Algoritmi avanzati
A.A. 2012-2013

Roberto Battiti
Paolo Campigotto

AVVERTENZA: lucidi da usare come ausilio mnemonico e lista degli argomenti svolti a lezione.

Non sostituiscono in alcun modo il libro di testo che va usato per lo studio approfondito.

Orari/Libro/Esami

Martedi' 8:30 – 10:30, aula A207

Giovedi' 10:30 – 12:30 (UPDATED!), aula A207

Libro

Introduction to algorithms

(ultima edizione se acquistato nuovo)

Cormen Leiserson Rivest Stein

MIT Press 2001

Esame

Scritto su tutto il programma (almeno 18)

Orale facoltativo

Due provette (50% peso, se favorevole)

http://www.intelligent-optimization.org/AA

Email: campigotto@disi.unitn.it, campo "oggetto": AA2012

Algoritmi Avanzati, perche'?

- Basis of advanced applications (see. Google, social networks, web)
- Abstraction / unification is power
- Define precise models of complex processes, use them to build solutions and reason about them

The sciences do not try to explain, they hardly even try to interpret,
they mainly make models.
By a model is meant a mathematical construct which,
with the addition of certain verbal interpretations,
describes observed phenomena.
The justification of such a mathematical construct
is solely and precisely that it is expected *to work*.

(Johann Von Neumann)

Topics

- **Matrix operations - Linear algebra**
- **Matrix operations - Eigenvalues and eigenvectors**
- **Continuous optimization: linear programming**
- **Approximation algorithms**
- **Continuous optimization: nonlinear programming**
- **Combinatorial optimization:**
  - **Local search and variations**
  - **Stochastic local search**
- **Reactive search optimization**
- **Interactive visualization**
- **Computational geometry**
- **String matching**

Operazioni con le matrici

testo pag. 725-733, 742-755,760-765

- Al cuore del calcolo scientifico
- Equazioni lineari → decomposizione LUP
- Calcolo della matrice inversa
- Approssimazione ai minimi quadrati

...da Matematica Discreta 1

- Spazio vettoriale
- Base
- Matrice ↔ trasformazione **lineare**
- Linearita'
- Autovalori, autovettori
- Diagonalizzabilita', polinomio caratteristico di una matrice
- Risoluzione sistemi lineari: $A\mathbf{x}=\mathbf{b}$
  - calcolo matrice inversa: A^{-1} $A\mathbf{x}=\mathbf{b}$; $\mathbf{x}=A^{-1}\mathbf{b}$
  - Costo computazionale!
  - Numericamente instabile!

Matrici

- Da ricordare
  - Moltiplicazione di matrici
  - Prodotto interno, prodotto esterno
  - Matrici inverse (nxn) e singolari
  - (In)dipendenza lineare *
  - Rango (rango pieno s.s.se non singolare)*
  - Vettore nullo per A: A $\mathbf{x} = \mathbf{0}$, $\mathbf{x}$ non nullo
  - nxn definita positiva $\mathbf{x}^T A \mathbf{x} > 0$, $\mathbf{x}$ diverso da $\mathbf{0}$
  - Se A a rango colonna pieno, $A^T A$ definita positiva
  - Matrice P di permutazione delta p(i) j
  - Norma Euclidea
  - determinante

Decomposizione L'UP

PA = LU

(unit lower triangular) (upper triangular)

Ax = b iff **PA x = Pb**

LUx = Pb

L y = Pb

U x = y

Sistemi di equazioni lineari

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1 , \\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2 , \\
&\vdots \\
a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n &= b_n .
\end{aligned}
$$

$$
\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}
$$

$$\boxed{A\mathbf{x}=\mathbf{b}}$$

Rango pieno = n

Se piu' o meno equazioni rispetto alle incognite?

