23/09

NUMERICAL APPROXIMATION

Parole chiave: Approssimazione, ottimizzazione, learning e statistica

Approssimazione: $v \in V$, $a_n \to v$, $a_n \in y \subset V$ $\left(\begin{matrix} \text{es } V = C[a,b] \\ y = \mathbb{R}_n[x] \end{matrix}\right)$

Ottimizzazione: $f: X \to \mathbb{R}$, $\min_{x \in X} f$ (ordinamento)

Learning: $f: X \to \{0,1\}$

Meglio però ragionare con $f: X \to [0,1]$ ← probabilità

Dal momento che è difficile trovare f, cerco una f_n che la approssima

---

24/09

Il problema più semplice è trovare una legge che rappresenti dei dati di un problema reale

Approssimazione: trovare una funzione che fa minimo errore

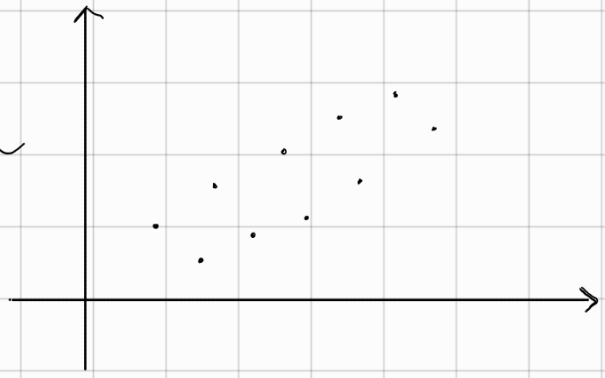
Ottimizzazione: trovare minimo

Esempio:

(SVD, approccio statistico) → è un problema di approssimazione

Voglio approssimare la matrice con una di rango più basso.

Minimi quadrati lineari:



Ci chiediamo quale sia la migliore retta che genera questi punti
Costruisco allora, scelta una retta $y = mx + q$,
l'errore che questa retta commette in ogni punto.

Creato il vettore errore ε, se ne minimizza la norma euclidea

$$\varepsilon_i = mx_i + q - y_i \qquad i = 1, \dots, n$$

$$\|\varepsilon\| = \sqrt{\sum_{i=1}^{n} |\varepsilon_i|^2}$$

Posso minimizzare $\|\varepsilon\|^2$ tanto se $f \geq 0$ i minimi coincidono.

$$f(m, q) = \sum_{i=1}^{n} |mx_i + q_i - y_i|^2$$

$$\frac{\partial f}{\partial m}(m, q) = \sum_{i=1}^{n} 2x_i (mx_i + q_i - y_i)$$

$$\frac{\partial f}{\partial q}(m, q) = \sum_{i=1}^{n} 2 (mx_i + q_i - y_i)$$

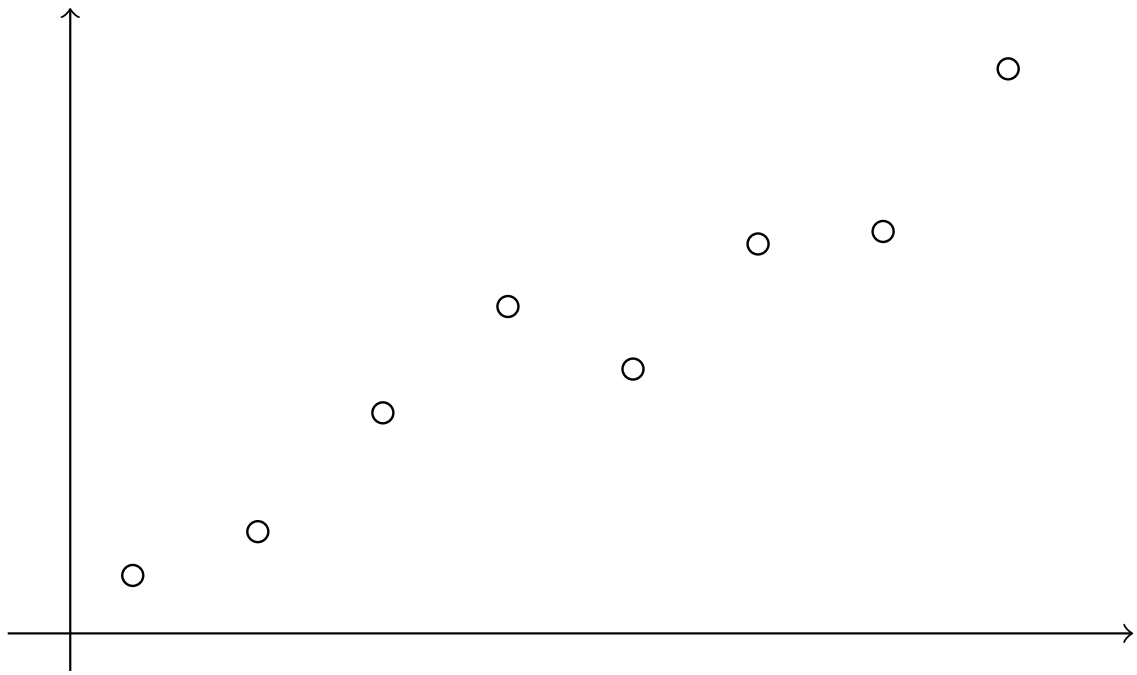
Poniamo $\frac{\partial f}{\partial m} = \frac{\partial f}{\partial q} = 0$

(i punti con $\frac{\partial f}{\partial m} = 0 = \frac{\partial f}{\partial q}$ si dicono stazionari, non necessariamente minimi)

Lo sono se vale il Th di Analisi 2.

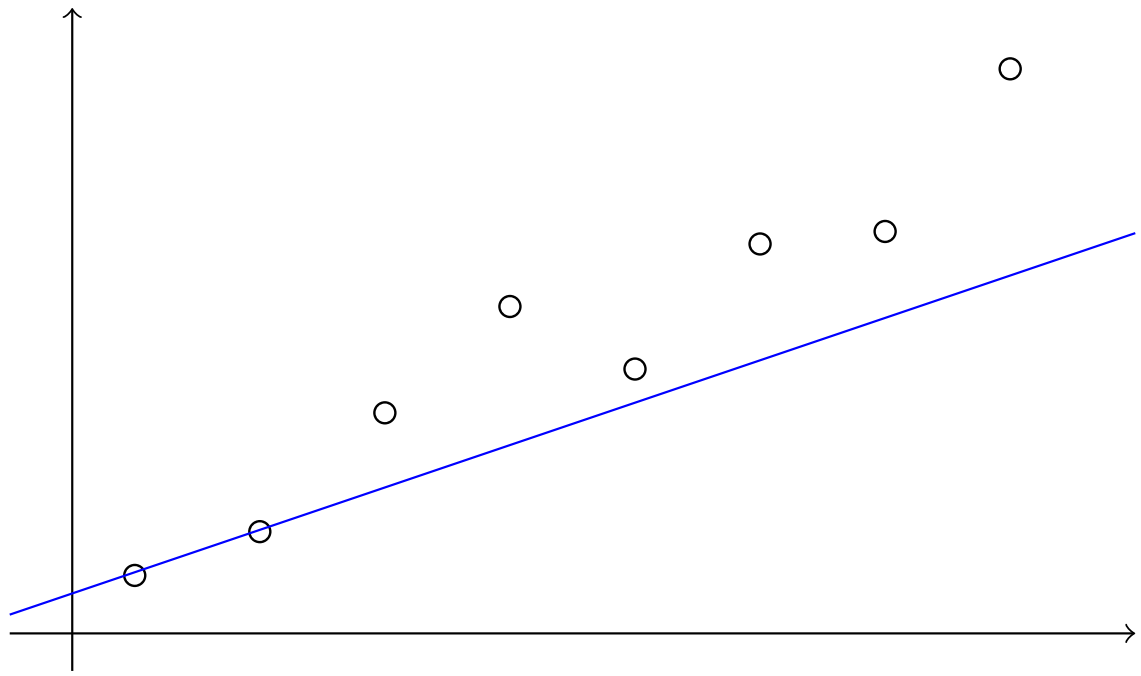
An approximation problem

"Fit" the following data. Do data hide a linear law with errors?



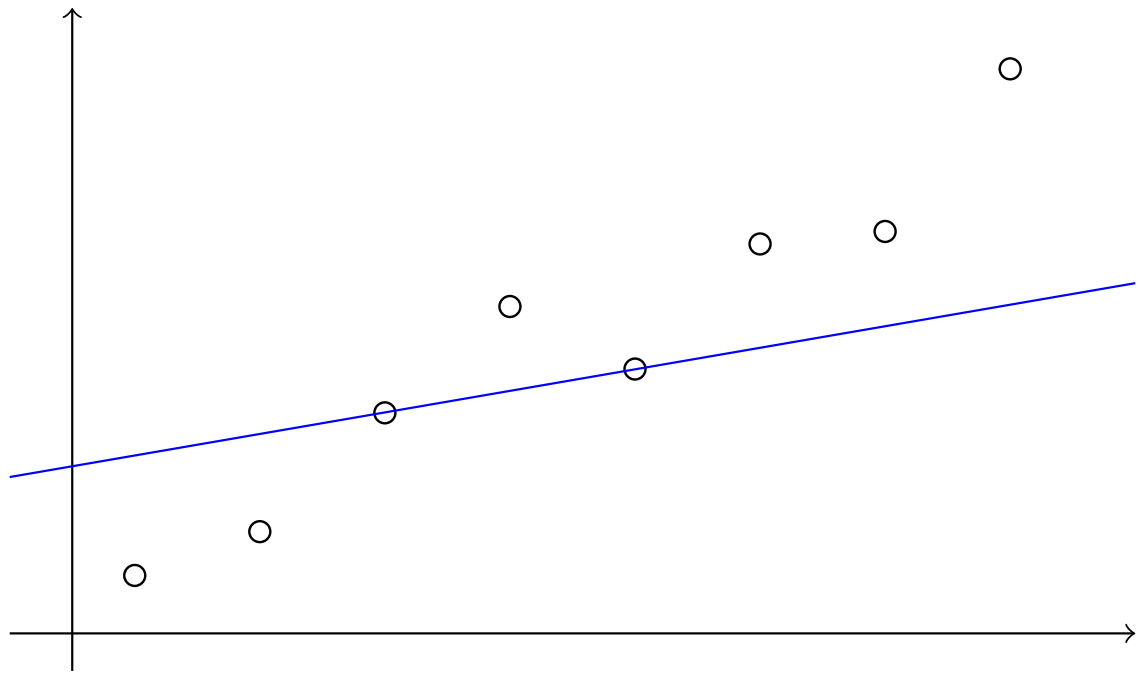
An approximation problem

"Fit" the following data. Do data hide a linear law with errors?



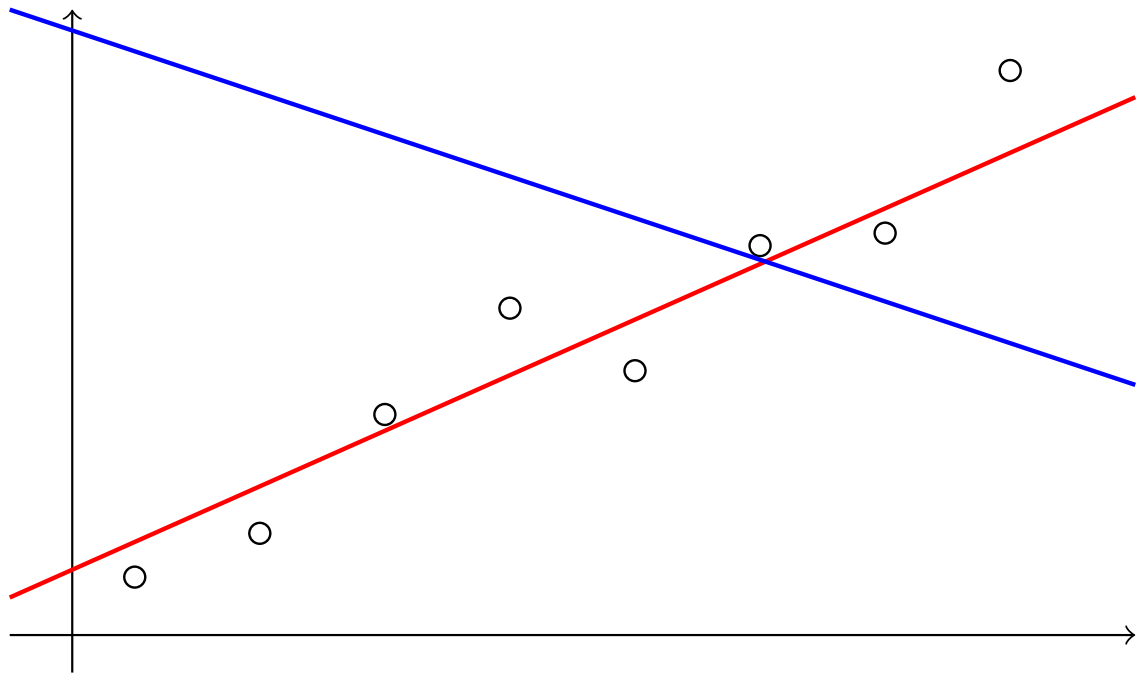
An approximation problem

"Fit" the following data. Do data hide a linear law with errors?



An approximation problem

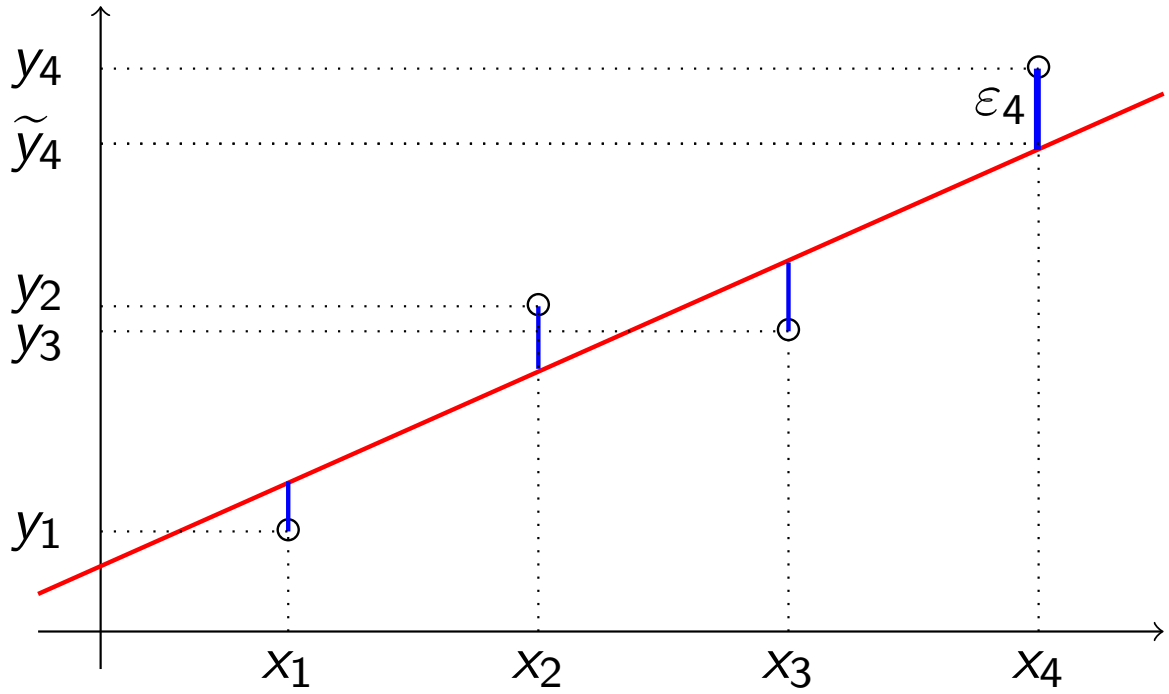
"Fit" the following data. Do data hide a linear law with errors?



First approach

Search the **"nearest"** line to the set of points.

We look for the function $y = mx + q$. Idea: keep the errors low in approximating y_i with $\widetilde{y}_i = mx_i + q$.



Define the (absolute) error on the data x_i as

$$\varepsilon_i(m, q) = \widetilde{y}_i - y_i = mx_i + q - y_i.$$

First approach

Define the (absolute) error on the data x_i as

$$\varepsilon_i(m, q) = \widetilde{y}_i - y_i = mx_i + q - y_i.$$

We get the vector $\varepsilon = \begin{bmatrix} \varepsilon_1 & \cdots & \varepsilon_n \end{bmatrix}^T$.

We have the following problem

$$\text{argmin}_{m,q\in\mathbb{R}} \|\varepsilon\| = \text{argmin}_{m,q\in\mathbb{R}} \left(\sum_{i=1}^{n} |mx_i + q - y_i|^2 \right)^{\frac{1}{2}},$$

or, equivalently

$$\text{argmin}_{m,q\in\mathbb{R}} \sum_{i=1}^{n} (mx_i + q - y_i)^2.$$

First approach

Let us minimize $f(m,q) := \sum_i (mx_i + q - y_i)^2$.

The Hessian matrix

$$H = 2\begin{bmatrix} \sum_i x_i^2 & \sum_i x_i \\ \sum_i x_i & n \end{bmatrix},$$

is positive semidefinite: $h_{11} \geqslant 0$, $h_{22} \geqslant 0$ and $\det(H) = n\sum_i x_i^2 - (\sum_i x_i)^2 \geqslant 0$ follows from the Cauchy-Schwarz inequality

$$\sum_i x_i = \langle e, x\rangle \leqslant \|e\|\|x\| = \sqrt{n\sum_i x_i^2}.$$

It is positive definite if and only if $x_i \neq x_j$ for some i and j.

$A \in \mathbb{C}^{n\times n}$ é definida positiva se $A^* = A$ e $v^*Av > 0$, $v \in \mathbb{C}^n \setminus \{0\}$

The function f is convex.

Teorema:
Se $A \in \mathbb{C}^{n\times n}$ è hermitiana, è definita positiva se e solo se vale uno dei seguenti

1. $\det A_\ell > 0$, $\ell = 1, \dots, n$ (A_ℓ minore principale di testa)
2. autovalori tutti positivi ($\neq 0$)
3. $\exists R$ triangolare superiore con elementi $\mathbb{R}^+$ sulla diagonale tale che $A = R^* R$ ($\operatorname{rank} A = \operatorname{rank} R$)
   (Fattorizzazione di Cholesky, simile a quella LU)
4. $\exists B \in \mathbb{C}^{n\times m}$ di rango pieno ($\operatorname{rg} B = \min\{n, m\}$) t.c. $A = B^* B$

Tornando a noi, usiamo il metodo dei minori

$2\sum x_i^2 > 0 \iff x_i$ non sono tutti nulli (nel nostro problema è un caso da considerare)

$$\det H = 2^n \left(n \sum x_i^2 - \left(\sum x_i\right)^2 \right)$$

Usiamo la disuguaglianza di Cauchy-Schwartz:
se scelgo $v = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$, $w = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \Rightarrow v^* w = \sum x_i$

$$\Rightarrow \left|\sum x_i\right| \leq \sqrt{n}\sqrt{\sum x_i^2}$$

$$\Rightarrow \left(\sum x_i\right)^2 \leq n \sum x_i^2$$

$\Rightarrow$ effettivamente quel det è > 0 (se $\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$ non è multiplo di $\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$, cioè se e solo se gli x_i non sono tutti coincidenti)

Quindi se i dati non sono tutti coincidenti allora $H > 0$ (usiamo questo simbolo per def positiva) $\Rightarrow$ i punti stazionari sono di minimo (e $H > 0$ implica anche che la funzione è convessa strettamente)

$$\nabla \; A \begin{bmatrix} m \\ q \end{bmatrix} = b$$

$$A = 2\begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & n \end{bmatrix}, \quad b = \begin{bmatrix} 2\sum x_i y_i \\ 2\sum y_i \end{bmatrix}$$

Le soluzioni del sistema $*$ sono i punti stazionari dato che $A = H$.

Se x_i non sono tutti coincidenti allora la matrice è invertibile $\Rightarrow$ c'è un solo punto stazionario che è minimo poiché $H > 0$

Se x_i sono coincidenti $\sum x_i^2 \geq 0$ e

$$2^n \left(n \sum x_i^2 - \left(\sum x_i \right)^2 \right) = 0$$

Criterio:

$A \in \mathbb{C}^{n \times m}$ hermitiana è semidefinita $\Leftrightarrow$ vale una di

1. $v^* A v \geq 0$
2. se i $\det A_k \geq 0$ per ogni minore principale (non per forza di testa)
3. tutti gli autovalori sono ≥ 0
4. $\exists B \in \mathbb{C}^{n \times m}$ t.c. $A = B^* B$

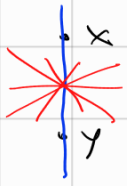
Nel nostro caso controllando anche $n > 0$ abbiamo che H è semidefinita positiva $\Rightarrow$ anche se x_i non sono tutti coincidenti H è semidefinita positiva $\Rightarrow$ tutti i punti stazionari sono di minimo debole.

Che accade se invece della norma euclidea, che risente della grandezza dell'errore, prendessi un'altra norma?

