

$$\begin{pmatrix} \cos\theta, & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$



A Givens rotation is ~~the~~ the orthogonal matrix $n \times n$ giving a rotation in the (i,j)-plane of $\mathbb{R}^n$:

$$R(i,j,\theta) = \begin{array}{c c} & \begin{array}{ccccccc} & & i & & j & & \end{array} \\ \begin{array}{c} \\ \\ i \\ \\ j \\ \\ \end{array} & \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & 1 & & & & \\ & & & \cos\theta & & -\sin\theta & \\ & & & & 1 & & \\ & & & & & \ddots & \\ & & & \sin\theta & & 1 \ \cos\theta & \\ & & & & & & 1 \ \ddots \\ & & & & & & \ddots \ 1 \end{pmatrix} \end{array}$$

The QR factorization can be computed with Givens rotations similarly as with Householder rotations, zeroing one entry at a time.

given x, i and j, we can zero out x_j by choosing θ such that

$$\begin{pmatrix} \cos(\theta) & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x_i \\ x_j \end{pmatrix} = \begin{pmatrix} (x_i^2 + x_j^2)^{1/2} \\ 0 \end{pmatrix}$$

or equivalently

$$\cos\theta = \frac{x_i}{(x_i^2 + x_j^2)^{1/2}} \quad \text{and} \quad \sin\theta = \frac{-x_j}{(x_i^2 + x_j^2)^{1/2}}$$

(inverse trigonometric functions are not needed)

Example: $m = 3$, $n = 2$ and

$$A = \begin{pmatrix} 1 & -3 \\ 0 & 2 \\ -1 & -1 \end{pmatrix}$$

Set

$$R_1 = \begin{pmatrix} 0.71 & 0 & -0.71 \\ 0 & 1 & 0 \\ 0.71 & 0 & 0.71 \end{pmatrix}$$

Then $A_1 = R_1 \cdot A = \begin{pmatrix} 1.41 & -1.41 \\ 0 & 2 \\ 0 & -2.82 \end{pmatrix}$

Set $$R_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0.58 & -0.82 \\ 0 & +0.82 & 0.58 \end{pmatrix}$$

and so $$A_2 = R_2 \cdot A_1 = \begin{pmatrix} 1.41 & -1.41 \\ 0 & 3.47 \\ 0 & 0 \end{pmatrix} = \tilde{R}$$

We conclude that

$$A = R_1^T R_2^T \cdot \tilde{R} = \underbrace{\begin{pmatrix} 0.71 & -0.58 \\ 0 & 0.58 \\ 0.71 & -0.58 \end{pmatrix}}_{Q} \cdot \underbrace{\begin{pmatrix} 1.41 & -1.41 \\ 0 & 2 \end{pmatrix}}_{R}$$

The complexity of the QR factorization using Givens rotations is ~~twice~~ $\frac{3}{2}$ times the complexity using Householder reflections.

It is useful for special situations, for instance in the Hessenberg case:

$$A = \begin{pmatrix} * & & & \\ * & * & & \text{\huge *} \\ & * & \ddots & \\ & \text{\huge 0} & \ddots & \ddots \\ & & & * \;\; * \end{pmatrix}$$

Solving the LSP with QR factorization (using Householder or Givens) is <u>more numerically stable</u> than solving the normal equation when A is close to rank deficient.

V.4 Normal equations vs QR factorization

First recall the definition of the condition number of

$$A \in F^{n \times n} \quad \text{nonsingular} \quad (F = \mathbb{R} \text{ or } \mathbb{C})$$

with respect to the <u>2-norm</u>:

$$K_2(A) = \|A^{-1}\|_2 \cdot \|A\|_2$$

and

$$\|A\|_2 = \left(\lambda_{max}(A^* \cdot A)\right)^{1/2}$$

largest eigenvalue

Hence

$$K_2(A) = \left(\frac{\lambda_{max}(A^* A)}{\lambda_{min}(A^* A)}\right)^{1/2}$$

For

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

we define its <u>condition number</u> as

$$K_2(A) := K_2(A^T \cdot A)^{1/2} = \left(\frac{\lambda_{max}(A^T \cdot A)}{\lambda_{min}(A^T \cdot A)}\right)^{1/2}$$

The <u>forward error analysis</u> (= sensitivity to perturbations) of the LSP is <u>controlled by $K_2(A)$</u>, see [D, § 3.3] for details.

QR factorization via Householder or Givens is backward stable: if

$$A = Q \cdot R$$

and $Q + \delta Q$ and $R + \delta R$ are the round offs of Q and R, then

$$A + \delta A = (Q + \delta Q) \cdot (R + \delta R)$$

with

$$\frac{\|\delta A\|_2}{\|A\|_2} \leq n \cdot \varepsilon$$

machine epsilon

relative error

see [D, § 3.4.3] for details.

Hence QR factorization via Householder or Givens solves the LSP with a loss of precision of

$$\approx \log_b(\kappa_2(A)) \text{ digits}$$

(b = base of the floating point system)

with $\approx$ $2n^2m$ flops or $\approx$ $3n^2m$ flops resp. when $m \gg n$.