Decomposizione LUP

Sostituzione in avanti

$$
\begin{aligned}
y_1 &= b_{\pi[1]} , \\
l_{21} y_1 + y_2 &= b_{\pi[2]} , \\
l_{31} y_1 + l_{32} y_2 + y_3 &= b_{\pi[3]} , \\
&\vdots \\
l_{n1} y_1 + l_{n2} y_2 + l_{n3} y_3 + \cdots + y_n &= b_{\pi[n]} .
\end{aligned}
$$

$$y_2 = b_{\pi[2]} - l_{21} y_1 .$$

$$y_3 = b_{\pi[3]} - (l_{31} y_1 + l_{32} y_2) .$$

Decomposizione LUP

Sostituzione all'indietro

$$
\begin{aligned}
u_{11}x_1 + u_{12}x_2 + \cdots + u_{1,n-2}x_{n-2} + u_{1,n-1}x_{n-1} + u_{1n}x_n &= y_1 , \\
u_{22}x_2 + \cdots + u_{2,n-2}x_{n-2} + u_{2,n-1}x_{n-1} + u_{2n}x_n &= y_2 , \\
&\vdots \\
u_{n-2,n-2}x_{n-2} + u_{n-2,n-1}x_{n-1} + u_{n-2,n}x_n &= y_{n-2} , \\
u_{n-1,n-1}x_{n-1} + u_{n-1,n}x_n &= y_{n-1} , \\
u_{n,n}x_n &= y_n .
\end{aligned}
$$

Thus, we can solve for $x_n, x_{n-1}, \ldots, x_1$ successively as follows:

$$
\begin{aligned}
x_n &= y_n/u_{n,n} , \\
x_{n-1} &= (y_{n-1} - u_{n-1,n}x_n)/u_{n-1,n-1} , \\
x_{n-2} &= (y_{n-2} - (u_{n-2,n-1}x_{n-1} + u_{n-2,n}x_n))/u_{n-2,n-2} , \\
&\vdots
\end{aligned}
$$

$$x_i = \left(y_i - \sum_{j=i+1}^{n} u_{ij}x_j \right) / u_{ii} .$$

Soluzione avendo LUP

```
LUP-SOLVE(L, U, π, b)
n ← rows[L]
for i ← 1 to n
        do y_i ← b_{π[i]} − Σ_{j=1}^{i−1} l_{ij} y_j
for i ← n downto 1
        do x_i ← (y_i − Σ_{j=i+1}^{n} u_{ij} x_j) / u_{ii}
return x
```

Costo computazionale?

Come calcolo LU per A nxn non-singolare

Eliminazione di Gauss

$$A = \begin{pmatrix} a_{11} & w^T \\ v & A' \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 \\ v/a_{11} & I_{n-1} \end{pmatrix} \begin{pmatrix} a_{11} & w^T \\ 0 & A' - vw^T/a_{11} \end{pmatrix}$$

Complemento di Shur di A rispetto ad a_{11}

$$A' - vw^T/a_{11}$$

A nonsingolare ➔ Complemento di Shur nonsingolare (rango pieno)

Calcolare LU

Computazione ricorsiva

$$
\begin{aligned}
A &= \begin{pmatrix} 1 & 0 \\ v/a_{11} & I_{n-1} \end{pmatrix} \begin{pmatrix} a_{11} & w^{\mathrm{T}} \\ 0 & A' - vw^{\mathrm{T}}/a_{11} \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ v/a_{11} & I_{n-1} \end{pmatrix} \begin{pmatrix} a_{11} & w^{\mathrm{T}} \\ 0 & L'U' \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ v/a_{11} & L' \end{pmatrix} \begin{pmatrix} a_{11} & w^{\mathrm{T}} \\ 0 & U' \end{pmatrix} \\
&= LU \, ,
\end{aligned}
$$

Calcolare LU

Versione iterativa (“tail recursion”)