Si può usare qualsiasi norma → stesso che i calcoli sono più difficili, non sappiamo qual è la migliore

Potrei cercare il minimo di $\sum \delta^2\left(\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, r\right)$ con r retta

Sono problemi diversi perché se x e y sono nella stessa ascissa il secondo problema trova solo una soluzione e l'altro problema ne trova infinite



First approach

The gradient is zero if

$$\frac{\partial f}{\partial m} = \sum_{i=1}^{n} 2x_i(mx_i + q - y_i) = 0,$$

$$\frac{\partial f}{\partial q} = \sum_{i=1}^{n} 2(mx_i + q - y_i) = 0.$$

This leads to the linear system

$$M\widetilde{x} = c, \quad M = \begin{bmatrix} 2\sum x_i^2 & 2\sum x_i \\ 2\sum x_i & 2n \end{bmatrix}, \quad \widetilde{x} = \begin{bmatrix} m \\ q \end{bmatrix}, \quad c = \begin{bmatrix} 2\sum x_i y_j \\ 2\sum y_i \end{bmatrix}.$$

Second approach

Se procedo con la retta di regressione, l'approccio è diverso? no.

The result is obtained from a random vector

$$(X, Y) : \Omega \to \mathbb{R}^2$$

and we look for a regression between variables of the type $Y = \beta_1 X + \beta_0$.

We known that, estimators for β_1 and β_0 are (if $\text{var}(X) \neq 0$)

$$\beta_1 = \frac{\text{cov}(X, Y)}{\text{var}(X)}, \qquad \beta_0 = \mathbb{E}[Y] - \beta_1 \mathbb{E}[X].$$

Exercise

Is this a different solution?

Third approach

Let L^2 bet the space of functions on $\{x_1, \ldots, x_n\}$ with the measure that "counts the points" (isomorphic to $\mathbb{R}^n$)

$$\int f d\mu := \sum_i f(x_i).$$

We look for the best approximation with respect to $\mathcal{F} = \{1, x\}$.

The solution is of the type $\alpha_0 + \alpha_1 x$, where $\alpha = \begin{bmatrix} \alpha_0 & \alpha_1 \end{bmatrix}^T$ solves the linear system

$$A\alpha = b, \qquad A = \begin{bmatrix} \langle \varphi_0, \varphi_0 \rangle & \langle \varphi_1, \varphi_0 \rangle \\ \langle \varphi_0, \varphi_1 \rangle & \langle \varphi_1, \varphi_1 \rangle \end{bmatrix}, \qquad b = \begin{bmatrix} \langle \varphi_0, \psi \rangle \\ \langle \varphi_1, \psi \rangle \end{bmatrix},$$

where $\varphi_0 = 1$, $\varphi_1 = x$ and $\psi = [y_1, \ldots, y_n]$.

Exercise

Is this a different solution?

Fitting lineare

$c_1 e^x + c_2 e^{2x}$

Combinazione lineare di funzioni che sia combinazione lineare di ...

Linear fitting

Let $\mathcal{F} = \text{span}\{\varphi_0, \ldots, \varphi_\ell\}$, with $\varphi_0, \ldots, \varphi_\ell$ linearly independent real functions.

Let

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \quad \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}, \quad \ldots, \quad \begin{bmatrix} x_n \\ y_n \end{bmatrix},$$

find $f \in \mathcal{F}$ such that

$$\sum_{i=1}^{n} |f(x_i) - y_i|^2$$

is minimum.

(Problema di ottimizzazione)

Linear fitting

Find $f \in \mathcal{F}$ such that

$$\sum_{i=1}^{n} |f(x_i) - y_i|^2$$

is minimum.

Writing $f = \alpha_0\varphi_0 + \cdots + \alpha_\ell\varphi_\ell$, the problem can be stated as

Find the minimum of $\|Ax - b\|$, with $A \in \mathbb{C}^{n \times \ell}$

$$A = \begin{bmatrix} \varphi_0(x_1) & \cdots & \varphi_\ell(x_1) \\ \vdots & \ddots & \vdots \\ \varphi_0(x_n) & \cdots & \varphi_\ell(x_n) \end{bmatrix}, \qquad b = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \qquad x = \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_\ell \end{bmatrix}.$$

$$\|Ax-b\|_2^2 = \sum_i \left| \sum_j \varphi_j(x_i)\alpha_i - y_i \right|^2$$

$\Rightarrow$ Il problema si pone come trovare $\alpha_0, \dots \alpha_k$ t.c

$$\sum_i \left| \sum_j \varphi_j(x_i)\alpha_j - y_i \right|^2$$

30/09

Prendiamo dei dati $\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \dots \begin{bmatrix} x_n \\ y_n \end{bmatrix} \in \mathbb{R}^2$ e

$V_j = \text{span}\{\varphi_0, \dots \varphi_n\} \subset C[a,b]$, φ_i L.I e $x_i \in [a,b]$

Trovare la soluzione di $\underset{f \in V_j}{\text{argmin}} \|\varepsilon\|_2$

$$\varepsilon = \begin{bmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{bmatrix}, \quad \varepsilon_i = f(x_i) - y_i$$

$f = \sum_{j=0}^{n} \alpha_j \varphi_j$ $\Rightarrow$ Il vero problema è trovare

$$\underset{\alpha_0, \dots \alpha_n \in \mathbb{R}}{\text{argmin}} \left\| \sum_{j=0}^{m} \alpha_j \varphi_j(x_i) - y_i \right\|_2^2$$

Se raccolgo i dati noti in una matrice

$$A = \begin{bmatrix} \varphi_0(x_1) & \cdots & \varphi_m(x_1) \\ \vdots & & \vdots \\ \varphi_0(x_n) & \cdots & \varphi_m(x_n) \end{bmatrix} \quad b = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} \quad x = \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_n \end{bmatrix}$$

allora dobbiamo trovare $\underset{x \in \mathbb{R}^{m+1}}{\text{argmin}} \|Ax-b\|^2$

ove $(Ax-b)_i = \sum_{j=0}^{m} \alpha_j \varphi_j(x_i) - y_i = \varepsilon_i$

(Il problema è lo stesso scritto però in forma compatta)

Overdetermined systems

A classical problem is the solution of a linear system $Ax = b$ where $A \in \mathbb{R}^{m \times n}$, with $m \geqslant n$, $b \in \mathbb{R}^m$ and $x \in \mathbb{R}^n$.

If $m > n$ the system is said to be **overdetermined** and has no solution for almost every $b \in \mathbb{R}^m$.

When a system has no solution one might be interested in x such that $\|Ax - b\|$ is minimus (hoping that $Ax \approx b$).

Same problem as before.

The linear least squares problem

Problem

Let $A \in \mathbb{C}^{m \times n}$ and $b \in \mathbb{C}^m$, find $x \in \mathbb{C}^n$ such that $\|Ax - b\|$ is minimum.

For any solution x^*, the quantity $Ax^* - b$ is said to be **residual** and we are interested in its norm.

Exercise

If the linear system $Ax = b$ has a solution, then the set of solutions of the linear least squares problem coincides with the set of solutions of the linear system.

In the general case, we have a complete characterization about the existence and uniqueness of solutions.

The linear least squares problem

Theorem

Let $A \in \mathbb{C}^{m \times n}$ with $m \geqslant n$ and $b \in \mathbb{C}^m$. The linear least squares problem with data A and b has solutions and the set of solutions coincides with the solution of the square linear system $n \times n$

$$A^* A x = A^* b.$$

There exists a unique solution if and only if $\operatorname{rank}(A) = n$.
There exists unique a solution x^* of minimum norm and it belongs to $\ker(A^* A)^\perp$.

- The problem has **always** a solution.
- It can be **reduced** to a square linear system with Hermitian positive semidefinite coefficient.
- **Uniqueness** is obtained by choosing the solution with minimum norm.
- We have both a linear algebra and an analysis proof.

The linear least squares problem

Theorem

Let $A \in \mathbb{C}^{m\times n}$ with $m \geqslant n$ and $b \in \mathbb{C}^m$. The linear least squares problem with data A and b has solutions and the set of solutions coincides with the solution of the square linear system $n \times n$

$$A^* Ax = A^* b.$$

Proof.

Observe that $\text{range}(A)^\perp = \ker A^*$

$$v \in \text{range}(A)^\perp \iff v^* Aw = 0, w \in \mathbb{C}^n$$

$$\iff \langle A^* v, w\rangle = 0, w \in \mathbb{C}^n \iff A^* v = 0 \iff v \in \ker A^*.$$

We know that $\mathbb{C}^m = \text{range}(A) \oplus \text{range}(A)^\perp$, so that there exist unique $b_1 \in \text{range}(A)$ and $b_2 \in \ker A^*$ such that $b = b_1 + b_2$.

The linear least squares problem

Theorem

Let $A \in \mathbb{C}^{m\times n}$ with $m \geqslant n$ and $b \in \mathbb{C}^m$. The linear least squares problem with data A and b has solutions and the set of solutions coincides with the solution of the square linear system $n \times n$

$$A^*Ax = A^*b.$$

Proof.

There exist unique $b_1 \in \text{range}(A)$ and $b_2 \in \ker A^*$ such that $b = b_1 + b_2$

$$\|Ax - b\|^2 = \langle Ax - b_1 - b_2, Ax - b_1 - b_2\rangle$$

$$\langle Ax - b_1, Ax - b_1\rangle + \langle b_2, b_2\rangle = \|Ax - b_1\|^2 + \|b_2\|^2.$$

Since $\langle Ax - b_1, b_2\rangle = 0$, because $Ax - b_1 \in \text{range}(A)$ and $b_2 \in \text{range}(A)^\perp$.

The linear least squares problem

Theorem

Let $A \in \mathbb{C}^{m\times n}$ with $m \geqslant n$ and $b \in \mathbb{C}^m$. The linear least squares problem with data A and b has solutions and the set of solutions coincides with the solution of the square linear system $n \times n$

$$A^*Ax = A^*b.$$

Proof.

There exist unique $b_1 \in \text{range}(A)$ and $b_2 \in \ker A^*$ such that $b = b_1 + b_2$

$$\|Ax - b\|^2 = \|Ax - b_1\|^2 + \|b_2\|^2.$$

The minimum is attained when $\|Ax - b_1\|^2$ is minimum, but since $b_1 \in \text{range}(A)$, the system $Ax = b_1$ has a solution, and the solutions of the least squares problem are solution of the system

$$Ax = b_1.$$

The linear least squares problem

Theorem

Let $A \in \mathbb{C}^{m\times n}$ with $m \geqslant n$ and $b \in \mathbb{C}^m$. The linear least squares problem with data A and b has solutions and the set of solutions coincides with the solution of the square linear system $n \times n$

$$A^*Ax = A^*b.$$

Proof.

There exist unique $b_1 \in \operatorname{range}(A)$ and $b_2 \in \ker A^*$ such that $b = b_1 + b_2$. We prove that $Ax = b_1$ is equivalent to $A^*Ax = A^*b$.

$$A^*Ax = A^*b \overset{A^*(Ax-b)=0}{\iff} Ax - b_1 - b_2 \in \ker A^*$$

$$\overset{b_2\in\ker A^*}{\iff} Ax - b_1 \in \ker A^* \overset{\ker A^*=\operatorname{range}(A)^\perp}{\iff} Ax - b_1 \in \operatorname{range}(A)^\perp$$

$$\overset{Ax-b_1\in\operatorname{range}(A)}{\iff} Ax - b_1 = 0.$$

Since $\operatorname{range}(A) \cap \operatorname{range}(A)^\perp = 0$. □

The linear least squares problem

Theorem

Let $A \in \mathbb{C}^{m\times n}$ with $m \geqslant n$ and $b \in \mathbb{C}^m$. The linear least squares problem with data A and b has solutions and the set of solutions coincides with the solution of the square linear system $n \times n$

$$A^*Ax = A^*b.$$

There exists a unique solution if and only if $\operatorname{rank}(A) = n$.
There exists unique a solution x^* of minimum norm and it belongs to $\ker(A^*A)^\perp$.

Proof.

The linear system $A^*Ax = A^*b$ has unique solution if and only if $\operatorname{rank}(A^*A) = n$, but

$$\operatorname{rank}(A^*A) = \operatorname{rank}(A).$$

because [a]

$$A^*Av = 0 \Leftrightarrow v^*A^*Av = 0 \Leftrightarrow \|Av\|^2 = 0 \Leftrightarrow Av = 0.$$

[a] If $M \in \mathbb{C}^{m\times n}$ with $m \geqslant n$, $\operatorname{rank}(M) < n$ if and only if $Mv = 0$, for $v \neq 0$.

The linear least squares problem

Theorem

Let $A \in \mathbb{C}^{m \times n}$ with $m \geqslant n$ and $b \in \mathbb{C}^m$. The linear least squares problem with data A and b has solutions and the set of solutions coincides with the solution of the square linear system $n \times n$

$$A^* A x = A^* b.$$

There exists a unique solution if and only if $\operatorname{rank}(A) = n$.
There exists unique a solution x^* of minimum norm and it belongs to $\ker(A^* A)^{\perp}$.

Proof.

The solutions of a linear system are an affine subspace, where there exists a point of minimum norm.

For the last statement see the references. □

Esempio (si ottiene un sistema sovradeterminato)

$Ax=b \quad A \in \mathbb{C}^{m \times n}, b \in \mathbb{C}^m$ ← Se $m > n$ è sovradeterminato e spesso quindi la sol $\nexists$ (non è ben posto)

Spesso il problema pratico ha sol. ma il sist sovradet associato non ha sol poiché $rg(A) \neq rg(A|b)$

Allora in questo caso si ragiona così: si cerca un vettore il più vicino possibile a una sol. intuitiva data dalle informazioni che ho

Cerco la x per cui $Ax \simeq b$

$\Rightarrow$ il problema diventa

$$\arg\min_{x \in \mathbb{C}^n} \|Ax-b\|$$

Def

Se x^* è una soluzione allora chiamiamo Ax^*-b residuo

Esercizio:

Se il sistema lineare $Ax=b$ ha soluzione, allora l'insieme W delle soluzioni nel problema dei minimi quadrati (LS), con A, b come sotto, allora W coincide con l'insieme V delle sol del sistema lineare

Se $x^* \in V$, $Ax^*-b=0$ per ogni $x \in \mathbb{C}^n$

$\|Ax-b\| \geq 0 = \|Ax^*-b\| \Rightarrow x^* \in W$

Se $\hat{x} \in W$ e $x^* \in V \Rightarrow \|A\hat{x} - b\| \leq \|Ax^* - b\| = 0$

$\Rightarrow \|A\hat{x} - b\| = 0 \Leftrightarrow A\hat{x} = b \Rightarrow \hat{x} \in V$

Teorema del problema dei minimi quadrati:

Sia $A \in \mathbb{C}^{m \times n}$, $m \geq n$ e $b \in \mathbb{C}^m$. Il problema dei minimi quadrati con dati A e b ha soluzione

+ slide

oss:
- Prob ha sempre sol (non sempre unica)
- Il problema si riduce a un sistema lineare

Richiami di analisi:

$f: \Omega \to \mathbb{R}$, $\Omega \subseteq B$, B sp. di Banach con norma $\|\cdot\|$

f è differenziabile in $x \in \Omega$ se $\exists$ un'applicazione lineare continua $Df(x): B \to \mathbb{R}$ t.c

$$\lim_{h \to 0} \frac{|f(x+h) - f(x) - Df(x)[h]|}{\|h\|} = 0$$

Df si chiama derivata di Frechet di f in x o anche

$$f(x+h) = f(x) + Df(x)[h] + o(\|h\|)$$

$$f(x+h) = f(x) + Df(x)[h] + \frac{1}{2} D^2 f(x)[h,h] + o(\|h\|^2)$$

$D^2 f(x): B \times B \to \mathbb{R}$

derivata seconda ↗ ↖ bilineare

$\Omega \subseteq \mathbb{R}^n \quad \Omega \to \mathbb{R} \quad Df(x): \mathbb{R}^n \to \mathbb{R} \quad A \in \mathbb{R}^{1\times n}$

$$A = \left[\frac{\partial f}{\partial x_1}(x), \dots, \frac{\partial f}{\partial x_n}(x) \right]$$

$$Df(x)[h] = \langle \nabla f(x), h \rangle, \quad h \in \mathbb{R}^n, \quad \nabla f(x) \in \mathbb{R}^n \cong \mathbb{R}^{n\times 1}$$

($\nabla f(x)$: gradiente)

$$D^2 f(x)[h,h] = h^* H h$$

(H: hessiana)

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & - \\ - & - \end{bmatrix}$$

Dimostrazione ① del Teorema dei minimi quadrati:

$\langle Av, w \rangle = \langle v, A^* w \rangle$

Voglio trovare i minimi di $\|Ax-b\|^2 = \varphi(x)$

$$\varphi(x+h) - \varphi(x) = \langle A(x+h)-b, A(x+h)-b \rangle - \langle Ax-b, Ax-b \rangle =$$

$$= \cancel{\langle Ax-b, Ax-b \rangle} + \langle Ah, Ah \rangle + 2\langle Ax-b, Ah \rangle - \cancel{\langle Ax-b, Ax-b \rangle} =$$

$$= \underbrace{\langle \overbrace{2A^*(Ax-b)}^{\nabla\varphi}, h \rangle}_{\text{lineare}} + \underbrace{h^* A^* A h}_{\text{bilineare } (\circ(\|h\|)) \text{ perché}}$$

$\|h^* A^* A h\| \le \|h^*\| \|A^* A\| \|h\| < \|h\|$
$= \circ(\|h\|)$

$$\nabla\varphi(x) = 2(A^*Ax - A^*b) = 0 \iff A^*Ax = A^*b$$

$D^2\varphi(x) = A^*A \ge 0$ (condizione che le sole siano di minimo loc)

Dimostrazione ②:

$\mathrm{Im}A^{\perp} = \ker A^*$ (dimostriamo questo)

$v \in \mathrm{Im}A^{\perp} \iff \langle v, Aw \rangle = 0 \quad w \in \mathbb{C}^n$ (Aw)

$\iff \langle A^*v, w \rangle = 0 \quad w \in \mathbb{C}^n \iff A^*v = 0 \iff v \in \ker A^*$

ma $\mathbb{C}^m = \mathrm{Im}A \oplus \mathrm{Im}A^{\perp}$

$\Rightarrow b = b_1 + b_2$ con $b_1 \in \mathrm{Im}A$, $b_2 \in \mathrm{Im}A^{\perp} = \ker A^*$

$\Rightarrow \|Ax-b\|^2 = \|Ax - b_1 - b_2\|^2 = \|Ax-b_1\|^2 + \|b_2\|^2 - 2\mathrm{Re}\langle Ax-b_1, b_2 \rangle =$ ($= 0 \ \forall x$; $Ax-b_1 \in \mathrm{Im}A$, $b_2 \in \mathrm{Im}A^{\perp}$)

$= \|Ax-b_1\|^2 + \|b_2\|^2$ ($\|b_2\|^2$ costante; $\arg\min f = \arg\min f + c$)

$\Rightarrow \arg\min \|Ax-b\|^2 = \arg\min \|Ax-b_1\|^2$

ma $Ax = b_1$ ha soluzione perché $b_1 \in \mathrm{Im}A$

$\Rightarrow \arg\min \|Ax-b\|^2 = \{x \mid Ax = b_1\}$

Dimostriamo che $Ax = b_1 \iff A^*Ax = A^*b$

$A^*Ax = A^*b \iff A^*(Ax-b) = 0 \iff Ax-b \in \ker A^*$

$\iff Ax - b_1 - b_2 \in \mathrm{Im}A^{\perp}$ ($b_2 \in \mathrm{Im}A^{\perp}$) $\iff Ax - b_1 \in \mathrm{Im}A^{\perp}$ ($Ax-b_1 \in \mathrm{Im}A$)

ma $\mathrm{Im}A^{\perp} \cap \mathrm{Im}A = \{0\} \iff Ax - b_1 = 0 \iff Ax = b_1$

Dimostriamo la parte della minimalità se sistema ha soluzione unica se e solo se A^*A è invertibile

A^*A è semidefinita positiva:

Se $\text{rg}\, A < n \Rightarrow Ax = 0$ ha sol non nulla $\Leftrightarrow \|Ax\|^2 = 0$
$\Leftrightarrow x^* A^* A x = 0$
Se $A^*Av = 0 \Rightarrow v^* A^* A v = 0 \Rightarrow \|Av\|^2 = 0 \Rightarrow Av = 0$
(A^*A non invert)

$\Rightarrow$ Se $\text{rg}\, A = n$ allora A^*A è invertibile

Vediamo tre algoritmi per risolvere il sistema:
1 - Fattorizzazione di Cholesky ($\text{rg}\, A = n$)
2 - Fattorizzazione QR
3 - SVD

The Choleksy approach

Theorem

Let $A \in \mathbb{C}^{n\times n}$ be positive definite, there exists unique $R \in \mathbb{C}^{n\times n}$ upper triangular such that $A = R^*R$ and the diagonal elements of R are positive.

Proof.

The principal minor of a positive definite matrix are positive definite $\Rightarrow$ A admits a unique LU factorization.

Write $U = D\widetilde{U}$, where D is diagonal a $\widetilde{U}$ is upper triangular with 1 on the diagonal

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ddots & 0 & u_{nn} \end{bmatrix} = \begin{bmatrix} u_{11} & 0 & \cdots & 0 \\ 0 & u_{22} & \cdots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ddots & 0 & u_{nn} \end{bmatrix} \begin{bmatrix} 1 & u_{12}/u_{11} & \cdots & u_{1n}/u_{11} \\ 0 & 1 & \cdots & u_{2n}/u_{22} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ddots & 0 & 1 \end{bmatrix}$$

The Choleksy approach

Theorem

Let $A \in \mathbb{C}^{n\times n}$ be positive definite, there exists unique $R \in \mathbb{C}^{n\times n}$ upper triangular such that $A = R^*R$ and the diagonal elements of R are positive.

