Perché i sistemi lineari sono importanti

Dario A. Bini

Dipartimento di Matematica, Università di Pisa
www.dm.unipi.it/~bini

4 Febbraio 2008

1. Premessa

2. Sistemi lineari
   - Dove si incontrano
   - Come si risolvono
   - Chi li ha inventati

Matematica: tra Astrazione e Applicazione

Una domanda

Ma a cosa serve la Matematica?

La Matematica è un bellissimo giocattolo che

- non stanca mai e non si rompe mai
- sviluppa il pensiero libero la creatività e la fantasia
- permette di creare quasi senza limiti strutture eleganti col massimo rigore logico
- non ha colorazioni politiche nè religiose
- è proprietà di tutti
- è uno strumento **indispensabile** per risolvere una gran parte di problemi del mondo reale.

Matematica: tra Astrazione e Applicazione

Una domanda

Ma a cosa serve la Matematica?

La Matematica è un bellissimo giocattolo che

- non stanca mai e non si rompe mai
- sviluppa il pensiero libero la creatività e la fantasia
- permette di creare quasi senza limiti strutture eleganti col massimo rigore logico
- non ha colorazioni politiche nè religiose
- è proprietà di tutti
- è uno strumento **indispensabile** per risolvere una gran parte di problemi del mondo reale.

La matematica intorno a noi

Ogni giorno ciascuno di noi fa uso inconsapevole di matematica

- la fotografia digitale: sharpening, deblurring, jpeg...
- la musica digitale: CD, mp3, ipod
- film e tv digitali: alta definizione, dvd, digitale terrestre, digitale satellitare
- telefonia mobile
- gps, cartografia nautica, navigatori satellitari
- volo automatico
- crittografia (bancomat, internet)
- sport
- internet (ricerca di informazioni)

La matematica intorno a noi

- La TAC, la RNM
- modelli cardiaci
- modelli della circolazione sanguigna (aneurismi, ostruzioni)
- modelli epidemiologici
- previsioni del tempo
- indagini statistiche, exit poll
- modelli di code
- analisi di rischio (assicurazioni)
- modelli finanziari

- le armi “intelligenti”
- sistemi antimissile
- aerei killer

I sistemi lineari sono un ingrediente comune nella maggior parte di applicazioni della matematica

Molti nostri comportamenti di tutti i giorni vengono resi possibili dalla risoluzione efficiente di sistemi lineari

Sistemi lineari

$$\begin{cases} ax + by = e \\ cx + dy = f \end{cases}$$

2 equazioni con 2 incognite

$$\begin{cases} ax + by + cz = \ell \\ dx + ey + fz = m \\ gx + hy + iz = n \end{cases}$$

3 equazioni con 3 incognite

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n & = & b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n & = & b_2 \\ \vdots & \vdots & \vdots \\ a_{n,1}x_1 + a_{n,2}x_2 + \cdots + a_{n,n}x_n & = & b_n \end{cases}$$

n equazioni con n incognite

Sistemi lineari

$$\begin{cases} ax + by = e \\ cx + dy = f \end{cases}$$

2 equazioni con 2 incognite

$$\begin{cases} ax + by + cz = \ell \\ dx + ey + fz = m \\ gx + hy + iz = n \end{cases}$$

3 equazioni con 3 incognite

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n & = & b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n & = & b_2 \\ \vdots & \vdots & \vdots \\ a_{n,1}x_1 + a_{n,2}x_2 + \cdots + a_{n,n}x_n & = & b_n \end{cases}$$

n equazioni con n incognite

Sistemi lineari

$$\begin{cases} ax + by = e \\ cx + dy = f \end{cases}$$

2 equazioni con 2 incognite

$$\begin{cases} ax + by + cz = \ell \\ dx + ey + fz = m \\ gx + hy + iz = n \end{cases}$$

3 equazioni con 3 incognite

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n & = & b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n & = & b_2 \\ \vdots & \vdots & \vdots \\ a_{n,1}x_1 + a_{n,2}x_2 + \cdots + a_{n,n}x_n & = & b_n \end{cases}$$

n equazioni con n incognite

Sistemi Lineari

$$\sum_{j=1}^{n} a_{i,j} x_j = b_i, \qquad i = 1, 2, \ldots, n$$

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,n} \\ a_{2,1} & a_{2,2} & \ldots & a_{2,n} \\ \vdots & \vdots & \ldots & \vdots \\ a_{n,1} & a_{n,2} & \ldots & a_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

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

A: matrice del sistema

$\mathbf{x}$: vettore delle incognite

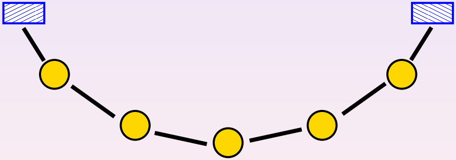
$\mathbf{b}$: vettore dei termini noti

Dove intervengono i sistemi?