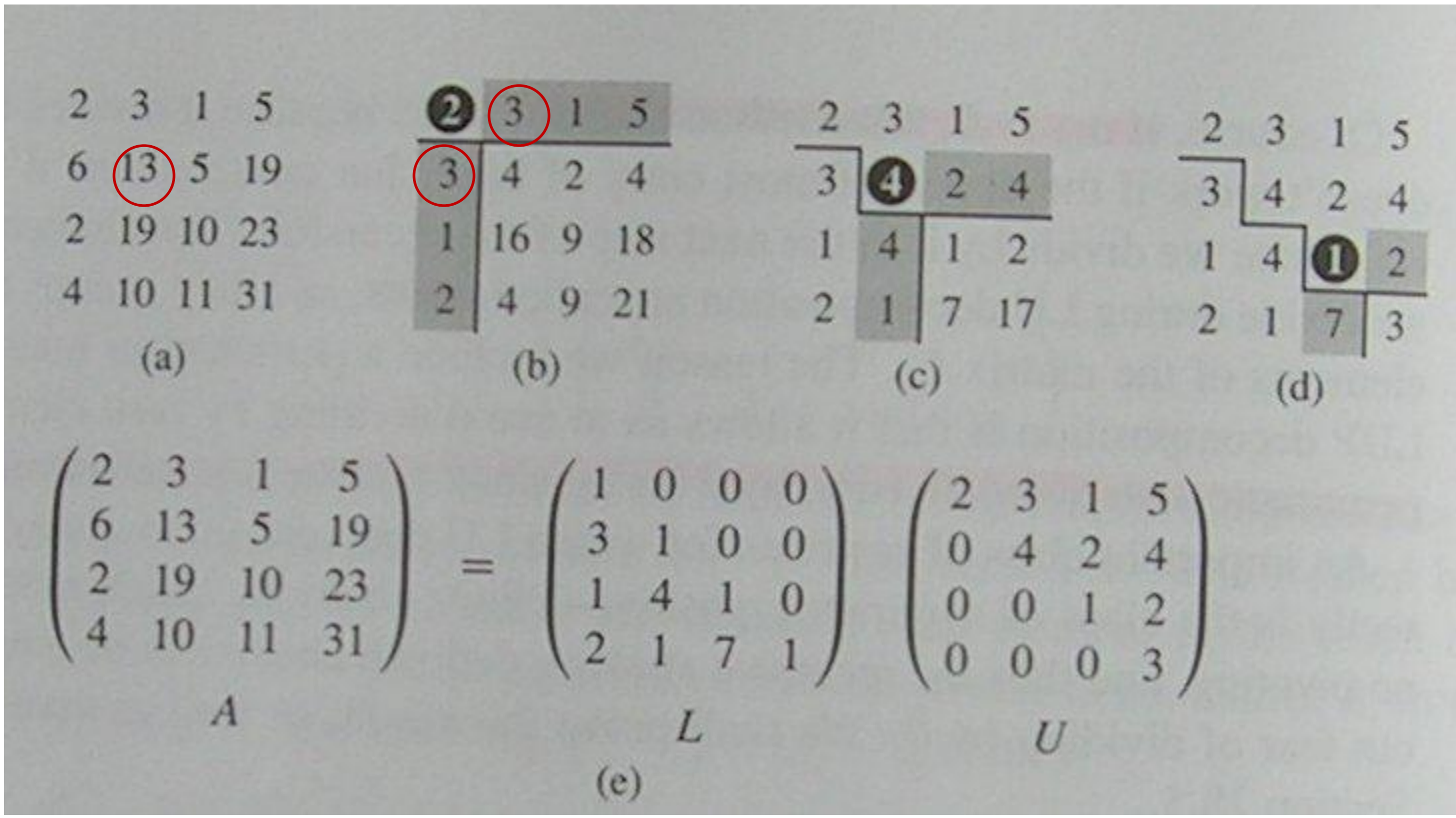
```
LU-DECOMPOSITION(A)
n ← rows[A]
for k ← 1 to n
        do u_kk ← a_kk
           for i ← k + 1 to n
                do l_ik ← a_ik / u_kk        ▷ l_ik holds v_i
                   u_ki ← a_ki               ▷ u_ki holds w_i^T
           for i ← k + 1 to n
                do for j ← k + 1 to n
                        do a_ij ← a_ij − l_ik u_kj
return L and U
```

Complemento di Shur

Ho veramente bisogno di due matrici?

Calcolare LU

"in place"



Calcolare LUP

Non si puo' dividere per zero (o per numeri molto piccoli)
➔ pivoting

$$QA = \begin{pmatrix} a_{k1} & w^{T} \\ v & A' \end{pmatrix},$$

$$\begin{aligned} QA &= \begin{pmatrix} a_{k1} & w^{T} \\ v & A' \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ v/a_{k1} & I_{n-1} \end{pmatrix} \begin{pmatrix} a_{k1} & w^{T} \\ 0 & A' - vw^{T}/a_{k1} \end{pmatrix}. \end{aligned}$$