Proof.

Since

$$A = LU = L(D\widetilde{U}), \qquad A = A^* = \widetilde{U}^* D^* L^*$$

$L(D\widetilde{U})$ and $\widetilde{U}^*(D^*L^*)$ are LU factorizations of A, by uniqueness of LU factorization $L = \widetilde{U}^*$ and we can write $A = LDL^*$.

□

The Choleksy approach

Theorem

Let $A \in \mathbb{C}^{n\times n}$ be positive definite, there exists unique $R \in \mathbb{C}^{n\times n}$ upper triangular such that $A = R^*R$ and the diagonal elements of R are positive.

Proof.

We have $A = LDL^*$, we prove that D has positive diagonal elements. If $v \neq 0$, then

$$v^*Dv = v^*L^{-1}AL^{-*}v = w^*Aw > 0,$$

with $w = L^{-*}v \neq 0$ since $v \neq 0$. This implies that D is definite positive and thus has a positive diagonal.

□

The Choleksy approach

Theorem

Let $A \in \mathbb{C}^{n\times n}$ be positive definite, there exists unique $R \in \mathbb{C}^{n\times n}$ upper triangular such that $A = R^*R$ and the diagonal elements of R are positive.

Proof.

$A = LDL^*$, D has positive diagonal, we can write

$$D = \begin{bmatrix} u_{11} & 0 & \cdots & 0 \\ 0 & u_{22} & \cdots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ddots & 0 & u_{nn} \end{bmatrix} = \begin{bmatrix} \sqrt{u_{11}} & 0 & \cdots & 0 \\ 0 & \sqrt{u_{22}} & \cdots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ddots & 0 & \sqrt{u_{nn}} \end{bmatrix} \begin{bmatrix} \sqrt{u_{11}} & 0 & \cdots & 0 \\ 0 & \sqrt{u_{22}} & \cdots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ddots & 0 & \sqrt{u_{nn}} \end{bmatrix} = FF^*$$

and thus $A = (LF)(F^*L)$. Setting $R = F^*L$ we have the proof since $A = R^*R$, R is upper triangular and R has positive diagonal elements.

□

The Cholesky approach

Features of the Cholesky factorization.

- Characterizes positive definite matrices;
- Parametrize (complex) positive definite matrices as $\mathbb{R}^{n(n-1)} \times \mathbb{R}_+^n$;
- Can be computed by an algorithm requiring $n^3/3 + o(n^3)$ arithmetic operations.

If A has rank n, the Cholesky factorization of A^*A allows one to solve the least squares problem

$$A^*Ab = A^*b,$$

Drawback: A^*A must be computed and can have a large condition number (larger than the one of A).

1 - Fattorizzazione di Cholesky:

Teorema

Sia $A \in \mathbb{C}^{n \times n}$ def positiva, $\exists ! R \in \mathbb{C}^{n \times n}$ triangolare superiore con elementi positivi sulla diagonale t.c $A = R^* R$ (fattorizzazione di Cholesky)

Dimo:

Per ind. su n:

$n=1$: $A = a > 0$ $\quad A = \sqrt{a}\sqrt{a}$ in modo unico

$R = R^* = \sqrt{a}$

$n-1 \Rightarrow n$: $A \in \mathbb{C}^{n \times n}$ $\quad A = \left[\begin{array}{c|c} A_{n-1} & u \\ \hline u^* & \alpha \end{array}\right]$

Mostriamo che A_{n-1} è def positiva perché A è def positiva

e anche α è >0 perché A è def positiva ($\mu\nu\omega > 0$)

Inoltre A invert e anche A_{n-1}

$$\begin{bmatrix} I & 0 \\ -u^* A_{n-1}^{-1} & 1 \end{bmatrix} \begin{bmatrix} A_{n-1} & u \\ u^* & \alpha \end{bmatrix} =$$

($\det = 1$, $\det > 0$)

$$= \begin{bmatrix} A_{n-1} & u \\ -u^* A_{n-1}^{-1} A_{n-1} + u^* & \alpha - u^* A_{n-1}^{-1} u \end{bmatrix}$$

$\alpha - u^* A_{n-1}^{-1} u = \gamma$

$$\det = \det A_{n-1} \cdot \gamma > 0 \Rightarrow \gamma > 0$$

($\det A_{n-1} > 0$)

$$A = R^* R \qquad R = \begin{bmatrix} R_{n-1} & y \\ 0 & x \end{bmatrix}$$

$$\Rightarrow \left[\begin{array}{c|c} A_{n-1} & u \\ \hline u^* & \alpha \end{array}\right] = \begin{bmatrix} R_{n-1}^* & 0 \\ y^* & \bar{x} \end{bmatrix} \begin{bmatrix} R_{n-1} & y \\ 0 & x \end{bmatrix} =$$

$$= \begin{bmatrix} R_{n-1}^* R_{n-1} & R_{n-1}^* y \\ y^* R_{n-1} & y^* y + \bar{x} x \end{bmatrix}$$

$$\begin{cases} A_{n-1} = R_{n-1}^* R_{n-1} & \text{per Hp induttiva ho sol unica} \\ u = R_{n-1}^* y & \leftarrow R_{n-1} \text{ ho el positivi su diag} \Rightarrow \det R_{n-1} > 0 \\ & \Rightarrow \det R_{n-1}^* > 0 \Rightarrow \text{ES ISTANZA ho sol unica } y \\ \alpha = y^* y + \bar{x} x, \ x > 0 \Rightarrow & |x|^2 = \alpha - y y^* \end{cases}$$

una $y = R_{n-1}^{*-1} u \Rightarrow y^* = u^* \left(R_{n-1}^{*-1}\right)^* = u^* R_{n-1}^{-1}$

$\Rightarrow |x|^2 = \alpha - u^* R_{n-1}^{-1} R_{n-1}^{*-1} u = \alpha - u^* A_{n-1}^{-1} u = \gamma$

$\Rightarrow x$ è unica sol possibile $x = \sqrt{\gamma}$

OSS:

$\rightarrow A_{n-1} = R_{n-1}^* R_{n-1} \quad A_{n-1}^{-1} = (R_{n-1}^* R_{n-1})^{-1} = R_{n-1}^{-1} (R_{n-1}^*)^{-1}$

$\rightarrow u = R_{n-1}^* y \Rightarrow y = (R_{n-1}^*)^{-1} u$

$\rightarrow u^* = (R_{n-1}^* y)^* = y^* R_{n-1} \Rightarrow y^* = u^* R_{n-1}^{-1}$

Algoritmo che calcola fattorizzazione di Cholesky

$A = R^* R$ (vediamo in $\mathbb{C}^{3\times 3}$)

$$\begin{bmatrix} r_{11} & 0 & 0 \\ \bar{r}_{12} & r_{22} & 0 \\ \bar{r}_{13} & \bar{r}_{23} & r_{33} \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ 0 & r_{22} & r_{23} \\ 0 & 0 & r_{33} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ \bar{a}_{12} & a_{22} & a_{23} \\ \bar{a}_{13} & \bar{a}_{23} & a_{33} \end{bmatrix}$$

$$\begin{cases} r_{11}^2 = a_{11} \\ r_{11} r_{12} = a_{12} \\ r_{11} r_{13} = a_{13} \\ \bar{r}_{12} r_{12} + r_{22}^2 = a_{22} \\ \bar{r}_{12} r_{13} + r_{22} r_{23} = a_{23} \\ \bar{r}_{13} r_{13} + \bar{r}_{23} r_{23} + r_{33}^2 = a_{33} \end{cases}$$

$$\begin{cases} r_{11} = \sqrt{a_{11}} \\ r_{12} = a_{12} / r_{11} \\ r_{13} = a_{13} / r_{11} \\ r_{22} = \sqrt{a_{22} - \bar{r}_{12} r_{12}} \\ r_{23} = \dots \\ r_{33} = \dots \end{cases}$$

Vediamolo simbolicamente

$$(R^* R)_{ij} = a_{ij}, \quad i \le j$$

$$(R^* R)_{ij} = \sum_{l=1}^{n} (R^*)_{il} (R)_{lj} = \sum_{l=1}^{n} \bar{r}_{li} r_{lj} = \quad \hookrightarrow 0 \text{ se } l > j \quad \hookleftarrow 0 \text{ se } l > i$$

$$= \sum_{l=1}^{\min\{i,j\}} \bar{r}_{li} r_{lj} = \sum_{l=1}^{i} \bar{r}_{li} r_{lj} = a_{ij} \qquad i = 1, \dots, n \quad i \le j$$

$\Rightarrow$?

$$\sum_{l<i} \bar{r}_{li} r_{li} + \bar{r}_{ii} r_{ii} = a_{ii} \qquad i = j$$

$$\Rightarrow r_{11} = \sqrt{a_{ii} - \sum_{l<i} |r_{li}|^2}$$

Per $i < j$ $\quad \sum_{l<i} \bar{r}_{li} r_{lj} + r_{ii} r_{ij} = a_{ij}$

$$\Rightarrow r_{ij} = \frac{a_{ij} - \sum_{l<i} \bar{r}_{li} r_{lj}}{r_{ii}}$$

(devo calcolare in ordine crescente per righe da sx a dx)

L'implementazione segue la forma

per $i = 1, \dots, n$ si $r_{ii} = \dots$

per $j = i+1, \dots, n$ si $r_{ij} = \dots$

Ha senso se richiedo di r_{ii} e' positivo
e se $r_{ii} \neq 0$

Costo computazionale

Per r_{ii}: $2(i-1)$ ops + 1 sqrt → $\gamma = O(1)$ n di ops di sqrt
Per r_{ij}: $2(i-1)+1$ ops
↑ divisione

$$\Rightarrow \sum_{i=1}^{n} 2(i-1) + \gamma + \sum_{i=1}^{n} \sum_{j=i+1}^{n} 2(i-1)+1$$

per $n \to \infty$ $\quad O(n^2) + \sum_{i=1}^{n} (n-i)(2(i-1)+1) \simeq$

$$\simeq \sum_{i=1}^{n} 2i(n-i) = 2\left(n \sum i - \sum i^2\right) \sim 2\left(n\frac{n^2}{2} - \frac{n^3}{3}\right) \sim \frac{n^3}{3} \text{ ops}$$

(La metà del tempo decisamente Gauss rispetto a metà precedenti)

Dunque: 01/10
- costruire A^*A e A^*b $\longrightarrow 2mn^2$
- Fattorizzare A^*A come R^*R $\longrightarrow \frac{n^3}{3}$
- $R^*y = A^*b \longrightarrow n^2$
- $Rx = y \longrightarrow n^2$

Esempio: $A = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$ $b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ $Ax = b$

$\begin{cases} 2x = 1 \\ x = 1 \end{cases}$ non ha sol chiaramente

Ma vediamo il metodo per $\min \|Ax-b\|^2$

The QR factorization approach

If $A = QR = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$, with $R_1 \in \mathbb{C}^{n \times n}$

$$\|Ax - b\|^2 = \|QRx - b\|^2 = \|Q(Rx - Q^* b)\|^2 \overset{\|Qv\| = \|v\|}{=} \|Rx - Q^* b\|^2$$

$$= \left\| \begin{bmatrix} R_1 \\ 0 \end{bmatrix} x - \begin{bmatrix} Q_1^* \\ Q_2^* \end{bmatrix} b \right\|^2 = \left\| \begin{bmatrix} R_1 x - Q_1^* b \\ Q_2^* b \end{bmatrix} \right\|^2$$

$$\overset{\|\left[\begin{smallmatrix} v_1 \\ v_2 \end{smallmatrix}\right]\|^2 = \|v_1\|^2 + \|v_2\|^2}{=} \|R_1 x - Q_1^* b\|^2 + \|Q_2^* b\|^2.$$

The minimum of $\|Ax - b\|^2$ is reached for x such that $R_1 x = Q_1^* b$ and the norm of the residual is $\|Q_2^* b\|$.

Can deal also with the case $\text{rank}(A) < n$ (QR factorization with column pivoting).

QR factorization

A **QR factorization** of a matrix $A \in \mathbb{C}^{n \times n}$ consists in writing the matrix A as the product QR where $Q \in U(n)$ is unitary and $R \in \mathbb{C}^{n \times n}$ upper triangular.

When $A \in \mathbb{R}^{n \times n}$, we may require that Q is orthogonal and R is real.

Can be extended to rectangular matrices, where R is upper triangular with the same size as A.

Existence of the QR factorization

Theorem

Any square matrix admits a QR factorization.

Proof.

Idea: use the fact that any unitary vector $v \in \mathbb{C}^n$, can be completed to a basis of $\mathbb{C}^n$. □

Existence of the QR factorization

Theorem

Any square matrix admits a QR factorization.

Proof.

We proceed by induction on the size n of the matrix A. The case $n = 1$ is given by $A = QR$ with $Q = 1$ and $R = A$. □

Existence of the QR factorization

Theorem

Any square matrix admits a QR factorization.

Proof.

Assume that $n > 1$ and let $v \in \mathbb{C}^n$ be the first column of A. If $v = 0$ set $q_1 = [1\ 0\ \cdots\ 0]^T$, elsewhere set $q_1 = v/\|v\|$.

There exist $q_2, \ldots, q_n$ spanning an $n-1$ dimensional subspace of $\mathbb{C}^n$ orthogonal to span(q_1).

Let $U = [q_1|Q_2]$, where $Q_2 = [q_2|\cdots|q_n]$ and partition $A = [v|B]$ with $B \in \mathbb{C}^{n\times(n-1)}$. Notice that U is unitary and

$$U^*A = \begin{bmatrix} q_1* \\ Q_2^* \end{bmatrix} \begin{bmatrix} v & B \end{bmatrix} = \begin{bmatrix} \|v\| & q_1^*B \\ 0 & Q_2^*B \end{bmatrix}.$$

□

Existence of the QR factorization

Theorem

Any square matrix admits a QR factorization.

Proof.

By induction we can write $Q_2^* B = \widetilde{Q}\widetilde{R}$, with $\widetilde{Q} \in U(n-1)$ and $\widetilde{R}$ upper triangular. Setting $\widehat{Q} = \begin{bmatrix} 1 & 0 \\ 0 & \widetilde{Q}^* \end{bmatrix} \in U(n)$ we get that

$$R = \widehat{Q}(U^* A) = \begin{bmatrix} 1 & 0 \\ 0 & \widetilde{Q}^* \end{bmatrix} \begin{bmatrix} \|v\| & q_1^* B \\ 0 & Q_2^* B \end{bmatrix} = \begin{bmatrix} \|v\| & q_1^* B \\ 0 & \widetilde{R} \end{bmatrix}$$

is upper triangular, thus $A = QR$ with $Q = U\widehat{Q}^*$ is a QR factorization of A. □

Uniqueness of the QR factorization

The QR factorization is **not unique**, think to the one-dimensional case $a = e^{i\theta}(e^{-i\theta}a)$.

It is easy to characterize all QR factorizations of a nonsingular matrix in terms of phase matrices.

A **phase matrix** $F \in \mathbb{C}^{n\times n}$ is a diagonal unitary matrix

$$F = \begin{bmatrix} e^{i\theta} & & & \\ & e^{i\theta_2} & & \\ & & \ddots & \\ & & & e^{i\theta_n} \end{bmatrix}.$$

Exercise

Prove that a unitary and upper triangular matrix is a phase matrix.

Uniqueness of the QR factorization

Theorem

Let $A \in GL(n)$ and let QR be a factorization of A.
For any QR factorization of A, say $A = \widetilde{Q}\widetilde{R}$ there exists a phase matrix F such that $\widetilde{Q} = QF^*$ and $\widetilde{R} = FR$. Conversely, any phase matrix F yields a QR factorization $A = (QF^*)(FR)$.
In particular, there exists unique a QR factorization of A where R has positive diagonal entries.

Proof.

Let $A = Q_1R_1 = Q_2R_2$ be two QR factorizations, with A invertible.

Every matrix is invertible, and $Q_1R_1 = Q_2R_2$ implies that $F := Q_2^*Q_1 = R_2R_1^{-1}$, the matrix F is unitary and upper triangular and thus it is a phase matrix.

□

Uniqueness of the QR factorization

Theorem

Let $A \in GL(n)$ and let QR be a factorization of A.
For any QR factorization of A, say $A = \widetilde{Q}\widetilde{R}$ there exists a phase matrix F such that $\widetilde{Q} = QF^*$ and $\widetilde{R} = FR$. Conversely, any phase matrix F yields a QR factorization $A = (QF^*)(FR)$.
In particular, there exists unique a QR factorization of A where R has positive diagonal entries.

Proof.

Conversely, if F is a phase matrix, then QF^* is unitary and FR is upper triangular, and since $F^*F = I$, we have that $(QF^*)(FR)$ is a QR factorization of A.

□

Uniqueness of the QR factorization

Theorem

Let $A \in GL(n)$ and let QR be a factorization of A.
For any QR factorization of A, say $A = \widetilde{Q}\widetilde{R}$ there exists a phase matrix F such that $\widetilde{Q} = QF^*$ and $\widetilde{R} = FR$. Conversely, any phase matrix F yields a QR factorization $A = (QF^*)(FR)$.
In particular, there exists unique a QR factorization of A where R has positive diagonal entries.

Proof.

Consider the phase matrix $F = \operatorname{diag}(\overline{r}_{11}/|r_{11}|, \ldots, \overline{r}_{nn}/|r_{nn}|)$. We have that $\widetilde{R} := FR$ is upper triangular with positive diagonal entries and thus $\widetilde{Q}\widetilde{R}$, with $\widetilde{Q} = QF^*$, is the required QR factorization.[a]

The uniqueness follows from the fact that any two QR factorization differs by a phase matrix and if not the identity, it would yield a triangular factor with some nonpositive diagonal entry. □

[a] Since $\det(A) = \det(Q)\det(R)$, we have that $\det(R) \neq 0$ and $r_{ii} \neq 0$.

Uniqueness of the QR factorization

In the real case, a similar result holds with Q orthogonal and R real.

When A is singular, there is no simple result on the uniqueness is not known. Think to the case $A = 0$, where every unitary matrix Q yields the QR factorization $A = Q0$.

How to compute a QR factorization?

Recall the Gaussian elimination.

1. Apply plane rotations (Given algorithm).
2. Apply reflections wrt hyperplanes (Householder algorithm).
3. Orthogonalize (Gram-Schmidt algorithm).

Complex Givens matrices

A complex Givens matrix can be written as

$$G = \begin{bmatrix} c & -\overline{s} \\ s & c \end{bmatrix},$$

where $c = \cos\varphi$, $s = \psi \sin\varphi$ (with $|\psi| = 1$).

If $a, b \neq 0$, then with

$$\psi = -\frac{b}{a}\left|\frac{a}{b}\right|, \qquad \cos\varphi = \frac{|a|}{\sqrt{|a|^2 + |b|^2}}, \qquad \sin\varphi = \frac{|b|}{\sqrt{|a|^2 + |b|^2}},$$

we have that $G \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}$. (The cases $a = 0$ or $b = 0$ can be treated directly.)

Givens method

For any vector $w \in \mathbb{C}^2$, there exists a 2×2 Givens G such that Gw is a multiple of $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

Let $v \in \mathbb{C}^n$, there exists a Givens matrix $G_{n-1,n} \in \mathbb{C}^{n \times n}$ acting on the plane (x_{n-1}, x_n) such that

$$v^{(2)} := G_{n-1,n} v = \begin{bmatrix} v_1^{(2)} \\ v_2^{(2)} \\ \vdots \\ v_{n-1}^{(2)} \\ 0 \end{bmatrix}.$$

Notice that $v_1^{(2)} = v_1, \ldots, v_{n-2}^{(2)} = v_{n-2}$, that is, just the last two elements of v have been changed, by premultiplying it by $G_{n-1,n}$.

Givens method

If $n > 2$, we continue by constructing a matrix $G_{n-2,n-1}$ acting on the plane x_{n-2}, x_{n-1}, such that

$$v^{(3)} := G_{n-2,n-1} v^{(2)} = G_{n-2,n-1} G_{n-1,n} v = \begin{bmatrix} v_1^{(3)} \\ v_2^{(3)} \\ \vdots \\ v_{n-2}^{(3)} \\ 0 \\ 0 \end{bmatrix}.$$

Givens method

And so on, we can construct $G_{n-3,n-2}, \ldots, G_{23}, G_{12}$ such that

$$G_{12} G_{23} \cdots G_{n-2,n-1} G_{n-1,n} v = \begin{bmatrix} \gamma_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Where $|\gamma_1| = \|v\|$, since rotations are isometries.

In the real case, the Givens matrices are plane rotations and we can choose the last rotation G_{12} so that $\gamma_1 = \|v\|$.

Givens method

If v is the first column of a square matrix A, then we have

$$G_{12}G_{23}\cdots G_{n-2,n-1}G_{n-1,n}A = \begin{bmatrix} \gamma_1 & * \\ 0 & B_2 \end{bmatrix} := A_2.$$

We can produce a QR factorization of A by recursively repeating the same procedure for B_2, observing that there exist Givens matrices $\widetilde{G}_{12}, \ldots, \widetilde{G}_{n-2,n-1} \in U(n-1)$ such that

$$\widetilde{G}_{12}\widetilde{G}_{23}\cdots \widetilde{G}_{n-2,n-1}B_2 = \begin{bmatrix} \gamma_2 & * \\ 0 & B_3 \end{bmatrix},$$

with $B_3 \in \mathbb{C}^{(n-2)\times(n-2)}$.