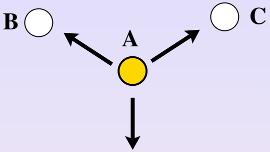
Esempio 1: problemi di equilibrio

Una collana è costituita da n perle collegate da un filo elastico dove la forza esercitata tra due perle è proporzionale alla distanza delle perle. Le perle sono soggette anche alla forza peso.

Qual è la configurazione di equilibrio se appendiamo la collana a due estremi fissi?



Condizione di equilibrio: per ogni perla la somma delle forze deve essere nulla



Peso + forza sinistra + forza destra=0

$$P + kAB + kAC = 0$$

Sistema di coordinate: $A = (x, y)$, $B(b_1, b_2)$, $C = (c_1, c_2)$

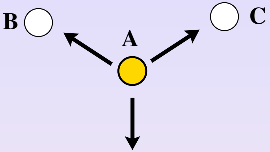
allora da $P + k(B - A) + k(C - A) = 0$ ho

$$(0, p) + k(b_1 - x, b_2 - y) + k(c_1 - x, c_2 - y) = 0$$

da cui

$$\begin{aligned} -b_1 + 2x - c_1 &= 0 \\ -b_2 + 2y - c_2 &= p/k \end{aligned}$$

Condizione di equilibrio: per ogni perla la somma delle forze deve essere nulla



Peso + forza sinistra + forza destra=0

$$P + kAB + kAC = 0$$

Sistema di coordinate: $A = (x, y)$, $B(b_1, b_2)$, $C = (c_1, c_2)$

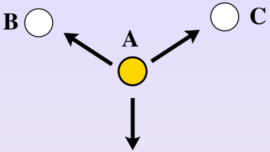
allora da $P + k(B - A) + k(C - A) = 0$ ho

$$(0, p) + k(b_1 - x, b_2 - y) + k(c_1 - x, c_2 - y) = 0$$

da cui

$$-b_1 + 2x - c_1 = 0$$
$$-b_2 + 2y - c_2 = p/k$$

Condizione di equilibrio: per ogni perla la somma delle forze deve essere nulla



Peso + forza sinistra + forza destra=0

$$P + kAB + kAC = 0$$

Sistema di coordinate: $A = (x, y)$, $B(b_1, b_2)$, $C = (c_1, c_2)$

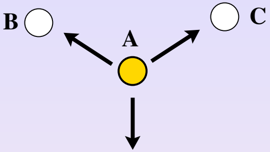
allora da $P + k(B - A) + k(C - A) = 0$ ho

$$(0, p) + k(b_1 - x, b_2 - y) + k(c_1 - x, c_2 - y) = 0$$

da cui

$$\begin{aligned} -b_1 + 2x - c_1 &= 0 \\ -b_2 + 2y - c_2 &= p/k \end{aligned}$$

Condizione di equilibrio: per ogni perla la somma delle forze deve essere nulla



Peso + forza sinistra + forza destra=0

$$P + kAB + kAC = 0$$

Sistema di coordinate: $A = (x, y)$, $B(b_1, b_2)$, $C = (c_1, c_2)$

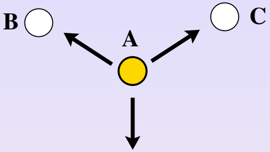
allora da $P + k(B - A) + k(C - A) = 0$ ho

$$(0, p) + k(b_1 - x, b_2 - y) + k(c_1 - x, c_2 - y) = 0$$

da cui

$$-b_1 + 2x - c_1 = 0$$
$$-b_2 + 2y - c_2 = p/k$$

Condizione di equilibrio: per ogni perla la somma delle forze deve essere nulla



Peso + forza sinistra + forza destra=0

$$P + kAB + kAC = 0$$

Sistema di coordinate: $A = (x, y)$, $B(b_1, b_2)$, $C = (c_1, c_2)$

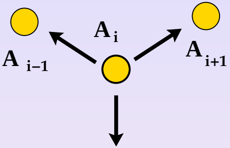
allora da $P + k(B - A) + k(C - A) = 0$ ho