Nonsingolare… si puo' trovare ric. una decomposizione LUP

Calcolare LUP

…

$$P'(A' - vw^{\mathrm{T}}/a_{k1}) = L'U' .$$

Define

$$P = \begin{pmatrix} 1 & 0 \\ 0 & P' \end{pmatrix} Q ,$$

$$\begin{aligned} PA &= \begin{pmatrix} 1 & 0 \\ 0 & P' \end{pmatrix} QA \\ &= \begin{pmatrix} 1 & 0 \\ 0 & P' \end{pmatrix} \begin{pmatrix} 1 & 0 \\ v/a_{k1} & I_{n-1} \end{pmatrix} \begin{pmatrix} a_{k1} & w^{\mathrm{T}} \\ 0 & A' - vw^{\mathrm{T}}/a_{k1} \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ P'v/a_{k1} & P' \end{pmatrix} \begin{pmatrix} a_{k1} & w^{\mathrm{T}} \\ 0 & A' - vw^{\mathrm{T}}/a_{k1} \end{pmatrix} \end{aligned}$$

Calcolare LUP

…

$$
\begin{aligned}
&= \begin{pmatrix} 1 & 0 \\ P'v/a_{k1} & I_{n-1} \end{pmatrix} \begin{pmatrix} a_{k1} & w^{\mathrm{T}} \\ 0 & P'(A' - vw^{\mathrm{T}}/a_{k1}) \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ P'v/a_{k1} & I_{n-1} \end{pmatrix} \begin{pmatrix} a_{k1} & w^{\mathrm{T}} \\ 0 & L'U' \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 \\ P'v/a_{k1} & L' \end{pmatrix} \begin{pmatrix} a_{k1} & w^{\mathrm{T}} \\ 0 & U' \end{pmatrix} \\
&= LU ,
\end{aligned}
$$

Calcolare LUP

$$a_{ij} = \begin{cases} l_{ij} & \text{if } i > j \,, \\ u_{ij} & \text{if } i \le j \,. \end{cases}$$

LUP-DECOMPOSITION(A)

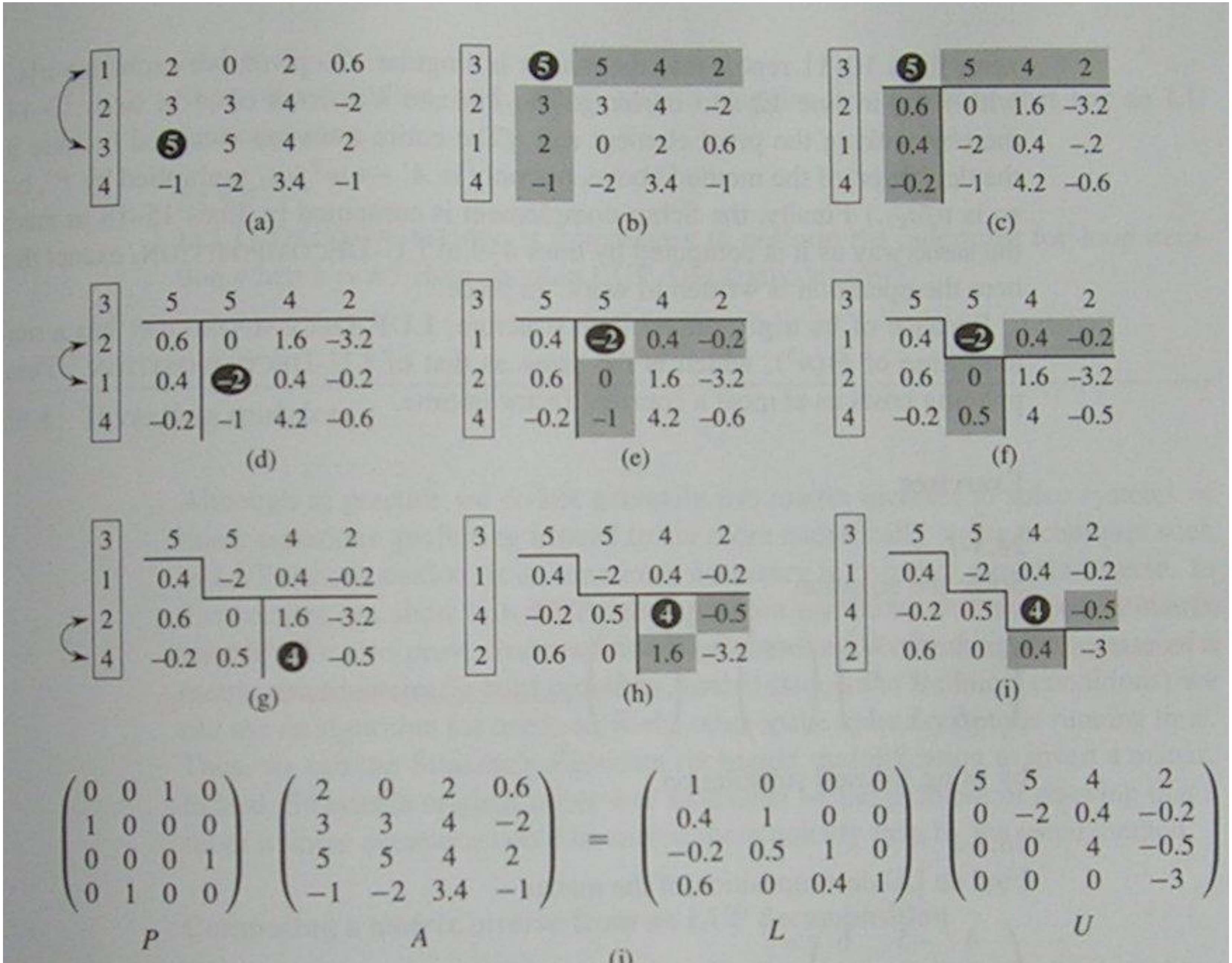
```
n ← rows[A]
for i ← 1 to n
      do π[i] ← i
for k ← 1 to n
      do p ← 0
         for i ← k to n
              do if |a_ik| > p
                    then p ← |a_ik|
                         k' ← i
         if p = 0
           then error "singular matrix"
         exchange π[k] ↔ π[k']
         for i ← 1 to n
              do exchange a_ki ↔ a_k'i
         for i ← k + 1 to n
              do a_ik ← a_ik / a_kk
                 for j ← k + 1 to n
                      do a_ij ← a_ij − a_ik a_kj
```