Givens method

From a Givens matrix $G \in U(n-1)$ acting on the plane x_i, x_j, we can construct a Givens matrix $\widehat{G} \in U(n)$

$$\widehat{G} = \begin{bmatrix} 1 & 0 \\ 0 & G \end{bmatrix},$$

acting on the coordinates x_{i+1}, x_{j+1}.

Thus, by defining

$$G^{(2)}_{i+1,i+2} = \begin{bmatrix} 1 & 0 \\ 0 & \widetilde{G}_{i,i+1} \end{bmatrix}, \qquad i = 1, \ldots, n-2$$

we obtain

$$G^{(2)}_{23} G^{(2)}_{34} \cdots G^{(2)}_{n-1,n-2} A_2 = \begin{bmatrix} \gamma_1 & * \\ 0 & \widetilde{G}_{12}\widetilde{G}_{23}\cdots\widetilde{G}_{n-2,n-1}B_2 \end{bmatrix}$$

$$= \begin{bmatrix} \gamma_1 & * & * \\ 0 & \gamma_2 & * \\ 0 & 0 & B_3 \end{bmatrix} := A_3.$$

Givens method

The procedure can be repeated for $n-1$ steps, getting a **upper triangular** matrix $R := A_n$, such that

$$\underbrace{G_{n-1,n}^{(n-1)} G_{n-2,n-1}^{(n-2)} G_{n-1,n}^{(n-2)} \cdots G_{23}^{(2)} \cdots G_{n-1,n-2}^{(2)} G_{12}^{(1)} \cdots G_{n-1,n}^{(1)}}_{Q^*} A = R,$$

where we have denoted G_{ij} by $G_{ij}^{(1)}$.

The matrix Q^* is such that $Q^*A = R$, that is $A = QR$ and we **have found a factorization** of A using $\frac{1}{2}(n-1)n$ Givens matrices.

Example

Applying real Givens matrices to an orthogonal matrix $A \in \mathcal{O}(n)$, and choosing the last rotation at each step so that the diagonal element is positive, at the end we get a phase matrix of the type

$$R = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & 1 & 0 \\ 0 & \cdots & \cdots & 0 & \pm 1 \end{bmatrix},$$

where the bottom left element is 1 if $A \in \mathcal{SO}(n)$. This means that any n dimensional notation can be written as the product of $\frac{1}{2}(n-1)n$ plane rotations. The corresponding angles, provide a **local parametrization** of the differentiable manifold $\mathcal{SO}(n)$.

$U(n)$ requires n^2 parameters.

Computational cost

$$\underbrace{G_{n-1,n}^{(n-1)} G_{n-2,n-1}^{(n-2)} G_{n-1,n}^{(n-2)} \cdots G_{23}^{(2)} \cdots G_{n-1,n-2}^{(2)} G_{12}^{(1)} \cdots G_{n-1,n}^{(1)}}_{Q^*} A = R,$$

$\frac{1}{2} n(n-1)$ matrix multiplications, but each multiplication modifies **just two rows**.

At the first step $n-1$ Givens rotations acting on two rows require $6(n-1)^2$ operations (excluding the ones needed to compute c and s)

At step k we have $6(n-k)^2$ operations, for a total cost of

$$\sum_{k=1}^{n-1} 6(n-k)^2 = 2n^3 + o(n^3)$$

operations.

Implementation

Apparently complicated but...

not really

```
for k = 1:n-1
  for h = n:-1:k+1
    A(h-1:h,:)  = givens(A(h-1,k),A(h,k))*A(h-1:h,:);
  end
end
```

Only one instruction.

Gram-Schmidt algorithm

Problem: find an orthogonal basis of a subspace $\mathcal{W} \subset \mathbb{C}^n$.

Let $v_1, \ldots, v_k \in \mathbb{C}^n$. An orthonormal basis of $\mathcal{W} = \text{span}\{v_1, \ldots, v_k\}$ can be obtained by the following algorithm, for $j = 1 : k$,

- Obtain u_j by removing from v_j the projection onto the span of the previous vectors $w_1, \ldots, w_{j-1}$;
- Normalize u_j to obtain w_j.

The set $w_1, \ldots, w_k$ is an orhtogonal basis of $\mathcal{W}$.

$$u_j = v_j - P_j v_j = v_j - \sum_{\ell=1}^{j-1} \frac{\langle w_\ell, v_j \rangle}{\langle w_\ell, w_\ell \rangle} w_\ell, \qquad w_j = \frac{u_j}{\|u_j\|}.$$

QR factorization and the Gram-Schmidt algorithm

$$u_j = v_j - P_j v_j = v_j - \sum_{\ell=1}^{j-1} \frac{\langle w_\ell, v_j \rangle}{\langle w_\ell, w_\ell \rangle} w_\ell, \qquad w_j = \frac{u_j}{\|u_j\|}.$$

This simple algorithm can be seen as a matrix factorization. Indeed,

$$v_j = \sum_{\ell=1}^{j-1} (w_\ell^* v_j) w_\ell + \|u_j\| w_j = \sum_{\ell=1}^{j} r_{\ell j} w_\ell, \qquad j = 1, \ldots, k,$$

that, can be written in matrix form as

$$A = QR,$$

where $A = [v_1 | \cdots | v_k]$, $Q = [w_1 | \cdots | w_k]$ and

$$R = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1k} \\ 0 & r_{22} & \ddots & r_{2k} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & r_{kk} \end{bmatrix}.$$

QR factorization and the Gram-Schmidt algorithm

When $k = n$, we get a QR factorization of the matrix A.

The Gram-Schmidt algorithm can be extended to the case where the vectors $v_1, \ldots, v_k \in \mathbb{C}^n$ are not necessarily independent (but one of them is nonzero).

The difference in the step is that the vector u can be zero, and in this case, the step of the algorithm is just skipped. We will get a number of orthogonal vectors equal to the dimension of the $\text{span}\{v_1, \ldots, v_k\}$.

Exercise

Explain how to get a QR factorization of a singular matrix using the Gram-Schmidt algorithm.

QR factorization and the Gram-Schmidt algorithm

In a finite arithmetic computation, the Gram-Schmidt algorithm is unstable, and a modified Gram-Schmidt algorithm is preferable

```
for i = 1 :  k
  u = v(:, i);
  for j = 1 :  i - 1 % this is empty if i = 1
    u = u - (u' * w(:,j)) * w(:, j);
  end
  w(:, i) = u / norm(u);
end
```

where the difference is that we project the current value of u on w_j and not v_j. The two algorithms are mathematically equivalent, but in finite arithmetic only the modified one is numerically stable.

Exercise

Determine the computational cost of the Gram-Schmidt method.

The Householder method

Householder matrices can be used to annihilate elements on a vector and this can be used to construct a QR factorization of a matrix.

Let $v \in \mathbb{R}^n$. There exist reflections P, with respect to a vector hyperplane, such that $Pv = \alpha e_1$. The vector hyperplane can be chosen orthogonal to $w = v + \|v\|e_1$ (or $w = v - \|v\|e_1$).

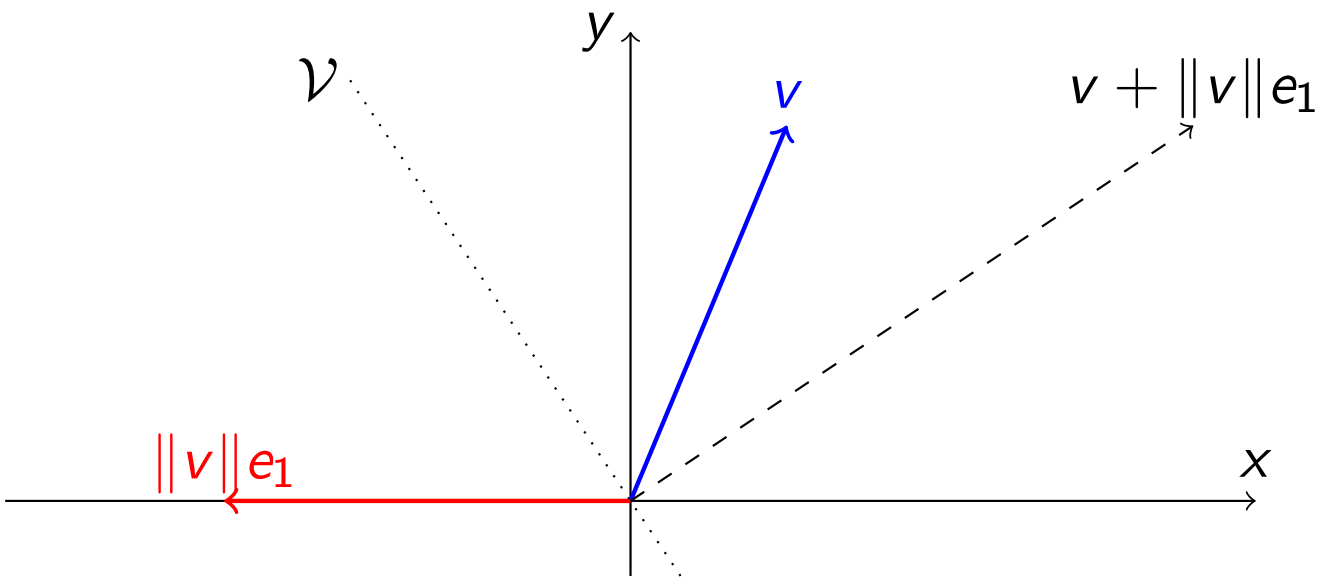


Figure: Moving the vector $v \in \mathbb{R}^2$ to $\|v\|e_1$ using a reflection with respect to a line $\mathcal{V}$ orthogonal to $v + \|v\|e_1$.

Householder method

In the complex case we choose the vector w such that $w + \theta\|v\|e_1$, where

$$\theta = \begin{cases} v_1/|v_1|, & v_1 \neq 0, \\ 1, & v_1 = 0. \end{cases}$$

(An equivalent choice is $w - \theta\|v\|e_1$.)

Householder method

Using Householder matrices one can obtain a **QR factorization** of a matrix $A \in \mathbb{C}^{n\times n}$ in $n-1$ steps.

At the first step, if the first column of A, say c_1, is not a multiple of e_1, then one constructs a Householder matrix P_1 such that $P_1 c_1 = \alpha_1 f_1$, else set $P_1 = I$. We have

$$A_2 = P_1 A = \begin{bmatrix} \alpha_1 & * \\ 0 & B_2 \end{bmatrix}.$$

The procedure can be repeated finding an Householder matrix (or the identity) $\widetilde{P}_2 \in \mathbb{C}^{(n-1)\times(n-1)}$ such that $\widetilde{P}_2 c_2 = \alpha_2 e_1$, where c_2 is the first column of B_2.

Householder method

Using Householder matrices one can obtain a **QR factorization** of a matrix $A \in \mathbb{C}^{n\times n}$ in $n-1$ steps.

At the first step, if the first column of A, say c_1, is not a multiple of e_1, then one constructs a Householder matrix P_1 such that $P_1 c_1 = \alpha_1 f_1$, else set $P_1 = I$. We have

$$A_2 = P_1 A = \begin{bmatrix} \alpha_1 & * \\ 0 & B_2 \end{bmatrix}.$$

The procedure can be repeated finding an Householder matrix (or the identity) $\widetilde{P}_2 \in \mathbb{C}^{(n-1)\times(n-1)}$ such that $\widetilde{P}_2 c_2 = \alpha_2 e_1$, where c_2 is the first column of B_2.Defining the matrix $P_2 = \begin{bmatrix} 1 & 0 \\ 0 & \widetilde{P}_2 \end{bmatrix}$ (that is still an Householder matrix or the identity), one has

$$A_3 = P_2 A_2 = \begin{bmatrix} 1 & 0 \\ 0 & \widetilde{P}_2 \end{bmatrix} \begin{bmatrix} \alpha_1 & * \\ 0 & B_2 \end{bmatrix} = \begin{bmatrix} \alpha_1 & * & * \\ 0 & \alpha_2 & * \\ 0 & 0 & B_3 \end{bmatrix}.$$

Householder method

The procedure can be repeated $n-1$ times obtaining at the end an **upper triangular matrix**

$$R = P_{n-1} P_{n-2} \cdots P_2 P_1 A,$$

from which we get the factorization

$$A = QR, \qquad Q = (P_{n-1} \cdots P_1)^{-1} = P_1 P_2 \cdots P_{n-1},$$

where we have used the fact that Householder matrices are unitary and Hermitian.

Since the vector parameter of the Householder matrices can be explicitly constructed, then the previous construction can be seen as an algorithm.

In the real case, at each step there are two choices, one can reflect using $v + \|v\| f_1$ or $v - \|v\| f_1$. This is mathematically equivalent, and one can choose the reflection in order to get a positive diagonal entry on the matrix R. In practice, one chooses the sign that gives the better numerical stability, that is "plus" when v_1 is positive and "minus" when v_1 is negative.

The Cartan-Dieudonné theorem.

The Householder method provide a constructive proof of the following.

Theorem

Any nontrivial linear isometry in $\mathbb{R}^n$ is the composition of at most n reflections with respect to hyperplanes.

Proof.

Let Q be the orthogonal matrix associated with the isometry in an orthogonal basis of $\mathbb{R}^n$. Applying the Householder method to Q, where the reflections are chosen so that the first $n-1$ diagonal elements of R are positive, yields

$$Q = P_1 P_2 \cdots P_{n-1} R,$$

where $P_1, \ldots, P_{n-1}$ are reflection or the identity, while R is upper triangular and orthogonal with positive diagonal elements, except at most r_{nn}. □

The Cartan-Dieudonné theorem.

The Householder method provide a constructive proof of the following.

Theorem

Any nontrivial linear isometry in $\mathbb{R}^n$ is the composition of at most n reflections with respect to hyperplanes.

Proof.

$$Q = P_1 P_2 \cdots P_{n-1} R,$$

The matrix R is a phase matrix and thus there are two cases: either $R = I$ or $R = \mathrm{diag}(1, \ldots, 1, -1)$, that is a reflection. Equation above is thus a decomposition of Q as the product of at most n reflections. □

The Cartan-Dieudonné theorem.

The Householder method provide a constructive proof of the following.

Theorem

Any nontrivial linear isometry in $\mathbb{R}^n$ is the composition of at most n reflections with respect to hyperplanes.

Proof.

□

The theorem is not true for the identity in dimension one since the isometry $x \to x$, corresponding to the matrix [1], cannot be written as the product of one reflection.

Implementation

The idea of the implementation is not to construct every reflection matrix, but to use just the parameter vectors.

The first step, with $P_1 = I - \beta w_1 w_1^*$, where $\beta = 1/\|w_1\|^2$, requires the product $P_1 A$. Without constructing P_1, one can compute the product as $(I - \beta w_1 w_1^*)A = A - \beta w_1 (w_1^* A)$ and this requires

- one vector norm and one multiplication to get β ($2n$ operation and one square root);
- one matrix multiplication $u = A^* w_1$ ($2n^2 - n$ operations);
- one scalar-vector product $v = \beta w_1$ (n operations);
- the product vu^* (n^2 operations);
- the sum $A - vu^*$ (n^2 operations).

The cost of the first step is then $4n^2 + o(n^2)$ operations.

Implementation

The k-th step require the same procedure in the right-lower $(n-k+1)\times(n-k+1)$ block, say B_k and thus the cost is of about $4(n-k+1)^2$ operations.

The total asymptotic cost of the Householder algorithm for computing R is of $\sum_{k=1}^{n-1} 4(n-k+1)^2 \sim \frac{4}{3}n^3$ arithmetic operations. This allows one to solve a linear system, but **does not compare favorably with Gaussian elimination** whose cost is $\frac{2}{3}n^3$ operations.

Implementation

A pseudocode implementation is

```
for k = 1 :  n-1
  B = A(k:n, k:n);
  w = B(:,k); % select the column k from the index k to the end
  w(1) = w(1) + sign(w(1)) * norm(w);
  beta = 2 / norm(w);
  u = w' * B;
  A(k:n, k:n) = B - (beta * w) * u;
end
```

If one needs the Q factor a further computation is needed.

Exercise

Describe a method to construct the factor Q of the QR factorization using Householder matrices and that require $\frac{4}{3}n^3 + o(n^3)$ operations.

Thin QR factorization

If $A \in \mathbb{C}^{m\times n}$ the algorithms for the QR factorization can be applied. If $m > n$, n steps are needed, since also the last column should be processed.

We get $A = QR$ with $Q \in U(m)$ and $R \in \mathbb{C}^{m\times n}$ upper triangular (a rectangular matrix R is upper triangular if $r_{ij} = 0$ for $i > j$).

When $m > n$ we can block the factor as $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ with $Q_1 \in \mathbb{C}^{m\times n}$ and $R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$, with $R_1 \in \mathbb{C}^{n\times n}$.

Using the block multiplication we discover that $A = Q_1 R_1$, where R_1 is square and upper triangular. This factorization is said to be **thin QR factorization** and might be useful in some cases. Notice that $Q_1 \in \mathrm{St}(m, n)$.

- Fattorizzazione QR:

$A \in \mathbb{C}^{n\times n}$, $A = QR$ fatt QR di A se Q è unitaria ($Q^*Q = I \Leftrightarrow Q^* = Q^{-1} \Leftrightarrow QQ^* = I$) e R triangolare superiore

Ricordiamo che se Q unitaria $\|Q\|_2 = 1$

$\Rightarrow \sqrt{\rho(Q^*Q)} = 1 \Rightarrow \mu(A) = \|A^{-1}\|\|A\|$

$\mu(Q) = \|Q^{-1}\|\|Q\|$

↑ condizionamento

Q^* è unitaria perché $Q^*(Q^*)^* = Q^*Q = I$

Inoltre, A invertibile,

$\|I\| = \|AA^{-1}\| \leq \|A\|\|A^{-1}\| = \mu(A)$

Se la norma è indotta $\|A\|_2 = \max_{x\neq 0} \frac{\|Ax\|}{\|x\|} \Rightarrow \|I\| = 1$

$\Rightarrow \mu(A) \geq 1$

OSS che il condizionamento dipende dalla norma, ad esempio con $\|\cdot\|_F$ $\|I\|_F = \sqrt{n} \Rightarrow \mu_F(A) \geq \sqrt{n}$

↑ Frobenius

Va bene comunque utilizzare qualsiasi norma

Se $A \in \mathbb{R}^{n\times n}$, allora $Q \in O(n)$ (ortogonale $Q^TQ = I$), $R \in \mathbb{R}^{n\times n}$ triangolare sup

Se $A \in \mathbb{C}^{m\times n}$ $A = QR$ $Q \in U(m)$ (unitaria) $R \in \mathbb{C}^{m\times n}$ $r_{ij} = 0 \ \forall i > j$

Ci chiediamo se esiste la fatt QR, e a differenza della fatt LU, la QR esiste sempre, dimostriamolo per induzione:

$n = 1$ $A = QR = [1]A$

$n > 1$ $(n-1 = 0n)$

Sia $v \in \mathbb{C}^n$ la prima colonna di A

$$q_1 = \begin{cases} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} & \text{se } v = 0 \\ \frac{v}{\|v\|} & \text{se } v \neq 0 \end{cases}$$

$\text{span}^{\perp}\{q_1\}$ è un sp vett di dim $n-1$ e uno sua base ortonormale è $q_2, \dots, q_n$

Costruiamo $U = \left(q_1 \middle| Q_2 \right)$, $Q_2 = \left(q_2 \middle| \dots \middle| q_n \right)$

U è unitaria (per vedremo)

$$A = \left[v \middle| B \right]$$

$\uparrow$
$\mathbb{C}^{n \times (n-1)}$

$$U^* A = \left[\frac{q_1^*}{Q_2^*} \right] \left[v \middle| B \right] = \left[\frac{q_1^* v \;\middle|\; q_1^* B}{Q_2^* v \;\middle|\; Q_2^* B} \right] = \left[\frac{\|v\| \;\middle|\; q_1^* B}{0 \;\middle|\; Q_2^* B} \right]$$

$$q_1^* v = \frac{v^* v}{\|v\|} = \frac{\|v\|^2}{\|v\|} = \|v\|$$

$$Q_2^* v = \left[\frac{q_2^*}{\vdots} \right] v = \left[\frac{q_2^* v}{\vdots} \right] = 0$$

$$q_n^* v$$

Per ipotesi induttiva $\exists \tilde{Q} \in U(n-1)$ e $\tilde{R} \in \mathbb{C}^{(n-1) \times (n-1)}$ triang sup t.c $Q_2^* B = \tilde{Q} \tilde{R}$

Creiamo

$$\hat{Q} = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{Q}^* \end{bmatrix} \in U(n)$$

$$\hat{Q}(U^*A) = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{Q} \end{bmatrix} \left[\begin{array}{c|c} \|v_1\| & q_1^* B \\ \hline 0 & Q_2^* B \end{array}\right] = \begin{bmatrix} \|v_1\| & q_1^* B \\ 0 & \tilde{Q}^*(Q_2^* B) \end{bmatrix}$$

ove $\tilde{R} = \tilde{Q}^* Q_2^* B$ $\left(\hat{Q} U^* A = R \Rightarrow A = \underbrace{U^* \hat{Q}^{-1}}_{Q} R\right.$

$$\Rightarrow \hat{Q}(U^*A) = \begin{bmatrix} \|v_1\| & * \\ 0 & \tilde{R} \end{bmatrix} = R \quad \square$$