$$(0, p) + k(b_1 - x, b_2 - y) + k(c_1 - x, c_2 - y) = 0$$

da cui

$$\begin{aligned} -b_1 + 2x - c_1 &= 0 \\ -b_2 + 2y - c_2 &= p/k \end{aligned}$$

In generale se considero una collana di n perle che indico con $A_i = (x_i, y_i)$, $i = 1, \ldots, n$, allora



Forza dx:	$k(x_{i+1} - x_i, y_{i+1} - y_i)$
Forza sx :	$k(x_{i-1} - x_i, y_{i-1} - y_i)$
Forza peso:	$(0, p_i)$

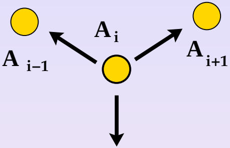
le equazioni diventano:

$$-x_{i-1} + 2x_i - x_{i+1} = 0, \quad i = 1, 2, \ldots, n$$

$$-y_{i-1} + 2y_i - y_{i+1} = p_i/k, \quad i = 1, 2, \ldots, n$$

Sono due sistemi di n equazioni in n incognite.

In generale se considero una collana di n perle che indico con $A_i = (x_i, y_i)$, $i = 1, \dots, n$, allora



Forza dx: $k(x_{i+1} - x_i, y_{i+1} - y_i)$
Forza sx : $k(x_{i-1} - x_i, y_{i-1} - y_i)$
Forza peso: $(0, p_i)$

le equazioni diventano:

$$-x_{i-1} + 2x_i - x_{i+1} = 0, \quad i = 1, 2, \dots, n$$
$$-y_{i-1} + 2y_i - y_{i+1} = p_i/k, \quad i = 1, 2, \dots, n$$

Sono due sistemi di n equazioni in n incognite.

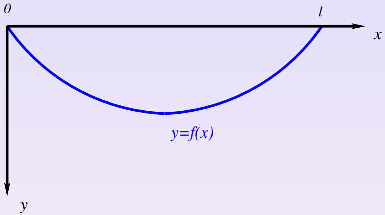
La matrice dei due sistemi è la stessa:

$$A = \begin{bmatrix} 2 & -1 & 0 & \dots & 0 \\ -1 & 2 & -1 & \ddots & \vdots \\ 0 & -1 & 2 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \dots & 0 & -1 & 2 \end{bmatrix}$$

A è una speciale matrice **tridiagonale**

Un problema più complesso

Se al posto di una collana consideriamo un elastico **continuo** allora il modello matematico è un po' diverso



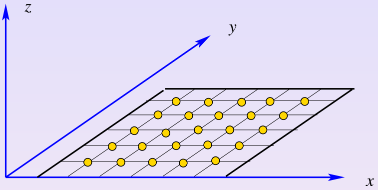
$$\begin{cases} kf''(x) = m(x) \\ f(0) = 0, \quad f(\ell) = 0 \end{cases}$$

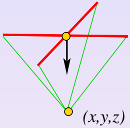
La soluzione dell'equazione differenziale si approssima calcolando $f(x)$ in n punti.
Si ottiene ancora un sistema $n \times n$: più grande è n più alta è la precisione nell'approssimazione.

$n = 100, 1000, 10000, 100000, \ldots$

Problemi multidimensionali

Ora vogliamo calcolare la deformazione di una rete “da materasso”





Stavolta ogni punto della rete è individuato da tre coordinate (x, y, z)

Inoltre, se la rete ha n maglie per lato abbiamo n^2 punti

In totale ci sono tre sistemi, ciascuno di n^2 equazioni e n^2 incognite

Se $n = 100000$, vale $n^2 = 10^{10}$

Tutti e tre i sistemi hanno la stessa matrice che li definisce. La matrice è fatta così:

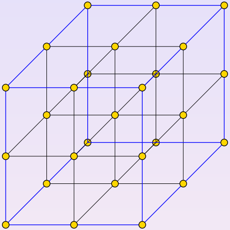
$$A = \begin{bmatrix} B & -I & 0 & \dots & 0 \\ -I & B & -I & \ddots & \vdots \\ 0 & -I & B & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -I \\ 0 & \dots & 0 & -I & B \end{bmatrix}$$

dove

$$B = \begin{bmatrix} 4 & -1 & & 0 \\ -1 & 4 & \ddots & \\ & \ddots & \ddots & -1 \\ 0 & & -1 & 4 \end{bmatrix}, \quad I = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix}$$

A è una matrice **tridiagonale a blocchi con blocchi tridiagonali**

Se al posto dello spostamento di un punto della rete dobbiamo studiare la pressione, la temperatura o l'umidità in una zona dello spazio rappresentata da un reticolo di $n \times n \times n$ punti



abbiamo un numero di incognite e di equazioni pari a n^3

Se $n = 100000$ allora $n^3 = 10^{15}$: un **milione di miliardi**

La matrice del sistema è tridiagonale a blocchi con i blocchi che a loro volta sono tridiagonali a blocchi con blocchi tridiagonali

È come un gioco di scatole cinesi....

Per problemi multidimensionali il gioco delle scatole cinesi si ripete tante volte quanta è la dimensione dello spazio

Problemi correlati

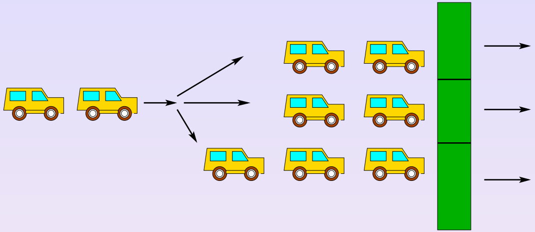
Calcolare le frequenze e i modi di vibrazione di sistemi complessi: corde di chitarra, tamburi, binari ferroviari, edifici, ponti, ecc

In questo caso il sistema da risolvere è del tipo

$$A\mathbf{x} = \lambda\mathbf{x}$$

dove λ è anch'essa un'incognita chiamata **autovalore** e ogni soluzione non nulla $\mathbf{x}$ è chiamata **autovettore**.

Il problema delle code

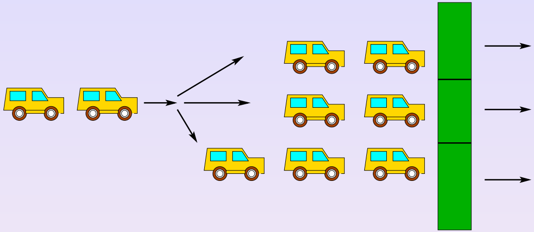


Il problema: Ci sono m caselli dell'autostrada aperti:

- ad ogni istante arrivano α auto con probabilità nota
- ogni auto sceglie la fila più corta
- ad ogni istante ogni casellante serve una macchina

Qual è la probabilità che al trascorrere del tempo si formino code?
Qual è la lunghezza media che mi devo aspettare?

Il problema delle code



Il problema: Ci sono *m* caselli dell'autostrada aperti:

- ad ogni istante arrivano α auto con probabilità nota
- ogni auto sceglie la fila più corta
- ad ogni istante ogni casellante serve una macchina

Qual è la probabilità che al trascorrere del tempo si formino code?
Qual è la lunghezza media che mi devo aspettare?

Problemi correlati

- Studio di code che si formano nei “servizi” (banche, supermercati, parcheggi, impianti di risalita)
- Protocolli di trasmissione dati: internet, wireless (protocollo IEEE 802.11), telefonia mobile e fissa
- Studio dei rischi (assicurazioni, investimenti)
- diagnostica medica

Se $a(t)$ è il numero totale di macchine in coda al tempo t allora

$$a(t+1) = \begin{cases} a(t) + \alpha - m & \text{se} \quad a(t) + \alpha - m \geq 0 \\ 0 & \text{altrimenti} \end{cases}$$

Questa formula permette di calcolare facilmente la probabilità:

$p_{i,j}$ che in un qualsiasi istante si passi da j macchine in coda a i macchine in coda

Indico con $x_i(t)$ la probabilità che ci siano i macchine in coda al tempo t.

Come si fa a calcolare $x_i(t+1)$ data $x_i(t)$?