find largest

Divisione per il pivot

Complemento di Shur

Calcolare LUP

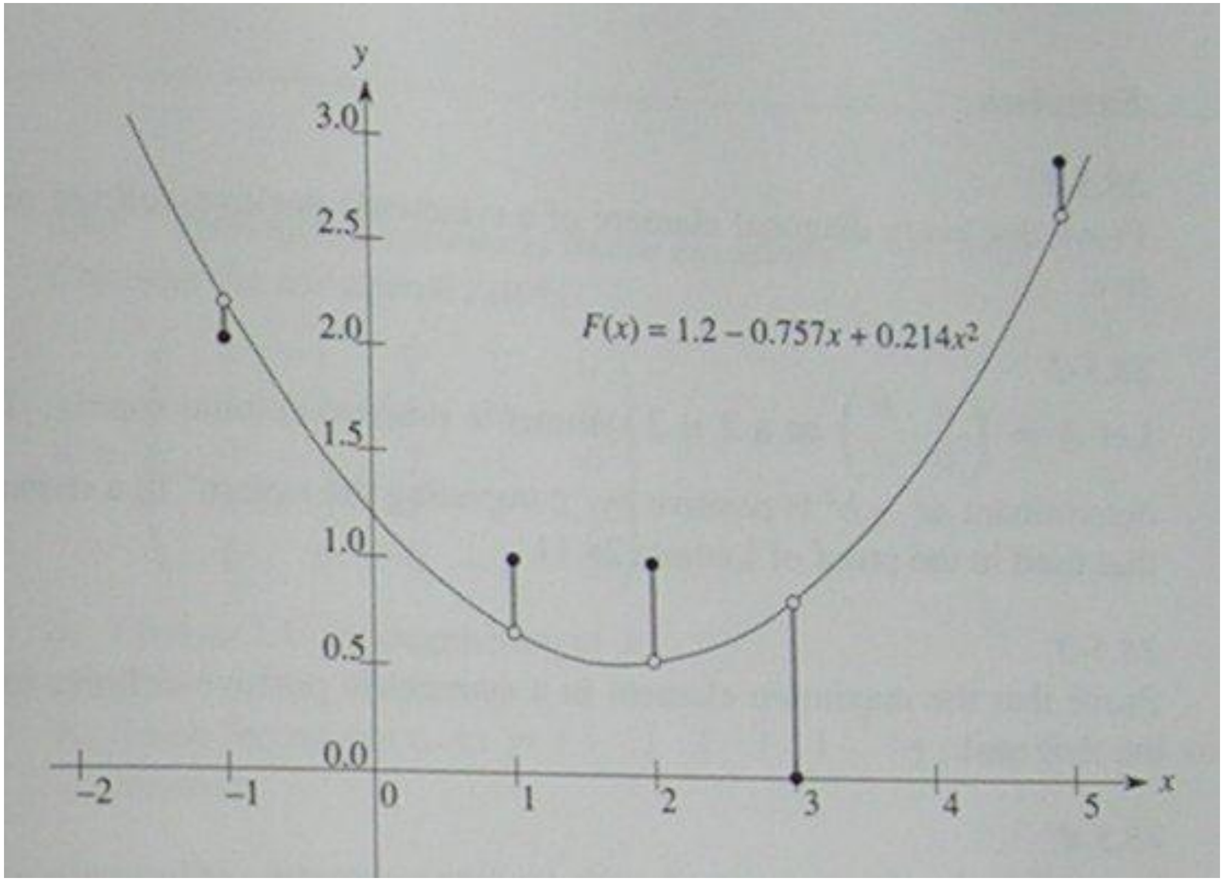


Inversione di matrici

- Posso usare i sistemi LUP (risolvere un insieme di sistemi lineari)?
- SI'! $\mathbf{A\ X = I_n}$
- $\mathbf{A\ x_i = e_i}$
- Quant' e' il costo computazionale?
- Si puo' fare "meglio"? Si veda moltiplicazione

Approsimazione con minimi quadrati, un esempio

$(x1, y1)$ ….. (xm,ym) $F(x) = c_1 + c_2\, x + c_3\, x^2$



Piu' punti sperimentali m di parametri n

Minimi quadrati

Somma pesata linearmente delle funzioni base

$$
\begin{aligned}
Ac &= \begin{pmatrix} f_1(x_1) & f_2(x_1) & \dots & f_n(x_1) \\ f_1(x_2) & f_2(x_2) & \dots & f_n(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x_m) & f_2(x_m) & \dots & f_n(x_m) \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} \\
&= \begin{pmatrix} F(x_1) \\ F(x_2) \\ \vdots \\ F(x_m) \end{pmatrix}
\end{aligned}
$$

Sovradeterminato!

Non cerco la soluzione esatta ma quella che si avvicina di piu’

Minimi quadrati

Errori di appros.: **η= A c - y**

$$\|\eta\| = \left(\sum_{i=1}^{m} \eta_i^2\right)^{1/2} .$$

Since

$$\|\eta\|^2 = \|Ac - y\|^2 = \sum_{i=1}^{m}\left(\sum_{j=1}^{n} a_{ij}c_j - y_i\right)^2 ,$$