Dimostriamo che se $U \in U(n)$ $\begin{bmatrix} I_{n-1} & 0 \\ 0 & U \end{bmatrix}$ è unitaria

$$\begin{bmatrix} I_{n-1} & 0 \\ 0 & U \end{bmatrix}^* \begin{bmatrix} I_{n-1} & 0 \\ 0 & U \end{bmatrix} = \begin{bmatrix} I_{n-1}^* & 0 \\ 0 & U^* \end{bmatrix} \begin{bmatrix} I_{n-1} & 0 \\ 0 & U \end{bmatrix} =$$

$$= \begin{bmatrix} I_{n-1} & 0 \\ 0 & U^* \end{bmatrix} \begin{bmatrix} I_{n-1} & 0 \\ 0 & U \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & U^* U \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} = I$$

$\Rightarrow$ è unitaria

Se $U = \left[q_1 \mid \cdots \mid q_n\right]$ unitaria $\Rightarrow q_1, \dots, q_n$ base ortonorm di $\mathbb{C}^n$

Vediamo ora l'unicità della fattorizzazione QR:

$n=1$ $\quad a = 1 \cdot a$ oppure $a = (-1)(-a)$ oppure
$a = i(-ia)$

Le matrici unitarie 1×1 sono:

$$U(1) = \{x^* x = 1\} \Rightarrow \bar{x} x = 1 \Rightarrow |x|^2 = 1 \Rightarrow |x| = 1$$

$\Rightarrow U(1) = \{ e^{i\theta} \mid \theta \in \mathbb{R} \}$ è un gruppo in $\mathbb{C}$

Ci sono infinite sol quand per $n=1$

$\rightarrow$ ha elemento positivo nel campo $\mathbb{R}$

Se $a \neq 0$ $\quad a = \underbrace{\frac{a}{|a|}}_{Q} \cdot \underbrace{|a|}_{R}$ questa è unica

Allora supponiamo di avere

$A = Q_1 R_1 = Q_2 R_2$, se $A \in GL(n, \mathbb{C})$

$\Rightarrow R_1, R_2$ invertibili ($r_{ii} \neq 0 \quad \forall i = 1, \dots n$)

$$\Rightarrow Q_1^* Q_1 R_1 R_2^{-1} = Q_1^* Q_2 R_2 R_2^{-1} \Rightarrow R_1 R_2^{-1} = Q_1^* Q_2 = F$$

$\rightarrow$ Se ci sono due fatt $QR \Rightarrow F$ è contemporaneamente unitaria e triangolare superiore

$$\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ & \ddots & & \vdots \\ 0 & & & r_{nn} \end{bmatrix} = \begin{bmatrix} e^{i\theta_1} & 0 & \cdots & 0 \\ 0 & e^{i\theta_2} & & \vdots \\ \vdots & & \ddots & \\ 0 & 0 & \cdots & e^{i\theta_n} \end{bmatrix}$$

Se fosse anche unitaria $\rightarrow$ le colonne sono vettori di norma 1 e anche le righe

$$\Rightarrow \sqrt{\sum_{i=0}^{n} |r_{i1}|^2} = 1 \Rightarrow |r_{11}| = 1 \Rightarrow r_{11} = e^{i\theta_1}$$

$$\Rightarrow \sqrt{|e^{i\theta_1}| + |r_{12}|^2 + \cdots + |r_{1n}|^2} = 1$$

$$\Rightarrow |r_{12}|^2 + \cdots + |r_{1n}|^2 = 0$$

$\rightarrow F$ è diagonale con elementi di modulo 1 sulla diagonale. Una matrice del genere è detta di fase

Teorema (slide)

...

Se la matrice non è invertibile ⇒ è + complicato

Come calcolare la fatt QR?
(cerco un algoritmo migliore di Gauss ⇒ vediamo se lo trovo)
Faccio rotazioni e riflessioni (↗ da Gauss che è combinazione di righe)

$$A = QR \Rightarrow Q^* A = R$$

$$Q^* \left[c_1 \mid \cdots \mid c_n \right] = \left[r_1 \mid \cdots \mid r_n \right]$$

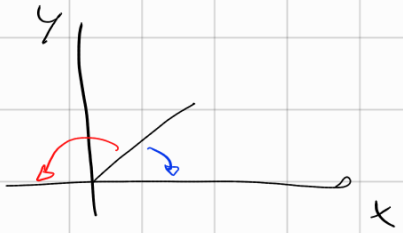
$$\Rightarrow \left[Q^* c_1 \mid \cdots \mid Q^* c_n \right] = \left[r_1 \mid \cdots \mid r_n \right]$$

trovare U unitaria per cui $U c_i = r_i \quad i = 1, \dots, n$

Ci sono 3 modi per calcolare la fatt QR
Usiamo le rotazioni planari:

Givens matrices

Dato $v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \in \mathbb{R}^2$
esiste una rotazione lineare che manda v sull'asse x? Sì.



$$O(n) = \{ Q \in \mathbb{C}^{n \times n} \mid Q^T Q = I \} \quad \det Q = \pm 1$$

$$1 = \det Q^T Q = \det Q^T \det Q = \det^2 Q$$

$SO(n) = \{ Q \in O(n), \det Q = 1 \}$ è un gruppo

↑ gruppo ortogonale speciale (rotazioni)

Le rotazioni $SO(2)$ sono $\left\{ \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix}, \theta \in [0, 2\pi] \right\}$

rotazione di angolo θ antiorario

La matrice complessa di Givens è

$$G = \begin{bmatrix} c & -\bar{s} \\ s & c \end{bmatrix} + \text{slides.}$$

devo trovare G per cui

$$\begin{bmatrix} c & -\bar{s} \\ s & c \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix} \quad \dots$$

Una sola rotazione mi permette di trovare la fattorizzazione QR.

$A = QR$, $Q \in U(n)$ R trang superiore

$$\begin{bmatrix} a \\ b \end{bmatrix} \in \mathbb{R}^2 \to \begin{bmatrix} \alpha \\ 0 \end{bmatrix}$$

$$|\alpha| = \sqrt{|a|^2 + |b|^2}$$

↖ Matrice di Givens.

$$\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \quad \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \quad c^2 + s^2 = 1$$

$$\begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix} \quad \begin{cases} ca - sb = \alpha \\ sa + cb = 0 \end{cases}$$

Se $b=0 \Rightarrow c=1$ e $s=0$ $(c=-1, s=0)$

Se $b\neq 0 \Rightarrow c=-\frac{sa}{b} \Rightarrow -\frac{sa^2}{b}-sb=s$

$$-\frac{s(a^2+b^2)}{b}=\pm\sqrt{a^2+b^2} \Rightarrow s=\mp\frac{b}{\sqrt{a^2+b^2}}$$

$$c=\pm\frac{a}{\sqrt{a^2+b^2}}$$

CASO COMPLESSO:

$$\begin{bmatrix} c & -\bar{s} \\ s & c \end{bmatrix}$$ Non sono rotazioni nel caso complesso

$$c=\cos\varphi=\frac{|a|}{\sqrt{|a|^2+|b|^2}}, \quad s=\operatorname{sen}\varphi\cdot e^{i\theta}$$

$a\neq 0\neq b$

$$\operatorname{sen}\varphi=\frac{|b|}{\sqrt{|a|^2+|b|^2}}, \quad \frac{\varphi}{|a|}=\frac{a}{|a|}\cdot\frac{|b|}{b}$$

Se $a=0$ $\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$, $c=0$, $s=\pm 1$

Vediamo una rotazione ortonormale su $\mathbb{R}^3$

- Sul piano (x_1, x_2)

$$\begin{bmatrix} \cos\theta & -\operatorname{sen}\theta & 0 \\ \operatorname{sen}\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}=\begin{bmatrix} G\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\ x_3 \end{bmatrix}$$

- Sul piano (x_2, x_3)

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\operatorname{sen}\theta \\ 0 & \operatorname{sen}\theta & \cos\theta \end{bmatrix}$$

- Sul piano (x_1, x_3) $\begin{bmatrix} \cos\theta & 0 & -\operatorname{sen}\theta \\ 0 & 1 & 0 \\ \operatorname{sen}\theta & 0 & \cos\theta \end{bmatrix}$

Definiamo allora la matrice di Givens su $\mathbb{R}^n$ nel caso dei complessi:

Una matrice di Givens è del tipo

$$G_{ij} = \begin{bmatrix} 1 & & 0 & & -\bar{s} & & 0 \\ & \ddots & & & & & \\ & & 1 & & & & \\ 0 & & & c & & & 0 \\ & & & & \ddots & & \\ & s & & & & c & \\ & & & & & & \ddots \\ 0 & & 0 & & & & 1 \end{bmatrix}$$

(rotazione i, j)

Agisce nel piano (x_i, x_j)

$$G_{ij} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_{i-1} \\ cx_i - \bar{s}x_j \\ x_{i+1} \\ \vdots \\ x_{j-1} \\ sx_i + cx_j \\ x_{j+1} \\ \vdots \\ x_n \end{bmatrix}$$

$$(G_{ij})_{hl} = \begin{cases} 1 & h = l,\ h \neq i,\ h \neq j \\ c & h = l = i,\ h = l = j \\ s & h = j,\ l = i \\ -\bar{s} & h = i,\ l = j \\ 0 & \text{altrove} \end{cases}$$

Metodo di Givens per la fattorizzazione QR:

- PASSO 1: Come eliminare gli elementi di un vettore tramite "rotazione" (per i complessi non è rotazione)

$v \in \mathbb{C}^n$ $\quad v = \begin{bmatrix} v_1^{(1)} \\ \vdots \\ v_n^{(1)} \end{bmatrix}$ Trovare matrici di Givens t.c $\quad Gv = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

$$G v = G \begin{bmatrix} v_{n-1}^{(1)} \\ v_n^{(1)} \end{bmatrix} = \begin{bmatrix} v_{n-1}^{(2)} \\ 0 \end{bmatrix}$$

$$G_{n-1,n} = \begin{bmatrix} 1 & & & \\ & \ddots & & 0 \\ & & 1 & \\ & 0 & & G \end{bmatrix}$$

$$G_{n-1,n} \begin{bmatrix} v_1^{(1)} \\ \vdots \\ v_n^{(1)} \end{bmatrix} = \begin{bmatrix} v_1^{(1)} \\ \vdots \\ v_{n-2}^{(1)} \\ v_{n-1}^{(2)} \\ 0 \end{bmatrix} = \begin{bmatrix} v_1^{(2)} \\ \vdots \\ v_{n-2}^{(2)} \\ v_{n-1}^{(2)} \\ 0 \end{bmatrix}$$

PASSO 2:

$$G \begin{bmatrix} v_{n-2}^{(2)} \\ v_{n-1}^{(2)} \end{bmatrix} = \begin{bmatrix} v_{n-2}^{(3)} \\ 0 \end{bmatrix}$$

$$G_{n-2,n-1} = \begin{bmatrix} 1 & & & \\ & \ddots & & 0 \\ & & G & \\ & 0 & & 1 \end{bmatrix}$$

$$G_{n-2,n-1} \begin{bmatrix} v_1^{(2)} \\ \vdots \\ v_{n-1}^{(2)} \\ 0 \end{bmatrix} = \begin{bmatrix} v_1^{(2)} \\ \vdots \\ v_{n-3}^{(2)} \\ v_{n-2}^{(2)} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} v_1^{(3)} \\ \vdots \\ v_{n-2}^{(3)} \\ 0 \\ 0 \end{bmatrix}$$

Dopo $n-1$ passi si ottiene un vettore $\begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix}$

$$G_{12} \cdot G_{23} \cdot G_{34} \cdot \ldots \cdot G_{n-2,n-1} \cdot G_{n-1,n} \cdot v = \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Se $v \in \mathbb{R}^n$ è possibile scegliere l'ultima G_{12} t.c $\alpha > 0$

Se parto ora da $A_1 = \left[\, v \mid B_1 \,\right]$ e applico $G_{12} \cdots G_{n-1,n}$

$$\Rightarrow \underbrace{G_{12} \cdots G_{n-1,n}}_{Q} \cdot A_1 = Q \left[\, v \mid B_1 \,\right] = \left[\, Q_1 v \mid Q_1 B_1 \,\right] =$$

$$= \left[\begin{array}{c|c} \gamma_1 & * \\ \hline 0 & B_2 \end{array}\right] = A_2$$

Posso interpretare questo come il primo passo di un algoritmo per calcolare la fatt QR di A.
Sia $v^{(1)}$ la prima colonna di A, si costruiscono $G_{12}^{(1)}, \ldots, G_{n-1,n}^{(1)}$ e le applichiamo ad A

$$G_{12}^{(1)} \cdots G_{n-1,n}^{(1)} \, v^{(1)} = \begin{bmatrix} \gamma_1 \\ 0 \end{bmatrix}$$

$$\Rightarrow A_2 = G_{12}^{(1)} \cdots G_{n-1,n}^{(1)} \, A_1 = \begin{bmatrix} \gamma_1 & * \\ 0 & B_2 \end{bmatrix} = A_2$$

$$\in \mathbb{C}^{(n-1)\times(n-1)}$$

$A_2 = \left[\begin{array}{c|c} \gamma_1 & * \\ \hline 0 & B_2 \end{array}\right]$, sia $v^{(2)}$ la prima colonna di B_2

$\exists\, \tilde{G}_{12}^{(2)}, \ldots, \tilde{G}_{n-2,n-1}^{(2)}$ di Givens t.c

$$\tilde{G}_{12}^{(2)} \cdots \tilde{G}_{n-2,n-1}^{(2)} \, v^{(2)} = \begin{bmatrix} \gamma_2 \\ 0 \end{bmatrix} \Rightarrow B_2 = \left[\begin{array}{c|c} \gamma_2 & * \\ \hline 0 & B_3 \end{array}\right]$$

Per ogni A_2 si costruiscono delle Givens estese ad A_2 che lasciano la prima componente invariata:

$$G_{23}^{(2)} = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{G}_{12}^{(2)} \end{bmatrix}, \dots, G_{i,i+1}^{(2)} = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{G}_{i-1,i}^{(2)} \end{bmatrix}$$

con $i = 2, \dots, n-1$

Il secondo passo è dunque

$$G_{23}^{(2)} G_{34}^{(2)} \cdots G_{n-1,n}^{(2)} A_2 = \left[\begin{array}{c|cc} \gamma_1 & * & \\ \hline 0 & \gamma_2 & * \\ \hline & 0 & B_3 \end{array}\right] = A_3$$

Al passo k sono necessarie $n-k$ rotazioni:
Sia $v^{(k)}$ la prima colonna di B_k, costruisco

$$\tilde{G}_{12}^{(k)} \cdots \tilde{G}_{n-k,n-k+1}^{(k)} v^{(k)} = \begin{bmatrix} \gamma_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$G_{k,k+1}^{(k)} = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{G}_{k,k+1}^{(k)} \end{bmatrix} \cdots$$

$$A_{k+1} = G_{k,k+1}^{(k)} \cdots G_{n-1,n}^{(k)} A_k \qquad k = 1, \dots, n-1$$

$$R = A_n = G_{n-1,n}^{(n-1)} A_{n-1} = G_{n-1,n}^{(n-1)} G_{n-2,n-1}^{(n-2)} G_{(n-1,n)}^{(n-2)} A_{n-2} =$$

$$= \underbrace{G_{n-1,n}^{(n-1)} \cdot G_{n-2,n-1}^{(n-2)} \cdot G_{n-1,n}^{(n-2)} \cdots G_{k,k+1}^{(k)} \cdots G_{k-1,n}^{(k)} \cdots G_{12}^{(1)} \cdots G_{n-1,n}^{(1)}}_{U} \cdot A$$

$$R = UA \Rightarrow A = U^* R \quad , \quad A = QR$$

In tutto si hanno $n-k$ ad ogni passo k

$$\Rightarrow \sum_{k=1}^{n-1} (n-k) = \sum_{k=1}^{n-1} k = \frac{n(n-1)}{2}$$

A differenza della fatt LU, la QR ∃ sempre perché il nostro algoritmo non ha condizioni (ad es Gauss deve PIVOT ≠ 0)

```
G = givens(a,b)

For l = 1 : n-1
    % passo l
    For i = l : n-l
        % "costruiamo" G^(l)_{i,i+1} e "moltiplichiamo" per A
        G = givens(A(i,l), A(i,l+1));
        A(i:i+1, l:n) = G * A(i:i+1, l:n);
```

Costo computazionale:

$6(n-l)$ operazioni
↑ 2 combinazioni lineari di righe

$$\Rightarrow \sum_{l=1}^{n-1} \sum_{i=l}^{n-1} 6(n-l+1) = \sum_{l=1}^{n-1} 6(n-l+1)(n-l) =$$

$$= \sum_{l=1}^{n-1} 6(n-l)^2 + \sum_{l=1}^{n-1} 6(n-l) \sim \frac{6n^3}{3} + \Theta(n^3) \sim 2n^3$$

Questo costo computazionale è un po' più alto di quello dell'algoritmo di Gauss, ma ho motivi evitare ed ho informazioni in più.

Se applico l'algoritmo a una matrice unitaria

$Q \in \mathbb{R}^{3\times 3}$ $\quad Q^T Q = I$ $\quad \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix}$

con una rotazione $(1,2) \longrightarrow$ (posso anche $\theta \in [0,\pi]$) $\begin{bmatrix} q_{11} & q_{12} & q_{13} \\ \tilde{q}_{21} & \tilde{q}_{22} & \tilde{q}_{23} \\ 0 & \tilde{q}_{32} & \tilde{q}_{33} \end{bmatrix}$

$\xrightarrow[(1,3)]{\text{rot}} \begin{bmatrix} x & 0 & 0 \\ 0 & \hat{q}_{22} & \hat{q}_{23} \\ 0 & \tilde{q}_{32} & \tilde{q}_{33} \end{bmatrix} \xrightarrow[(2,3)]{\text{rot}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & q_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \pm 1 \end{bmatrix}$ (perché unitaria)

$\Rightarrow$ Con 3 rotazioni sono riuscita ad ottenere una matrice unitaria con det 1 v -1

Se $Q \in SO(3)$ $\exists\, G_{23}, G_{12}, \tilde{G}_{23}$ t.c

$$G_{23} G_{12} \tilde{G}_{23} Q = I \Rightarrow Q = \tilde{G}_{23}^T G_{12}^T G_{23}^T$$

$SO(3)$ è varietà topologica di dim 3 (si dimostra che è differenziabile)

Posso fattorizzare una matrice $SO(n)$ con n^2 angoli (parametri)

Fattorizzazione QR magra (Thin QR factorization)

Se $A \in \mathbb{C}^{m\times n}$, $m \geq n$, $\exists\, U \in U(m)$, $R \in \mathbb{C}^{m\times n}$ t.c

$r_{ij} = 0$ $i > j$, $A = QR$ (trang sup)

$$A = QR = \left[\underbrace{Q_1}_{n} \,\middle|\, \underbrace{Q_2}_{m-n}\right] \begin{bmatrix} R_1 \\ 0 \end{bmatrix}^{m\times n} = Q_1 R_1 + Q_2 0$$

Osserviamo che $Q_1 \in St(m,n,\mathbb{C}) = \{U \in \mathbb{C}^{m\times n},\ U^*U = I\}$
Varietà

14/10

Def

$A \in \mathbb{C}^{m\times n}$ è detta di Hessenberg superiore se

$a_{ij} = 0 \quad i > j+1$

Metodo di Householder:
impiega $\frac{4}{3}n^3$ operazioni per R e $\frac{4}{3}n^3$ per Q

$A \in \mathbb{C}^{m\times n}$ $m \geq n$

Supponiamo di avere una fattorizzazione QR di A
Trovare il punto di minima norma $\|Ax-b\|$

$$\|Ax-b\|^2 = \|QRx - \underbrace{QQ^*}_{=I} b\|^2 = \|Q(Rx - Q^*b)\|^2 =$$

$$R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix} \qquad Q^* = \begin{bmatrix} Q_1^* \\ Q_2^* \end{bmatrix}$$

$$Q = \underbrace{\underbrace{[Q_1 | Q_2]}_{n}}_{m}$$

$$= \left\| \begin{bmatrix} R_1 x - Q_1^* b \\ Q_2^* b \end{bmatrix} \right\| = \|R_1 x - Q_1^* b\|^2 + \|Q_2^* b\|^2$$

Ma la x è presente solo nel primo addendo

$$\Rightarrow \operatorname{argmin} \|Ax-b\|^2 = \operatorname{argmin} \|R_1 x - Q_1^* b\|^2$$

Se $\operatorname{rg} A = n$, $\underset{m\times n}{A} = \underset{m\times n}{Q}\underset{n\times n}{R}$, $R = \begin{bmatrix} R_1 \\ 0 \end{bmatrix}$

R_1 è invertibile

$\Rightarrow R_1 x = Q^* b$ ha sol unica $\Rightarrow$ è anche soluzione dei minimi quadrati.