Solving the normal equations via Cholesky solves the LSP with a loss of precision of

$$\approx \log_b \kappa_2(A^T A) = \mathbf{2} \cdot \log_b \kappa_2(A) \text{ digits}$$

because

$$\kappa_2(A^T A) = \kappa_2(A)^2$$ (prove it!)

with $\approx n^2 m$ flops when $m \gg 0$

NE is the method of choice when A is well-conditionned, to solve the LSP
If A is badly conditionned, we should rather apply the QR factorization or the SVD

to be discussed later

V.5 Rank deficient LSP [D, §3.5]

Let A $m\times n$-matrix with

$$r = \mathrm{rank}(A)$$

In general, $r \leq n$

If $r<n$, the solution $\hat{x}$ to the LSP is not unique: $\forall\, x \in \mathrm{Ker}(A) \simeq \mathbb{R}^{n-r}$

$$A\cdot(\hat{x}+x) = A\hat{x} + Ax = A\hat{x}$$

also solves the LSP.

In data analysis, ~~usu~~ typically the data matrix A is close to rank deficient (better studied with SVD)

Example: Medical research on the effect of a drug on sugar levels in blood

We take the following data ~~from~~ patients:

initial blood level (sugar)

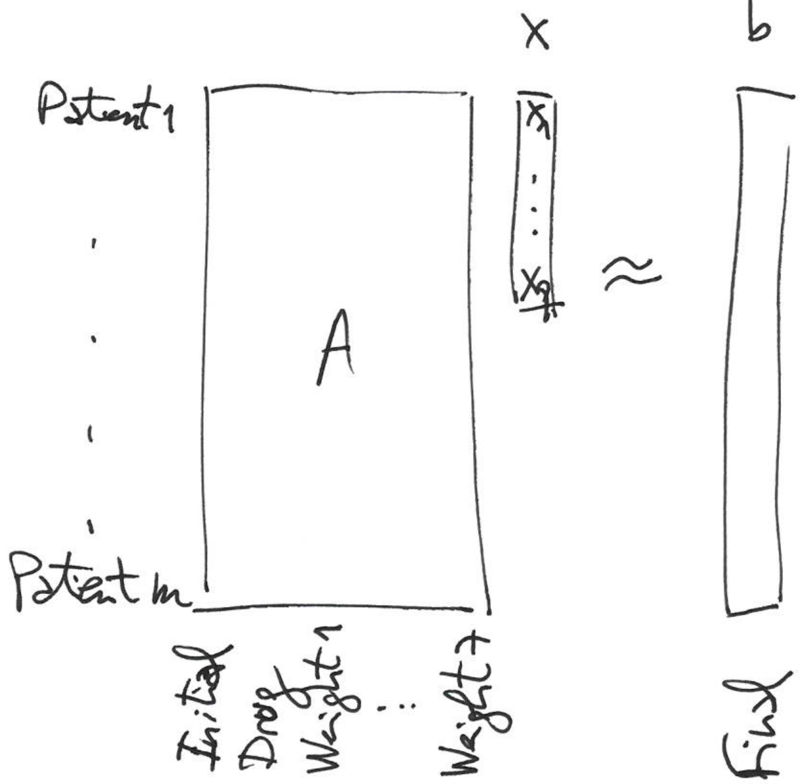
amount of drug

weight on day i, $i=1,\dots,7$

final blood level (sugar)

The aim is to predict the final blood level in terms of the rest of the data.

We consider the LSP



The solution $\hat{x}$ should predict

$$b_i \approx \text{Patient}_i \cdot \hat{x}$$

A is close to rank 3 : columns 3 – 9 "should" be identified (from knowledge of the problem)

Otherwise we might take $\hat{x}$ very large, and a patient changing weight during the week would receive a bad prediction.

We can solve a rank deficient LSP with QR:

if $r<n$

r×r non singular

$$A = QR = Q \cdot \begin{pmatrix} R_{11} & R_{12} \\ 0 & 0 \end{pmatrix} \quad r\times(n-r)$$

With roundoff, we hope to compute

$$R = \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix}$$

machine epsilon

with R_{22} $(n-r)\times(n-r)$-matrix that is small $(\approx \varepsilon \|A\|_2)$

In this case we set

$$R_{22} = 0$$

and minimize $\|Ax-b\|_2$ as follow:

complete Q to $(Q\ \tilde{Q})$ $m\times m$-orthogonal.

Then

an orthogonal map does not change the 2-norm

$$\|Ax-b\|_2^2 = \left\| \begin{pmatrix} Q^T \\ \tilde{Q}^T \end{pmatrix} (Ax-b) \right\|_2^2 = \|Rx - Q^Tb\|_2^2 + \|\tilde{Q}^Tb\|_2^2$$

Write $Q = (\underset{r}{Q_1}\ \underset{n-r}{Q_2})$ and $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \begin{matrix} r \\ n-r \end{matrix}$. Then

$$\|Ax-b\|_2^2 = \|R_{11}x_1 + R_{12}x_2 - Q_1^Tb\|_2^2 + \|Q_2^Tb\|_2^2 + \|\tilde{Q}^Tb\|_2^2$$

This is minimized by

$$x = \begin{pmatrix} R_{11}^{-1}\left(Q_1^T b - R_{12}x_2\right) \\ x_2 \end{pmatrix}$$

for any x_2 $(n-r)$-vector.

The typical choice is $\boxed{x_2 = 0}$

In general, it is not reliable since R might be close to rank deficient even if R_{22} is not small.

Instead, we should apply QR with pivoting:

$$A\underset{\text{permutation matrix}}{P} = QR$$

At step i, choose the largest column of A $(i \leq j \leq n)$

Then we compute the Householder reflection to zero in the i-th column, the entries $i+1, \dots, m$

This attempts to keep R_{11} well-conditioned and R_{22} small.