Come minimizzo gli errori di appros.?

$$\frac{d\,\|\eta\|^2}{dc_k} = \sum_{i=1}^{m} 2\left(\sum_{j=1}^{n} a_{ij}c_j - y_i\right) a_{ik} = 0 .$$

Minimi quadrati

$$(Ac - y)^T A = 0$$

or, equivalently (

$$A^T(Ac - y) = 0$$

which implies

$$A^T Ac = A^T y \,.$$

Equazione normale

A rango di colonna pieno

$$c = ((A^T A)^{-1} A^T) y$$
$$= A^+ y \,.$$

Pseudo-inversa di A

Autovalori ed autovettori

- Pagerank by Brin and Page
- Page, Lawrence; Brin, Sergey; Motswana, Rajeev and Winograd, Terry (1999). *The PageRank citation ranking: Bringing order to the Web*
- L'algoritmo di PageRank è stato brevettato (brevetto US 6285999) dalla Stanford University

Autovalori ed autovettori

- **$A\,x = \lambda\,x$**
- **$(A - \lambda\,I)\,x = 0$**
- **det $(A - \lambda\,I) = 0$ … campo complesso n radici**

- **A simmetrica definita positiva, base ortonormale di autovettori**
- **Decomposizione spettrale**
- **$A = \Sigma\,\lambda\,v\,v^T$ $\qquad$ $A\,x = \ldots$**

Autovalori ed autovettori

- **Come trovo l'autovettore corrispondente all'autovalore massimo? (autovettore dominante)**
- **Potenze**

x = a1 **v1 +** a2 **v2 +**a3 **v3 + …**

A x = ….