Una base ortogonale del sottospazio è una base della spazio ortogonale:

$$V = \text{span}\{v_1, \dots, v_l\} \subset \mathbb{C}^n, \quad v_1, \dots v_l \text{ L.I.}$$

(Si trova una base ortogonale con l'algoritmo di Gram-Schmidt in geometria)

$$V = \left[v_1 \mid v_2 \mid \dots \mid v_l \right] \in \mathbb{C}^{n \times l}$$

$$V = QR = \left[Q_1 Q_2 \right] \begin{bmatrix} R_1 \\ 0 \end{bmatrix} = Q_1 R_1$$

(fattorizzazione magra)

$$\Rightarrow v_i = \sum_{j=1}^{l} q_j r_{ji}$$

$q_1, \dots, q_l$ base di V, $q_j^* q_l = 0 \quad j > l$

$\Rightarrow q_{l+1}, \dots, q_n$ base di $V^\perp$

Algoritmo di Gram-Schmidt:

$v_1, \dots v_l \quad (\text{L.I.} \Rightarrow v_i \neq 0 \ \forall i)$

Voglio trovare una base ortonormale $w_1, \dots w_l$

$$w_1 = \frac{v_1}{\|v_1\|} \qquad w_2 = \frac{v_2 - \langle v_2, w_1 \rangle w_1}{\|v_2 - \langle v_2, w_1 \rangle w_1\|}$$

$w_2 \neq 0$ perché L.I.

$$\Rightarrow \quad w_i = \frac{v_i - \sum_{j=1}^{i} \langle v_i, w_j \rangle w_j}{\| v_i - \sum_{j=1}^{i} \langle v_i, w_j \rangle w_j \|}$$

OSS:

$v_2 = \|v_2\| w_2 + \langle v_2, w_1 \rangle w_1$ (è comb lineare di w_1 e w_2)

$$v_i = \sum_{j=1}^{i} \langle v_i, w_j \rangle w_j + \|u_i\| w_i$$

h_{ji} $\quad$ h_{ii}

è comb lineare

$\sum$ è w_i non normalizzato

$$W = \left[w_1 \mid \cdots \mid w_l \right]$$

$$v_i = \left[w_1 \mid \cdots \mid w_l \right] \begin{bmatrix} h_{1j} \\ h_{2j} \\ \vdots \\ h_{ii} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\Rightarrow V = \left[v_1 \mid \cdots \mid v_l \right] = \left[w_1 \mid \cdots \mid w_l \right] \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1l} \\ & h_{22} & & \\ & & \ddots & \\ 0 & & & h_{ll} \end{bmatrix}$$

fattorizzazione QR magia

Costo computazionale

$2n$ ops per $\langle \cdot, \cdot \rangle$

n ops quando $\langle \cdot, \cdot \rangle \cdot w_i$

n ops con le somme

$$\Rightarrow \quad 4n \sum_{i=1}^{l} \sum_{j=1}^{i-1} 1 + \sum_{i=1}^{l} (2n + \gamma) =$$

γ ops per la radice in precisione u

The SVD approach

The linear least squares can be solved with SVD. See the next lecture.

$$= 4n \sum_{i=1}^{l} i - 1 \sim 4n \frac{l^2}{2} = 2nl^2$$

Se $l = n \Rightarrow 2n^3$ (come Givens)

L'algoritmo di Gram-Schmidt è però poco stabile

15/10

SVD

$A \in \mathbb{C}^{m \times n}$, $A = U \Sigma V^*$, $U \in U(n)$, $V \in U(n)$,

, $\Sigma \in \mathbb{R}^{m \times n}$ t.c $\sigma_{ij} = 0 \quad i \neq j$

$\sigma_{ii} = \sigma_i \quad i = 1, \dots, p$,

$p = \min\{m, n\}$

$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$

$$\begin{bmatrix} & \\ & \end{bmatrix} = \begin{bmatrix} U \end{bmatrix} \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_p \\ & & \vdots \end{bmatrix} \begin{bmatrix} V^* \end{bmatrix}$$

$$\underset{m \times n}{\begin{bmatrix} & \\ & \end{bmatrix}} = \begin{bmatrix} U \end{bmatrix} \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_n \\ 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} V^* \end{bmatrix}$$

$\sigma_1, \dots, \sigma_p$ sono tutti i valori singolari di A

le colonne di U e V sono tutti i vettori singolari

PCA principal component analysis

Teorema:
SLIDE

(Teorema simile al Teorema spettrale)

Dimostrazione:

Per induzione su n e per ogni m

- Se $A = 0 \Rightarrow A = U 0 V^*$ è una SVD

- Se $A \neq 0 \Rightarrow A = U \Sigma V^*$

$\Rightarrow A^* = V \Sigma^* U$ è una SVD. (sappiamo $m \geq n$ senza perdita di gen.)

$\Sigma \uparrow \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ 0 & & 0 \end{bmatrix}$

Se $n = 1$, $A \in \mathbb{C}^{n \times 1} \Rightarrow \exists$ fatt QR

$\Rightarrow A = Q \begin{bmatrix} \alpha \\ 0 \\ \vdots \\ 0 \end{bmatrix} [1] = \quad |\alpha| = \|A\| > 0 \quad A \neq 0$

$\alpha = |\alpha| \left(\frac{\alpha}{|\alpha|}\right) \rightarrow U(1)$

$= Q \begin{bmatrix} |\alpha| \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{bmatrix} \frac{\alpha}{|\alpha|} \end{bmatrix}$

$\Rightarrow$ OK

$n - 1 \Rightarrow n$

$A \in \mathbb{C}^{m \times n}$, $m \geq n$, $A \neq 0$ $\left(\|A\|_2 = \max_{\|x\| = 1} \|Ax\|\right)$

$\exists x \in \mathbb{C}^n$ ($\|x\| = 1$) t.c. $\|Ax\| = \|A\| =: \sigma_1$

$y = \frac{Ax}{\sigma_1}$, $\|y\| = \frac{\|Ax\|}{\sigma_1} = 1$

$\Rightarrow \exists$ slide

References

The material of this presentation (and the missing proofs) can be found in

- D. Bini, M. Capovani, O. Menchi, *Metodi Numerici per l'Algebra Lineare,* Zanichelli, Bologna, 1982 (Capitolo 7), in Italian.
- J. Stoer, R. Burlisch, *Introduction to Numerical Analysis, Third edition,* Texts in Applied Mathematics, 12. Springer-Verlag, New York, 2002, in English.

Regression

$Y = \beta_1 X + \beta_0$, with

$$\beta_1 = \frac{\mathrm{cov}(X, Y)}{\mathrm{var}(X)}, \qquad \beta_0 = \mathbb{E}[Y] - \beta_1 \mathbb{E}[X].$$

Regression

$Y = \beta_1 X + \beta_0$, with

$$\beta_1 = \frac{\mathrm{cov}(X, Y)}{\mathrm{var}(X)}, \qquad \beta_0 = \mathbb{E}[Y] - \beta_1 \mathbb{E}[X].$$

Observe that

$$M = \begin{bmatrix} 2n\mathbb{E}[X^2] & 2n\mathbb{E}[X] \\ 2n\mathbb{E}[X] & 2n \end{bmatrix}, \qquad c = \begin{bmatrix} 2n\mathbb{E}[XY] \\ 2n\mathbb{E}[Y] \end{bmatrix}.$$

We claim that $M \begin{bmatrix} \beta_1 \\ \beta_0 \end{bmatrix} = c$:

Regression

$Y = \beta_1 X + \beta_0$, with

$$\beta_1 = \frac{\mathrm{cov}(X, Y)}{\mathrm{var}(X)}, \qquad \beta_0 = \mathbb{E}[Y] - \beta_1 \mathbb{E}[X].$$

Observe that

$$M = \begin{bmatrix} 2n\mathbb{E}[X^2] & 2n\mathbb{E}[X] \\ 2n\mathbb{E}[X] & 2n \end{bmatrix}, \qquad c = \begin{bmatrix} 2n\mathbb{E}[XY] \\ 2n\mathbb{E}[Y] \end{bmatrix}.$$

We claim that $M \begin{bmatrix} \beta_1 \\ \beta_0 \end{bmatrix} = c$:

$$2n\mathbb{E}[X]\beta_1 + 2n\mathbb{E}[Y] - 2n\mathbb{E}[X]\beta_1 = 2n\mathbb{E}[Y]$$

Regression

$Y = \beta_1 X + \beta_0$, with

$$\beta_1 = \frac{\mathrm{cov}(X, Y)}{\mathrm{var}(X)}, \qquad \beta_0 = \mathbb{E}[Y] - \beta_1 \mathbb{E}[X].$$

Observe that

$$M = \begin{bmatrix} 2n\mathbb{E}[X^2] & 2n\mathbb{E}[X] \\ 2n\mathbb{E}[X] & 2n \end{bmatrix}, \qquad c = \begin{bmatrix} 2n\mathbb{E}[XY] \\ 2n\mathbb{E}[Y] \end{bmatrix}.$$

We claim that $M \begin{bmatrix} \beta_1 \\ \beta_0 \end{bmatrix} = c$:

$$2n\mathbb{E}[X]\beta_1 + 2n\mathbb{E}[Y] - 2n\mathbb{E}[X]\beta_1 = 2n\mathbb{E}[Y]$$

$$2n\mathbb{E}[X^2]\beta_1 + 2n\mathbb{E}[X]\mathbb{E}[Y] - 2n\mathbb{E}[X]^2\beta_1$$
$$= 2n\left(\mathrm{var}(X)\frac{\mathrm{cov}(X, Y)}{\mathrm{var}(X)} + \mathbb{E}[X]\mathbb{E}[Y]\right) = 2n\mathbb{E}[XY].$$

Hilbert spaces approximation

$\mathcal{H} = L^2_\mu$, with $\int f d\mu = \sum_i f(x_i)$.

Find the best approximation on $\mathcal{F} = \text{span}\{\varphi_0, \varphi_1\}$, $\varphi_0 = 1$, $\varphi_1 = x$ of f such that $f(x_i) = y_i$.

$$A = \begin{bmatrix} \langle \varphi_1, \varphi_1 \rangle & \langle \varphi_0, \varphi_1 \rangle \\ \langle \varphi_1, \varphi_0 \rangle & \langle \varphi_0, \varphi_1 \rangle \end{bmatrix}, \qquad b = \begin{bmatrix} \langle f, \varphi_1 \rangle \\ \langle f, \varphi_0 \rangle \end{bmatrix}.$$

Where, for instance, $\langle \varphi_1, \varphi_1 \rangle = \int x^2 d\mu = \sum x_i^2$, and $\langle f, \varphi_1 = \int f x d\mu = \sum x_i y_i$

Minimization by derivation

In the real case, with $f(x) = \|Ax - b\|^2$, we have

$$
\begin{aligned}
f(x+h) &= \\
\|A(x+h) - b\|^2 &= \langle Ax - b + Ah, Ax - b + Ah \rangle \\
&= \langle Ax - b, Ax - b \rangle + 2\langle Ax - b, Ah \rangle + \langle Ah, Ah \rangle \\
&= f(x) + \langle 2(A^T Ax - A^T b), h \rangle + hA^T Ah
\end{aligned}
$$

Minimization by derivation

In the real case, with $f(x) = \|Ax - b\|^2$, we have

$$\begin{aligned} f(x+h) &= \\ \|A(x+h) - b\|^2 &= \langle Ax - b + Ah, Ax - b + Ah \rangle \\ &= \langle Ax - b, Ax - b \rangle + 2\langle Ax - b, Ah \rangle + \langle Ah, Ah \rangle \\ &= f(x) + \langle 2(A^T Ax - A^T b), h \rangle + hA^T Ah \end{aligned}$$

and we discover the Taylor series

$$Df(x)[h] = \langle \underbrace{2(A^T Ax - A^T b)}_{\text{gradient}}, h \rangle, \qquad D^2 f(x)(h,h) = h \ \underbrace{A^T A}_{\text{Hessian}} \ h,$$

Sia $A \in \mathbb{C}^{m \times n}$ con SVD $A = U \Sigma V^*$

$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_l > \sigma_{l+1} = 0 \quad (l = \rho = n \ \sigma_{l+1} = 0)$

$$A = \left[U_1 \mid U_2 \right] \left[\begin{array}{c|c} \sigma_1 & 0 \\ \hline 0 & 0 \end{array} \right] \left[\begin{array}{c} V_1^* \\ \hline V_2^* \end{array} \right] \begin{matrix} \} l \\ \} n-l \end{matrix} =$$

$$= \left[U_1 \mid U_2 \right] \begin{bmatrix} \Sigma V_1^* \\ 0 \end{bmatrix} = U_1 \Sigma_1 V_1^* \quad \text{SVD magra}$$

$$A = \sum_{i=1}^{\min\{n,m\}} \sigma_i u_i v_i^* = U \Sigma V^*$$

$\text{Im}(A) = \text{span}\{u_1, \dots, u_l\} \quad \sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_l > \sigma_{l+1} = 0$

$\text{rg } A = l$

$\ker A = \text{span}\{v_{l+1}, \dots, v_n\}$

$\Rightarrow$ algoritmi per calcolare queste 3 cose

+ Lezione 21/10

Ripasso di SVD

22/10

$A \in \mathbb{C}^{m \times n} \quad \exists\, U \in U(m),\ V \in U(n)$

$\Sigma \in \mathbb{R}^{m \times n} \qquad \Sigma = \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_n \\ & 0 & \end{bmatrix}, \quad \sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \dots \geq \sigma_n \geq 0$

t.c $A = U \Sigma V^*$

È poi è lì è il modo magno considerando solo i valori non nulli

+SZIDE +REC

$\sigma_1 = \|A\|_2$ +chiedi

Proviamo a risolvere il problema dei minimi quadrati con la SVD:

$$\|Ax-b\|_2^2 \overset{\text{ho SVD}}{=} \|U\Sigma V^* x - UU^* b\|_2^2 = \|U(\Sigma V^* x - U^* b)\|_2^2 =$$

Poniamo $y = V^* x$, allora , $m \geq n$

$$= \left\| \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_n \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} - \begin{bmatrix} u_1^* b \\ \vdots \\ u_n^* b \\ \vdots \\ u_m^* b \end{bmatrix} \right\|_2^2 =$$

$$= \left\| \begin{bmatrix} \sigma_1 y_1 - u_1^* b \\ \vdots \\ \sigma_n y_n - u_n^* b \\ \vdots \\ -u_m^* b \end{bmatrix} \right\|^2 = \sum_{i=1}^{l} |\sigma_i y_i - u_i^* b|^2 + \sum_{i=l}^{m} |u_i^* b|^2$$

Trovare il minimo rsp a x equivale a trovare min rsp y del problema appena trovato

I punti di minimo sono quelli che annullano

$$y_i = \frac{u_i^* b}{\sigma_i} \qquad i = 1, \dots, l$$

$$y_i \in \mathbb{C} \qquad i = l+1, \dots, n$$

$$x = Vy = \begin{bmatrix} v_1, \dots, v_n \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \sum y_i v_i = \sum_{i=1}^{l} \frac{u_i^* b}{\sigma_i} v_i + \sum_{i>l} y_i v_i$$

+REC

$$\|x\|_2 = \|Vy\|_2 = \|y\|_2 = |y_1|^2 + \dots + |y_l|^2 + |y_{l+1}|^2 + \dots + |y_n|^2$$

è minimo se $y_i = \frac{u_i^* b}{\sigma_i}$ $i = 1, \dots, l$

$y_i = 0$ $i > l$

$$\Rightarrow x = \sum_{i=1}^{l} \frac{u_i^* b}{\sigma_i} v_i$$

1. Come si calcola la SVD?
2. Come si calcolano gli autovalori?

$A \in \mathbb{C}^{m \times n}$ è inv se $m = n$, $\det A \neq 0$

$AB = BA = I$

Def
Data A SVD $A = U \Sigma V^*$, la pseudoinversa è

$A^+ = V \Sigma^+ U^*$ dove $\Sigma^+ \in \mathbb{R}^{n \times m}$ è $\Sigma^+ = \begin{bmatrix} 1/\sigma_1 & & 0 \\ & \ddots & \\ 0 & & 1/\sigma_n \\ 0 & \dots & 0 \end{bmatrix}$

$$(\Sigma^+)_{ij} = \begin{cases} 1/\sigma_i & i = j \le k+1 \\ 0 & \text{altrove} \end{cases}$$

ES

$$v = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} \qquad \underbrace{Q \begin{bmatrix} 3 \\ 0 \\ 0 \end{bmatrix}}_{V} = 1 \Rightarrow V^+ = \begin{bmatrix} 1/3 & 0 & 0 \end{bmatrix} Q^*$$

PROPRIETÀ DI A^+:

- Se A è invertibile $\quad A^{-1} = A^+$

  Dim:

  $$A = U \Sigma V^* = U \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_n \end{bmatrix} V^*$$

  $$A^{-1} = (V^*)^{-1} \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_n \end{bmatrix}^{-1} U^{-1} = V \overbrace{\begin{bmatrix} 1/\sigma_1 & & \\ & \ddots & \\ & & 1/\sigma_n \end{bmatrix}}^{\Sigma^+} U^* = A^+$$

- $AXA = A, \quad XAX = X$

- Se $m \ge n$ e $\det A = n \Rightarrow A^+ = (A^* A)^{-1} A^* \qquad A^+ A = I$

  è inversa sx

  Dim:

  $$A = U \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} V^* \qquad \Sigma_1 \text{ quadrata e invertibile}$$

  $$A^+ = V \begin{bmatrix} \Sigma_1^{-1} & 0 \end{bmatrix} V^*$$

  $$A^* A = V \begin{bmatrix} \Sigma_1 & 0 \end{bmatrix} \cancel{U^* U} \begin{bmatrix} \Sigma_1 \\ 0 \end{bmatrix} V^* = V \Sigma_1^2 V^*$$

$$(A^*A)^{-1} = V(\Sigma^2)^{-1}V^*$$

$$(A^*A)^{-1}A^* = V(\Sigma_1^2)^{-1}V^*V[\Sigma_1\ 0]U^* = V\begin{bmatrix}\Sigma_1^{-1} \\ 0\end{bmatrix}U^* = A^+$$

$\|Ax-b\|^2$, le soluzioni del problema dei minimi quadrati di norma minima è $x = A^+b$ + cose

+ SLIDE + REC

Dunque $P = A\underbrace{(A^*A)^{-1}A^*}_{A^+} =$

$$= U\begin{bmatrix}\Sigma_1 \\ 0\end{bmatrix}V^*V\begin{bmatrix}\Sigma_1^{-1} & 0\end{bmatrix}U^* = U\begin{bmatrix}I & 0 \\ 0 & 0\end{bmatrix}U^*$$

$\Rightarrow \text{span}\{u_1, \dots, u_n\} = V$

Def:

$z = \rho e^{i\theta}$, $z \neq 0$

ρ ↑ modulo misura modulo del pos.

θ ↑ argomento misura dell'angolo del positivo

$A \in \mathbb{C}^{n\times n}$ $A = UH$ si chiama decomposizione polare

↑ unitaria

+ SLIDE

Se $A = \tilde{U}\Sigma\tilde{V}^*$ $\Rightarrow U = \tilde{U}\tilde{V}^*$

$$A = \underbrace{\tilde{U}\tilde{V}^*}_{U} \underbrace{\left(\tilde{V}\Sigma\tilde{V}^*\right)}_{H}$$

+ SLIDE

28/10

Trasformata discreta di Fourier

$\omega_n = e^{i\frac{2\pi}{n}}$ $\quad z^n = 1 \quad e^{i\frac{2\pi}{n}j} \quad j = 0, \dots, n-1$

ω_n radice principale dell'unità

$$\left(e^{i\frac{2\pi}{n}}\right)^l = e^{i\frac{2\pi}{n}l}$$

$$G = \left\{ e^{i\frac{2\pi}{n}j}, \; j = 0, \dots, n-1 \right\}$$

$1, \omega_n, \omega_n^2, \dots, \omega_n^{n-1}$

Una radice dell'unità è detta positiva se le sue potenze generano tutte le radici

+ SLIDES

29/10

SLIDE SU FFT

12/11

$F_n \times \quad (F_n)_{i,j} = \omega_n^{i,j} \quad i, j = 0, \dots, n-1$

$\omega_n = e^{i\frac{2\pi}{n}}$

IDFT(x) valutazione $p(z) = x_0 + x_1 z + \dots + x_{n-1} z^{n-1}$

$F_n^{-1}(x) = \frac{1}{n} F_n^* x$

FFT $\left(\frac{3}{2} n^2 \log_2 n\right)$

+ slide (n 44 del gruppo FFT)

$$a(x) = \sum_{i=0}^{n-1} a_i x^i \quad , \quad b(x) = \sum_{i=0}^{n-1} b_i x^i$$

$$c(x) = \sum_{i=0}^{2n-2} c_i x^i = ab$$

Algoritmo standard lo interpreto come un prodotto matrice vettore

$$\begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_{2n-2} \end{bmatrix} = \begin{pmatrix} a_0 & & & \\ a_1 & a_0 & & \\ \vdots & & \ddots & \\ a_{n-1} & & & a_0 \\ & \ddots & & a_1 \\ & & \ddots & \vdots \\ & & & a_{n-1} \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{pmatrix}$$

$n = 2^p \qquad a(x) = \sum_{i=0}^{2^p - 1} a_i x^i$ (+ slide)

$2^{p-1} < 2n \leq 2^p$

$\lceil \log_2 n \rceil = p \quad , \quad a_n = \dots = a_{2^p - 1} = 0$

Prodotto tra polinomi tramite FFT

Sia p t.c $2^p \geq 2n-2$ e $2^{p-1} < 2n-2$

Scegliamo i n. complessi $\alpha_i = a(\omega_m^i) \quad i = 0, \dots, m-1$

con $m = 2^p$ $\beta_i = b(\omega_m^i) \quad i = 0, \dots, m-1$

Si calcola poi $\gamma_i = \alpha_i \beta_i \quad i = 0, \dots, m-1$

↑ prodotto di n complessi

Si trova p polinomio t.c. $p(\omega_n^i) = \gamma_i \quad i = 0, \dots, m-1$

$\Rightarrow p \quad p = a \cdot b$

p è il polinomio di interpolazione sulle radici dell'unità e vale γ_i, ma

$ab(\omega_n^i) = a(\omega_n^i) b(\omega_n^i) = \alpha_i \beta_i = \gamma_i$

il grado di ab rimane o uguale a m

$\Rightarrow ab = p$ poiché il polinomio di interpolazione è unico

Perciò:

- 2 IDFT
- n prodotti di numeri complessi $\rightarrow m$ ops
- DFT

$\frac{3}{2} m \log_2 m$ ops

$\Rightarrow O(m \log_2 m)$

⋮

Qual è il max n di cifre in base B della mantissa di un coefficiente di $\tilde{c}$? $(\leq (B-1)^2 N)$

$\Rightarrow \log_B \left((B-1)^2 N \right) =$

$$= \log_B N + 2\log_B (B-1) = c \log_2 N + \gamma = \Theta(\log_2 N)$$

$\vdots$

Ho due modi per vedere il fattoriale

$$n! \;\rightarrow\; n(n-1)! \qquad\qquad n! \;\rightarrow\; \prod_{i=1}^{n} i$$

$$F_n x = \begin{bmatrix} F_{\frac{n}{2}} x_p + D_{\frac{n}{2}} F_{\frac{n}{2}} x_D \\ F_{\frac{n}{2}} x_p - D_{\frac{n}{2}} F_{\frac{n}{2}} x_D \end{bmatrix} = \begin{bmatrix} F_{\frac{n}{2}} & D_{\frac{n}{2}} F_{\frac{n}{2}} \\ F_{\frac{n}{2}} & -D_{\frac{n}{2}} F_{\frac{n}{2}} \end{bmatrix} \begin{bmatrix} x_p \\ x_D \end{bmatrix} =$$

$$= \begin{bmatrix} I & D_{\frac{n}{2}} \\ I & -D_{\frac{n}{2}} \end{bmatrix} \begin{bmatrix} F_{n/2} & 0 \\ F_{n/2} & 0 \end{bmatrix} \begin{bmatrix} x_p \\ x_D \end{bmatrix} =$$

$$= \begin{bmatrix} I & D_{n/2} \\ I & -D_{n/2} \end{bmatrix} \begin{bmatrix} I & D_{\frac{n}{2}} & & 0 \\ I & -D_{n/2} & & \\ & & I & D_{\frac{n}{2}} \\ 0 & & I & -D_{\frac{n}{2}} \end{bmatrix} \begin{bmatrix} F_{n/4} & & & 0 \\ & F_{n/4} & & \\ & & F_{n/4} & \\ 0 & & & F_{n/4} \end{bmatrix} \underbrace{\begin{bmatrix} x_{i=0(4)} \\ x_{i=2(4)} \\ x_{i=1(4)} \\ x_{i=3(4)} \end{bmatrix}}_{\Pi_2 \Pi_1 x} =$$

$\Pi_2 \Pi_1 x$ permutazione vettore x

$$\begin{bmatrix} \begin{matrix} 1 & 1 \\ 1 & -1 \end{matrix} & & & 0 \\ & \begin{matrix} 1 & 1 \\ 1 & -1 \end{matrix} & & \\ & & \begin{matrix} 1 & 1 \\ 1 & -1 \end{matrix} & \\ 0 & & & \begin{matrix} 1 & 1 \\ 1 & -1 \end{matrix} \end{bmatrix} \leftarrow$$

$\rightarrow F_{n/4}$

→ I passi della FFT sono p

I blocchi sono 2^{p-l}

Dimensione dei blocchi è 2^l

Dimensione di D_q è 2^{p-l-1}

→ Implementazione con cicli for:

```
For l = 1:p          % Calcolo x = Δ_l x
    For j = 0: 2^(p-l)      % scorro i blocchi
        For q = 0: 2^(l-1)
            γ = ω^q * x(q + 2^(l-1))
            x(q + 2^(l-1)) = x(i) - γ
            x(q) = x(i) + γ
```

25/11

$Ax=b \quad A \in \mathbb{C}^{n\times n} \quad \det A \neq 0 \quad b \neq 0$

(A di grandi dimensioni e sparsa)

Metodi dei sottospazi di Krylov:

$k, L \subset \mathbb{C}^n$, $\dim k = \dim L = n$

$x_m = x_0 + k$ ← approssimazione della soluzione del sistema lineare

$b - Ax_n \perp L$

} metodo di proiezione

1. $k = L = k_n(A, r_0)$ FOM (full orthogonalization method)

$r(x) = b - Ax$

$r_0 = b - Ax_0$

$r_n = b - Ax_n$

CG (coniugate gradient se A è definita positiva)

$k_n(A, v) = \text{span}\{v, Av, A^2v, \dots, A^{m-1}v\}$

2. $k = k_n(A, r_0)$, $L = AK$ GMRES (A invertibile)

3. $k = k_n(A, r_0)$, $k_n(A^T, b)$ BIGG-STAB

Se $v \neq 0$ e $d = n$ numero intero di numero primi in $\mathbb{C}^{n\times n}$ $\dim \mathbb{C}^n$ $d \leq n$ $m \geq d$

$k_1(A, v) \subsetneq k_2(A, v) \subsetneq \dots \subsetneq k_d(A, v) = \dots = k_m(A, v)$

dove d è il grado di v rispetto ad A, grado minimo di un polinomio tc $p(A)v = 0$

$$K_m(A,v) = \{ p(A)v \,,\ \deg p < m \}$$

$$A^{-1} = p(A) \quad , \quad x_* = A^{-1}b = p(A)b \in K_m(A,b) \quad \left(\begin{matrix} \text{per } m \text{ grande} \\ m \geq d \end{matrix} \right)$$

$$x_* = A^{-1}b = q(A)b \quad \text{con } \deg q << n$$

La sol di 1. si può scrivere esplicitamente come:

$$V_m \in \mathbb{C}^{m \times n} \,, \quad \text{Span } V_m = K_m$$

$$K_m = K_m(A, r_0) \,, \quad \text{span } M = \text{span}\{\text{colonne}\}$$

$$\boxed{x_n = x_0 + V_m \left(V_m^* A V_m \right)^{-1} V_m^* r_0}$$

Le cose da fare ora sono

- trovare una base
- è vero che $V_m^* A V_m$ è invertibile?
- è necessario invertire $V_m^* A V_m$?
- Se m è piccolo, quanto è buona l'approssimazione?
- come scegliere m?

Se $m \leq d$ $v, Av, \dots, A^{m-1}v$ è una base, questa base però è mal condizionata ($\|M\|_2 \|M^{-1}\|_2$ grande)
Infatti se ad esempio A è diagonalizzabile allora sono

$$A = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix}$$

, se c'è un autovalore più grande

degli altri quando lo moltiplico molte volte per v il prodotto tenderà ad allinearsi al prodotto dell'autovalore massimo con le vettore

⇒ Ci serve un'altra base (ben condizionata cioè con $\mu_2(M) = \|M\|_2 \|M^{-1}\|_2$ piccolo)

⇒ Ci servono le matrici unitarie per cui sappiamo che $\mu_2(M) = 1$ se $M \in U(n)$

Trovare una base ortogonale dello spazio $K_n(A, v)$

Usiamo allora i metodi di Arnoldi: "come Gram-Schmidt ma sui poli"

(Prendere A e v e costruire $v_1, \dots, v_m$ base ortog di $K_n(A, v)$)

(0) $$v_1 = \frac{v}{\|v\|}$$

(1) $$A v_1 = u_1 \quad , \quad \tilde{u}_1 = u_1 - \underbrace{\langle u_1, v_1 \rangle}_{h_{11}} v_1 \quad (\tilde{u}_1 \perp v_1)$$

$$v_2 = \frac{\tilde{u}_1}{\|\tilde{u}_1\|} \quad (\text{se } \tilde{u}_1 \neq 0) \qquad \|\tilde{u}_1\| = h_{21}$$

(2) $$A v_2 = u_2 \quad , \quad \tilde{u}_2 = u_2 - \underbrace{\langle u_2, v_1 \rangle}_{h_{12}} v_1 - \underbrace{\langle u_2, v_2 \rangle}_{h_{22}} v_2$$

$$v_3 = \frac{\tilde{u}_2}{\|\tilde{u}_2\|} \qquad \|\tilde{u}_2\| = h_{32}$$

.
.
.

(1) $Av_m = u_m$, $\tilde{u}_m = u_m - \sum_{j \le m} \overbrace{\langle u_m, u_j \rangle}^{h_{j,m}} v_j$

$v_{m+1} = \dfrac{\tilde{u}_m}{\|\tilde{u}_m\|}$ ortonormalizzazione

$\|\tilde{u}_m\| = h_{m+1,m}$

prodotto matrice vettore

$\sim O(mn)$ flops $+ m$

La base che ho ottenuto è base di K_n?

Teorema:
Se il metodo non si arresta fino al passo l incluso allora $\text{span}\{v_1, \dots, v_i\} = K_i(A, v)$, $i \le l+1$
Se il metodo si arresta al passo l allora
$\text{span}\{v_1, \dots, v_{l-1}\} = K_m(A, v) = K_l(A, v)$ $m \ge l$ e
$A^{-1} v \in K_l(A, v)$ (LUCKY BREAK DOWN)

↑ perché nel caso di arresto va tutto bene (è la situazione migliore)

Teorema:
se il metodo di Arnoldi per $K_{m+1}(A, v)$ non si arresta fino al passo m e sia
$V_l = [v_1 | \dots | v_l] \in \mathbb{C}^{l \times n}$ dove $v_1, \dots, v_{m+1}$ sono i vettori ottenuti dal metodo

$$\bar{H}_m = \begin{bmatrix} h_{11} & \cdots & h_{1m} \\ h_{21} & \ddots & \vdots \\ & \ddots & \vdots \\ & & h_{m+1,m} \end{bmatrix} \in \mathbb{C}^{(m+1) \times m}$$

dove gli $h_{i,j}$ sono i coeff del metodo di Arnoldi

$H_m \in \mathbb{C}^{m \times m}$ la matrice ottenuta selezionando le prime righe di $\overline{H}_m$ allora valgono

1. $AV_m = V_{m+1} \overline{H}_m$

2. $AV_m = V_m H_m + h_{m+1,m} v_{m+1} e_m^T$

3. $V_m^* A V_m = H_m$

Dimostrazione

(1) $1 \le l \le m$

$$Av_l = u_l = \tilde{u}_l + \sum_{j \le l} h_{jl} v_j = h_{l+1,l} v_{l+1} + \sum_{j \le l} h_{jl} v_j =$$

$$= \sum_{j \le l+1} h_{jl} v_j = \begin{bmatrix} v_1 | \dots | v_{m+1} \end{bmatrix} \begin{bmatrix} h_{1l} \\ \vdots \\ h_{l+1,l} \\ 0 \\ \vdots \\ 0 \end{bmatrix} =$$

$$= V_m \begin{bmatrix} h_{1l} \\ \vdots \\ h_{l+1,l} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$AV_m = A \begin{bmatrix} v_1 | v_2 | \dots | v_m \end{bmatrix} = \begin{bmatrix} Av_1 | Av_2 | \dots | Av_m \end{bmatrix} =$$

ove

$$Av_1 = V_{m+1} \begin{bmatrix} h_{11} \\ h_{21} \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad Av_2 = V_{m+1} \begin{bmatrix} h_{12} \\ h_{22} \\ h_{32} \\ ? \\ 0 \end{bmatrix} \dots$$

$$\Rightarrow AV_m = \left[\begin{array}{c|ccc} h_{11} & h_{12} & \dots & h_{m1} \\ h_{21} & h_{22} & \ddots & \vdots \\ 0 & h_{32} & \ddots & \vdots \\ \vdots & 0 & \ddots & h_{mm} \\ 0 & 0 & & h_{m+1,m} \end{array}\right] = V_{m+1} \overline{H}_m$$

(2)

$$\left[V_m \mid v_{m+1} \right] \begin{bmatrix} H_m \\ 0 \cdots 0 \; h_{m+1,m} \end{bmatrix} =$$

V_m: $m \times m$ — v_{m+1}: $(m+1) \times 1$ — H_m: $m \times m$ — $h_{m+1,m}$: $1 \times (m+1)$

$$= V_m H_m + v_{m+1} \left[0 \cdots 0 \; h_{m+1,m} \right] = V_m H_m + h_{m+1,m} v_{m+1} f_m^T$$

(3)

$$V_m^* A V_m = \underbrace{V_m^* V_m}_{= I \text{ perché sono } \perp} H_m + \underbrace{V_m^* h_{m+1,m} v_{m+1} f_m^T}_{h_{m+1,m} \underbrace{V_m^* v_{m+1}}_{=0} f_m^T} = H_m$$

Vediamo i metodi:

FOM (se A è def positiva ho CG)

$$K = L = K_m(A, r_0)$$

$$\begin{cases} x_m = x_0 + V_m y_m \\ y_m = (V_m^* A V_m)^{-1} V_m^* r_0 \end{cases}$$

se del problema $\sim$ $\begin{cases} x_1 \in x_0 + K \\ b - A x_0 \perp L \end{cases}$

Se V_m è la matrice le cui colonne sono la base ottenuta tramite il metodo di Arnoldi

$$- V_m^* A V_m = H_m$$

$v_1 = \frac{r_0}{\|r_0\|_2}$ perché sono in $K_m(A, r_0)$

$$- V_m^* r_0 = \underbrace{V_m^* v_1 \|r_0\|_2}_{(1 0 \cdots 0)^T} = \|r_0\|_2 f_1$$

⇒ Prima si calcola V_m con Arnoldi, poi si risolve il sist lineare $H_m y_m = \|r_0\|_2 f_1$ (↑ $m \ll n$), poi l'approssimazione è $x_m = x_0 + V_m y_m$

Come lo scelgo però questo m?

Se A è def positiva ⇒ H_m è simmetrica ⇒ è tridiagonale (⇒ si accelera Arnoldi e si ottiene Lanczos)

$$\|x_m - x_*\|_A = \min_{x_n \in x_0 + K} \|x - x_m\|_A$$

① Posso stimare a priori l'errore

② Oppure lo trasformo in un metodo iterativo con stima a posteriori dell'errore

Teorema (per FOM)

$$\|b - Ax_m\|_2 = h_{m+1,m} |f_m^T y_m| \quad (\text{succ } \forall \text{ in } m)$$

Dim

$$b - Ax_m = b - Ax_0 - AV_m y_m = r_0 - AV_m y_m =$$

$$= r_0 - V_m H_m y_m - v_m h_{m+1,m} v_{m+1} f_m^T y_m =$$

$$= r_0 - V_m \|r_0\|_2 f_1 - V_m v_{m+1} f_m^T h_{m+1,m} y_m =$$

$$= \cancel{r_0} - \underbrace{V_1 \|r_0\|_2}_{r_0} - V_m v_{m+1} f_m^T h_{m+1,m} y_m =$$

$$\Rightarrow \|b - Ax_m\|_2 = \|y_m v_{m+1} f_m^T h_{m+1,m}\| = h_{m+1,m} |f_m^T y_m|$$

Teorema: ✱

Se A def positiva $(x_* = \bar{A}b, b \neq 0)$, $m = 0, 1, \dots, d$ d grado di (A,b)

$$\|x_m - x_*\|_A \leq 2 \left(\frac{\sqrt{\mu} - 1}{\sqrt{\mu} + 1} \right)^n \|x_m - x_0\|_A$$

con $\mu = n$ di condizionamento, $\mu = \|A\|_2 \|A^{-1}\|_2$

26/11

Costo computazionale del metodo di Arnoldi

$$\tilde{u}_i = u_i - \sum_{j \leq i} \langle u_i, v_j \rangle v_j \Rightarrow 4n$$

(u_i: n; $\langle u_i, v_j \rangle$: $2n$; v_j: n)

$$\frac{\tilde{u}_i}{\|\tilde{u}_i\|} \rightarrow 2n + 1 \text{ sqrt ops}$$

$$\Rightarrow \sum_{i=1}^{m} \sum_{j \leq i} 4n \sim \frac{4nm^2}{2} \sim 2nm^2$$

$\Rightarrow$ Arnoldi "costa" $2nm^2$ ops se A è sparsa

Per risolvere il sistema lineare posso avere $\frac{2}{3}n^3$ ops (GAUSS)

Per il passo 3 ($x_1 = x_0 + v_m y_m$) ho $2nm$

Si implementa così:

For $i = 1, 2, \dots$

% calcolo v_i con Arnoldi e calcolo y_i, τ_i

if $(\|b - Ax\|_m < tol)$

stop

else

Se A definita positiva FOM si chiama CG (gradiente coniugato) e si semplifica di molto

$$H_m = V_m^* A V_m, \quad H_m^* = V_m^* A^* (V_m^*)^* = V_m^* A V_m = H_m$$

$\Rightarrow H_m$ hermitiana + hessenberg $\Rightarrow$ tridiagonale

metodo di Lanczos

$\tilde{u}_i = u_i - \langle u_i, v_1 \rangle v_i - \langle u_i, v_{i-1} \rangle v_{i-1}$

↑ Il metodo di Arnoldi si semplifica di molto

$$x_{m+1} = x_m + \alpha_m d_m$$

Gli autovalori di H_m, detti valori di RITZ, approssimano gli autovalori di A

2. $\begin{cases} x_m = x_0 + k_m \\ b - Ax_m \perp Ak_m \end{cases}$ A invertibile

Metodo GMRES (generalized minimal residual)

Si dimostra che $b - Ax_m \perp AK_m$ equivale a dire $\|b - Ax_m\|_2 = \min_{x \in x_0 + K_m} \|b - Ax\|_2$

Calcolo V_m con Arnoldi senza breakdown

$$x_m = x_0 + V_m y_m \qquad r_0 = v_1 \|r_0\| = \|r_0\| V_{m+1} f_1 \Leftarrow v_1 = \frac{r_0}{\|r_0\|}$$

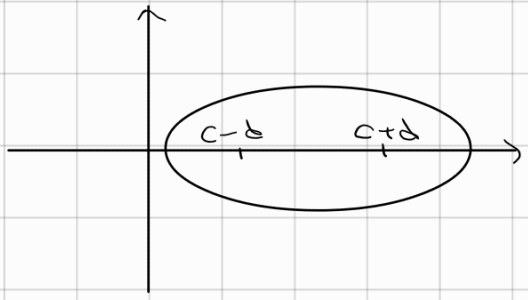
$$b - Ax_m = \underbrace{b - Ax_0}_{r_0} - AV_m y_m = r_0 - V_{m+1} \bar{H}_{m+1} y_m =$$

$$= V_{m+1} \left(\|r_0\| f_1 - \bar{H}_{m+1} y_m \right)$$

$$\|b - Ax_m\|_2 = \| V_{m+1} (\|r_0\| f_1 - \bar{H}_m y_m) \|_2$$

x_m che minimizza il residuo è del tipo $x_0 + V_m y_m$ dove y_m minimizza $\| \|r_0\| f_1 - \bar{H}_m y_m \|_2$

$\Rightarrow$ y_m è la soluzione del problema dei minimi quadrati con dati $\bar{H}_m$ e $\|r_0\| f_1$



Se gli autovalori di A si trovano nell'ellisse di centro c, fuochi c-d e c+d e semiasse maggiore a

e A è diagonalizzabile $X^{-1} A X = D = \begin{bmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{bmatrix}$

$$\|r_m\|_2 \leq \mu_2(X) \frac{T_m\left(\frac{a}{d}\right)}{|T_m\left(\frac{c}{d}\right)|} \|r_0\|$$

T_m pol di Chebyshev m-esimo complesso

Problemi di ottimizzazione

$f: \mathbb{R}^n \to \mathbb{R}$ continua

trovare $\min_{x \in \mathbb{R}^n} f(x)$ o $\arg\min_{x \in \mathbb{R}^n} f(x)$

} ottimizzazione non vincolata

$f: X \to \mathbb{R} \quad X \subsetneq \mathbb{R}^n$

$\min_{x \in X} f(x)$

} ottimizzazione vincolata

OSS

trovare min o max è indifferente poiché

$$\max_{x \in X} f = -\min_{x \in X} (-f)$$

e inoltre $\min_{x \in X} (f(x) + c) = \left(\min_{x \in X} f(x)\right) + c$

x^* min globale per f in X se $f(x) \geq f(x^*) \quad x \in X$

" locale " $f(x) \geq f(x^*)$ in $B(x^*, r)$

ed è stretto se vale la maggiorazione stretta.

L'ottimizzazione è divisa in:

- Analisi (∃ del minimo etc...)
- Metodi ottimizzazione

Esempi:

- $f(x,y) = x^2+y^2$ $\quad x^2+y^2 \geq 0$ $\quad x^2+y^2 = 0 \iff (x,y) = (0,0)$

$\Rightarrow \min_{(x,y)\in\mathbb{R}} f(x,y) = 0$ e $\operatorname{argmin}_{(x,y)\in\mathbb{R}} f(x,y) = \{(0,0)\}$

Non ha massimo

- $f(x,y) = -x^2-y^2$ ha massimo assoluto e non ha minimo

- $f(x,y) = x^2-y^2$ non ha né max né min globali (locali non lo so per ora)

- $$\begin{cases} x_0 \in \mathbb{R}^{n_0} \\ y_i = A_{i-1} x_{i-1} - b_{i-1}, \quad i = 1, \dots, l \\ x_i = \sigma_i(y_i) \end{cases}$$

reti neurali definite in modo ricorsivo

dove $A_i \in \mathbb{R}^{n_i \times n_{i-1}}$, $b_i \in \mathbb{R}^{n_i}$ (pesi)

$\sigma : \mathbb{R}^{n_i} \longrightarrow \mathbb{R}^{n_i}$, $(\sigma_i(x_i))_i = \varphi_i(x_i)$, $\varphi_i : \mathbb{R} \longrightarrow \mathbb{R}$

funzione di attivazione

(deve essere non lineare)

$$\begin{array}{ccc} \mathbb{R}^{n_0} & \xrightarrow{f} & \mathbb{R}^{n_l} \\ x_0 & \longrightarrow & x_l \end{array}$$

Ottimizzazione non vincolata:

$f \in C^1(\mathbb{R}^n)$, $x_0 \in \mathbb{R}^n$ esiste $L(x_0): \mathbb{R}^n \to \mathbb{R}$

$$f(x_0+h) = f(x) + L(x_0)[h] + o(h) \qquad h \in \mathbb{R}^n,\ h \to 0$$

$L(x_0)$ se esiste si chiama derivata di Frechet in x_0 ($Df(x_0)$)

Se $Df(x_0)$ esiste $\forall x_0 \in \mathbb{R}^n$ allora f è derivabile in $\mathbb{R}^n$

$$Df: \mathbb{R}^n \longrightarrow \mathrm{Hom}(\mathbb{R}^n, \mathbb{R})$$
$$x_0 \longmapsto Df(x_0)$$

$$\lim_{h \to 0} \frac{|f(x_0+h) - f(x_0) - L(x_0+h)|}{\|h\|} = 0$$

Proposizione

Per ogni $g: \mathbb{R}^n \to \mathbb{R}$ $\exists\, v \in \mathbb{R}^n$ t.c. $g(w) = \langle v, w \rangle$ $\forall w \in \mathbb{R}^n$

oss

f derivabile in x_0 $f: \mathbb{R}^n \to \mathbb{R}$ $Df(x_0): \mathbb{R}^n \to \mathbb{R}$

$$Df(x_0)[h] = \langle \nabla f(x_0), h \rangle \quad , \nabla f(x_0) \in \mathbb{R}^n$$

(gradiente)

$A \in \mathbb{R}^{1 \times n}$ associato a $Df(x_0)$

$$A = \left[\frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right] = \nabla f(x_0)^T$$

$Df(x_0)$ è detta derivata nella direzione h

$\|$

$$\nabla f(x_0)^T[h] = \lim \frac{f(x_0+th) - f(x_0)}{t} = \frac{\partial f}{\partial h}$$

Teorema (cond necessario del I ordine per l'ott vincolata)

Se $f \in C^1(\mathbb{R}^n)$ e x^* è minimo $\Rightarrow \nabla f(x^*) = 0$

Dim:

Se $f(x) \geq f(x^*)$ in $\in B(x^*, r)$

$\Rightarrow f(x^*+th) \geq f(x^*)$

$\Rightarrow$ per ogni $h \in \mathbb{R}^n$ $x^* + th \in B(x^*, r)$, $t \in [0, t_h]$

$$\lim_{t \to 0} \frac{f(x^*+th) - f(x^*)}{t} = \nabla f(x^*)^T h \geq 0$$

ma è anche ≤ 0

$\Rightarrow \nabla f(x^*)^T h = 0 \quad \forall h \in \mathbb{R}^n \Rightarrow \langle \nabla f(x^*), h \rangle = 0$

$\Rightarrow \nabla f(x^*) = 0$

$f(x) \in C^2(\mathbb{R})$

02/12
03/12

$\nabla f(x^*) = 0 \quad \nabla^2 f(x) \geq 0 \Leftarrow x^*$ minimo

Teorema:

$\nabla f(x^*) = 0 \;,\; \nabla^2 f(x) > 0 \Rightarrow x^*$ è minimo

Dim:
$h \in S^{n-1}$

$$f(x^* + th) - f(x^*) = \nabla f(x^*)[th] + \frac{1}{2} D^2 f(x^*)[th, th] + o(\|h\|^2)$$

$$\lim_{t \to 0} \frac{f(x^* + th) - f(x^*)}{t^2} = \frac{1}{2} h^2 \nabla^2 f(x^*) h > \frac{1}{2} \varepsilon \|h\|^2 = \frac{1}{2} \varepsilon$$

$\exists\, t_0(h)$ t.c. $t \in]0, t_0[h]]$, $f(x^* + th) > f(x^*)$

$(t, h) \longrightarrow f(x^* + th) - f(x^*)$ continua esiste $\mathcal{U}(h)$ intorno di h in S^{n-1}

$(t, h) \in [0, t_s] \times \mathcal{U}(h)$, $t_s > 0$

$$S^{n-1} = \bigcup_{h \in S^{n-1}} \mathcal{U}(h) = \bigcup_{s=1}^{n} \overline{\mathcal{U}(h_s)}^{\,t_s}$$

$h^T A h > \varepsilon h^T h = \varepsilon \|h\|^2$

$t \in \left]0, \min_{s=1,\dots,n} t_s\right[\Rightarrow f(x^* + th) > f(x^*)$, $B(x^*, \min t_s) \setminus \{x^*\}$

Esempio:

$A \in \mathbb{R}^{n\times n}$ simmetrica, $b \in \mathbb{R}^n$

Consideriamo $\varphi(x) = \frac{1}{2} x^T A x - b^T x$, $\nabla\varphi(x) = Ax - b$

$$\varphi(x+h) = \frac{1}{2}(x+h)^T A(x+h) - b^T(x+h) - \frac{1}{2}x^TAx - b^Tx =$$

$$= (x^TA - b^T)h + \frac{1}{2}h^TAh$$

I punti stazionari sono i punti in cui il gradiente è 0

I punti stazionari sono le soluzioni di $Ax - b = 0$

Se $A \in GL(n) \Rightarrow \exists!$ punto stazionario

Se $A > 0 \Rightarrow \exists$ minimo globale

Se A ha un autovalore negativo $\Rightarrow A \neq 0 \Rightarrow$ non è un minimo

$$\varphi(x) = \|Ax-b\|_2^2 = (Ax-b)^T(Ax-b) = x^TA^TAx - 2x^TA^Tb - b^Tb$$

$$2A^TAx - 2A^Tb = 0$$

Proprietà del gradiente:

$f: \mathbb{R}^n \to \mathbb{R}$ allora $f(x) = b^Tx$, $b \in \mathbb{R}^n$

Se $b = 0$ $\quad f(x) \equiv 0$

Se $b \neq 0$ $\quad f(x) = 0 \Leftrightarrow x \perp b \Rightarrow \ker f = \{b^\perp\} = \text{span}\{b\}^\perp$ ha dim $n-1$

$\ker f$ divide $\mathbb{R}^n$ in due parti in cui $f(x)$ ha segno opposto

$f \in C^1(\mathbb{R})$ nell'intorno di x_0 si comporta come

$$f(x) - f(x_0) \approx f(x_0) + \nabla f(x_0)^T \underbrace{(x - x_0)}_{h}, \quad x \in M(x_0)$$

Teorema:

Sia $f \in C^1(\mathbb{R}^n)$ e sia $x_0 \in \mathbb{R}^n$ t.c $\nabla f(x_0) \neq 0$ allora se $h \in \mathbb{R}^n$ t.c $\nabla f(x_0)^T h < 0$, esiste $t_0 > 0$ t.c se $t \in [0, t_0]$ $f(x_0 + th) < f(x_0)$

Dim:

$$f(x_0 + th) - f(x_0) = \underbrace{\langle \nabla f(x_0), th \rangle}_{\langle \cdot \rangle = t \nabla f(x_0)^T h} + o(t) \quad \text{poiché } f \in C^1(\mathbb{R}^n)$$

$$\lim_{t \to 0} \frac{f(x_0 + th) - f(x_0)}{t} = \nabla f(x_0)^T h < 0 \Rightarrow \exists t_0 > 0 \text{ t.c}$$

per $t \in [0, t_0]$,
$f(x_0 + th) < f(x_0)$

Metodi di ottimizzazione:

- Valore iniziale $x_0 \in \mathbb{R}^n$
- iterazione: scelgo la direzione $d_k \in \mathbb{R}^n$ $k = 0, 1, \ldots$
  e passo $\alpha_k \in \mathbb{R}^+$

Se $\nabla f(x_k) = 0 \Rightarrow$ mi fermo

Se $\nabla f(x_k) \neq 0$:

$$x_{k+1} = x_k + \alpha_k d_k$$

Da questo metodo non spero che converga a un minimo o a un massimo ($\nabla f(x_k)=0$ potrebbe essere di sella)
Spero solo che converga a un punto stazionario, anzi che sia una sottosuccessione a convergere a un punto stazionario.

Scegliamo d_k tale che $\nabla f(x_k)^T d_k < 0$
↑ direzione di discesa

Qual è la direzione di massima discesa?
- Una scelta potrebbe essere $-\nabla f(x_k)$
- Un'altra potrebbe essere $-A_k \nabla f(x_k)$ con A_k definita positiva ($-\nabla f(x_k)^T \underbrace{A \nabla f(x_k)}_{d_k} < 0$)

(Questi metodi si chiamano metodi del gradiente)

Metodo di Newton:

I CASO

$\nabla f(x) = 0$ → Sistema non lineare

$x \in \mathbb{R}^n$ incognite
$\nabla f(x) \in \mathbb{R}^n$ equazioni

$x_{k+1} = x_k - \frac{\hat{f}(x_k)}{\hat{f}'(x_k)}$ $\quad \hat{f}: \mathbb{R}^n \to \mathbb{R}^n$, $\hat{f}(x)=0$

$x_{k+1} = x_k - D\hat{f}(x_k)^{-1}[\hat{f}(x_k)] =$

$= x_k - \bar{J}^{-1}(x_k)\hat{f}(x_k)$
↑ Jacobiana

II CASO

Se $\hat{f} = \nabla f$, $x_{k+1} = x_k - H(x_k)^{-1}\nabla f(x_k)$ metodo di Newton

oppure considero $x_{k+1} = x_k + \alpha_k \underbrace{(-H(x_k)^{-1}\nabla f(x_k))}_{d_k}$

$H(x_k)$ def pos $\Rightarrow$ $H(x_k)^{-1}$ è def pos

Metodi che usano solo derivata prime sono del I ordine
Metodi che usano anche derivata seconda sono del II ordine

Fissato d_k, cerchiamo α_k

Siamo nel caso in cui non so dov'è il minimo

restrizione di f alla retta $x_k + \alpha d_k$, $\alpha \in \mathbb{R}$

$\varphi(\alpha) = f(x_k + \alpha d_k)$, cerco $\operatorname{argmin}_{\alpha \in \mathbb{R}} f(x_k + \alpha d_k) \Rightarrow \alpha_k$

exact line search

e in base di questo α_k

metodo del gradiente + exact line search $\Rightarrow$ steepest descent

Idea alternativa: inexact line search
cioè scelgo α_k in modo che garantisca che sia facile da trovare e che garantisca "decrescita sufficiente" per la convergenza globale.
Un esempio è:

- Regola di ARMIJO (backtracking)
  Ad ogni passo inizio (ad esempio $\bar{\alpha} = 1$), $\beta \in]0,1[$ poniamo che mi dice di quanto mi muovo

  $\sigma \in]0,1[$

Al passo k si sceglie il più piccolo intero non negativo m_k tale che

La decrescita a volte non garantisce la convergenza perciò

$$f(x_k + \bar{\alpha}\beta^{m_k} d_k) < f(x_k) + \sigma \bar{\alpha}\beta^{m_k} \nabla f(x_k)^T d_k$$

$\nabla f(x_k)^T d_k < 0$

$\frac{1}{2} x^T A x^T - x^T b$

- Calcolare il gradiente (simbolicamente)
  - differenziazione numerica
  - differenziazione automatica

- Valutare il gradiente
  (Nelle reti neurali si fa e calcolare e valutare)

$$\begin{cases} x_0 \\ y_i = A_i x_{i-1} + b_i \quad , \quad i = 1, \dots, l \\ x_i = \sigma(y_i) \end{cases}$$

(lo σ non ha pesi)

$$x_l = F(x_0; \overbrace{A_1, A_2, \dots, A_l, b_1, \dots, b_l}^{\text{pesi}})$$

$\tilde{x}_i, \tilde{y}_i$ $\quad i = 1, \dots, N$

LOSS FUNCTION

$$\mathcal{L} = \sum_{i=1}^{N} \| f(\tilde{x}_i, A_1, \dots A_l, b_1, \dots b_l) - \tilde{y}_i \|^2$$

Trovare $A_1, \dots A_l, b_1, \dots b_l$ tale che $\mathcal{L}$ sia minima

$$f_1(x) = \|x\|^2 = x^T x \,, \quad D f_1(x)[h] = 2 x^T h$$

$$f_2(x) = \sigma(x) \,, \quad f_2(x+h) - f_2(x) = \begin{bmatrix} \sigma(x_1 + h_1) - \sigma(x_1) \\ \sigma(x_2 + h_2) - \sigma(x_2) \\ \vdots \\ 1 \end{bmatrix} =$$

$$= \begin{bmatrix} \sigma'(x_1) h \\ \vdots \\ \sigma'(x_n) h \end{bmatrix} + o(\|h\|)$$

prodotto punto per punto

$\Rightarrow D f_2(x)[h] = D\sigma(x)[h] = \sigma'(x) \odot h$

$f_3(A) = Ax + b$, $Df_3(A)[H] = Hx$

$\mathbb{R}^{n \times n} \to \mathbb{R}^n$

$f_4(b) = Ax + b \qquad Df_4(b)[h] = h$

$\Rightarrow f(A,b) = Ax - b$, $Df(A,b)[H,h] = Hx + h =$

$$= \begin{bmatrix} x^T \otimes I & I \end{bmatrix} \begin{bmatrix} \text{vec}(H) \\ h \end{bmatrix}$$

09/12

$$D(f \circ g)(x)[h] = Df(g(x))\left[Dg(x)[h]\right]$$

Quindi se $f \in C^1 \Rightarrow \varphi'(\alpha) = Df(x+\alpha d)[d] = \nabla f(x+\alpha d)^T d$

$\Rightarrow \varphi'(0) = \nabla f(x)^T d$

$\|d\| = 1$, $\nabla f(x)^T d = (\cos\theta)\,\|\nabla f(x)\|$ (θ è max per $\theta = 0$, è min per $\theta = \pi$)

NOTAZIONE:

$d_k = -\nabla f(x_k) = -g(x_k)$

$d_k = A_k g(x_k)$, A_k def positiva

$d_k = -g(\hat{x}_k) + \beta_k g(x_{k-1})$ ← in questa categoria c'è il metodo del GC

(metodo simile è il momentum)

Si dimostra che applicando CG alla funzione quadratica $p(x) = \frac{1}{2} x^T A x - x^T b$, $A > 0$, allora ottengo le FOM.

Posso scegliere α_k attraverso:

- <u>PASSO FISSO</u>: quando $\beta I \geq H(x) \geq \alpha I$ ci sono dei passi fissi che garantiscono la convergenza globale

- ARMIJO'S RULE:

$\bar{\alpha}, \beta \in]0,1[$, $\sigma \in]0,1[$
ad ogni passo $\alpha_k = \bar{\alpha} \beta^{m_k}$ dove m_k è il più piccolo intero non negativo tale che

$$f(x_k + \alpha_k d_k) \leq f(x_k) + \sigma g(x_k), \quad d_k > \alpha_k$$
(sufficiente decrescita)

Lemma:

Sia $\gamma : I \to \mathbb{R}$, I intorno di 0 in $\mathbb{R}$, tale che
$\gamma(t) = \gamma_0 + \psi(t)$ con $\lim_{t \to 0} \psi(t) = 0$, allora per ogni

$0 < \gamma < 1$ esiste $t_0 > 0$ t.c $t \in [0, t_0] \Rightarrow \gamma(t) > \sigma \gamma_0$ Se $\gamma_0 > 0$
[$<$] [se $\gamma_0 < 0$]

Teorema:

Data $f \in C^1(\mathbb{R}^n)$, $x \in \mathbb{R}^n$ e $d \in \mathbb{R}^n$ direzione di decrescita per f in x, $0 < \sigma < 1$, esiste $t_0 > 0$ t.c $t \in [0, t_0]$

$$f(x + td) \leq f(x) + \sigma \nabla f(x)^T d t \quad (\nabla f(x) \neq 0)$$

Dim:

$$f(x+td) = f(x) + t\nabla f(x)^T d + \overbrace{\psi(t)t}^{\text{resto dove } \psi(t)\to 0}$$

per Lemma $\underbrace{\nabla f(x)^T d}_{<0} + \psi(t) \leq \sigma \nabla f(x)^T d$ per $t \in [0, t_0]$

Teorema:

Sia $\{x_k\}_k$ una successione infinita ottenuta dal metodo ($\nabla f(x)$ non è mai zero)
$x_0 \in \mathbb{R}^n$, $x_{k+1} = x_k + \alpha_k d_k$ per ottimizzare $f \in C^1(\mathbb{R}^n)$, dove la scelta di α_k è ottenuta tramite la regola di Armijo con parametri $\bar{\alpha} > 0$, $\beta \in]0,1[$, $\sigma \in]0,1[$.
Se $\{d_k\}_k$ è gradient-related allora ogni punto limite è stazionario

Def
$\{d_k\}_k$ successione di direzioni è gradient-related se per ogni sottosuccessione di $\{x_k\}_k$ che converge a un punto non stazionario

$$\limsup_{k \in \mathcal{L}_1} \langle g(x_k), d_k \rangle < 0$$

$\{\|d_k\|\}_{k \in \mathcal{L}_1}$ è limitata, $\mathcal{L}_1 \subseteq \{0,1,\dots\}$ sottoinsieme infinito

Dim:
$\mathcal{L}_1 \subseteq \mathcal{L}_0 = \{0,1,2,\dots\}$
P.A
1) Supponiamo che esiste $\{x_k\}_{k \in \mathcal{L}_1}$ t.c $\lim_{k \in \mathcal{L}_1} x_k = x^*$ $g(x^*) \neq 0$
2) Dimostriamo che $f(x_{k+1}) - f(x) \to 0$
Siccome f è continua allora $\lim_{k \in \mathcal{L}_1} f(x_k) = f(\lim_{k \in \mathcal{L}_1} x_k) = f(x^*)$

$\{f(x_k)\}_{k \in \mathcal{L}_0}$

3)

4)

5) $\tilde{\alpha}_k = \frac{\alpha_k}{\beta}\|d_k\|$, $\tilde{d}_k = \frac{d_k}{\|d_k\|}$, $d_k \neq 0$ perché $\{\|d_k\|\}_{k\in\mathcal{L}_1}$ limitata ...
gradienti relativi
ed è una direzione di discesa

Esiste una sottosuccessione $\mathcal{L}_5$ che converge

6) $k \in \mathcal{L}_5$

$$f(x_k + \tilde{\alpha}_k \tilde{d}_k) - f(x_k) = Df(x_k + \theta_k \tilde{\alpha}_k \tilde{d}_k)(\tilde{d}_k) =$$

$\exists\, \theta_k \in]0,1[$

Teorema del valor medio

$$\langle g(x^k), d^k \rangle \geq \sigma \langle g(x^k), d^k \rangle$$

$$A(x_{k+1} - x_k) = g(x_{k+1}) - g(x_k)$$

n° incognite, n° equazioni $\Rightarrow$ sottodeterminato

Oggi soluzione da un metodo detto quasi-Newton

$$d_{k+1} = -A g_k$$

Reti neurali

$$\begin{cases} x_0 \in \mathbb{R}^n \\ y_k = A_k x_{k-1} + b_k \qquad k = 1, \dots, L \\ x_k = \sigma(y_k) \\ f(x_0) = y = x_L \end{cases}$$

$$x_0^{(1)}, \dots, x_0^{(N)}$$
$$y_0^{(1)}, \dots, y_0^{(n)}$$

$$\mathcal{L} = \sum_{i=1}^{n} \| y_0^{(i)} - f(x_0^{(i)}, A_3, b_3, \dots A_1, b_1) \|^2 =$$

$$\mathcal{L}(A_3, b_3, \dots A_1, b_1) : \mathbb{R}^{n_0 \times n_3} \times \mathbb{R}^{n_3} \times \mathbb{R}^{n_3 \times n_2} \times \mathbb{R}^{n_2} \times \dots \longrightarrow \mathbb{R}$$

$$x_3 = \sigma(y_3) = \sigma(A_3 x_2 + b_3) = \sigma(A_3 \sigma(y_2) + b_3) =$$

$$= \sigma(A_3 + \sigma(A_2 x_1 + b_2) + b_3) = \sigma(A_3 + \sigma(A_2 \sigma(y_1) + b_2) + b_3) =$$

$$= \sigma(A_3 + \sigma(A + \sigma(A_1 x_0 + b_1) + b_2) + b_3)$$

$$DF(H, h) = D\sigma(y_3)[Hx_2 + h] = \sigma'(y_3) \odot [x_2^T \otimes I \; I] \begin{bmatrix} \mathrm{vec}(H) \\ h \end{bmatrix}$$

back propagation

costo computazionale : n di ops (non costa + della valutazione delle reti neurali)