Se $a(t)$ è il numero totale di macchine in coda al tempo t allora

$$a(t+1) = \begin{cases} a(t) + \alpha - m & \text{se} \quad a(t) + \alpha - m \geq 0 \\ 0 & \text{altrimenti} \end{cases}$$

Questa formula permette di calcolare facilmente la probabilità:

$p_{i,j}$ che in un qualsiasi istante si passi da j macchine in coda a i macchine in coda

Indico con $x_i(t)$ la probabilità che ci siano i macchine in coda al tempo t.

Come si fa a calcolare $x_i(t+1)$ data $x_i(t)$?

Se $a(t)$ è il numero totale di macchine in coda al tempo t allora

$$a(t+1) = \begin{cases} a(t) + \alpha - m & \text{se} \quad a(t) + \alpha - m \geq 0 \\ 0 & \text{altrimenti} \end{cases}$$

Questa formula permette di calcolare facilmente la probabilità:

$p_{i,j}$ che in un qualsiasi istante si passi da j macchine in coda a i macchine in coda

Indico con $x_i(t)$ la probabilità che ci siano i macchine in coda al tempo t.

Come si fa a calcolare $x_i(t+1)$ data $x_i(t)$?

Per le proprietà delle probabilità vale

$$x_i(t+1) = p_{i,1}x_1(t) + p_{i,2}x_2(t) + p_{i,3}x_3(t) + \cdots$$

cioè

$$x_i(t+1) = \sum_{j=1}^{\infty} p_{i,j}x_j(t)$$

In notazione di matrici:

$$\mathbf{x}(t+1) = P\mathbf{x}(t)$$

dove P è la matrice infinita i cui elementi sono $p_{i,j}$

$$P = \begin{bmatrix} p_{1,1} & p_{1,2} & p_{1,2} & \cdots \\ p_{2,1} & p_{2,2} & p_{2,3} & \cdots \\ p_{3,1} & p_{3,2} & p_{3,3} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Dall'espressione

$$\mathbf{x}(t+1) = P\mathbf{x}(t)$$

posso allora calcolare il vettore $\mathbf{x}(t+1)$ conoscendo il vettore $\mathbf{x}(t)$.

Per calcolare cosa succede quando $t \to \infty$, nell'ipotesi che esista

$$\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}$$

basta risolvere il sistema lineare **infinito**

$$\mathbf{x} = P\mathbf{x}.$$

Dall'espressione

$$\mathbf{x}(t+1) = P\mathbf{x}(t)$$

posso allora calcolare il vettore $\mathbf{x}(t+1)$ conoscendo il vettore $\mathbf{x}(t)$.

Per calcolare cosa succede quando $t \to \infty$, nell'ipotesi che esista

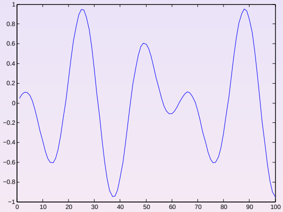
$$\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}$$

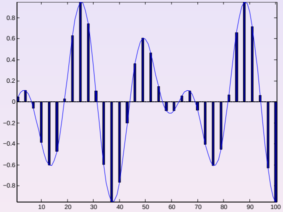
basta risolvere il sistema lineare **infinito**

$$\mathbf{x} = P\mathbf{x}.$$

Elaborazione di suoni

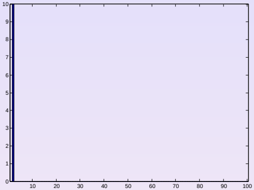
Suono come funzione da $\mathbb{R}$ in $\mathbb{R}$:
il dominio è il tempo, il codominio è l'intensità sonora
Con la tecnologia digitale un suono è individuato da una ennupla di numeri $(s_1, s_2, \ldots, s_n)$

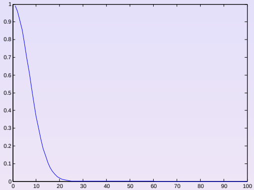




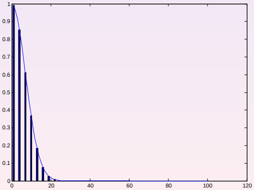
Un modello per il riverbero

Un suono impulsivo generato in un ambiente chiuso ampio genera un riverbero





In forma discreta, il riverbero è dato da $(a_1, a_2, a_3, \ldots,)$



Un suono qualsiasi lo posso vedere come somma di tanti suoni impulsivi che si manifestano uno dopo l'altro

Posso allora costruire il suono con riverbero sommando i singoli riverberi dei singoli suoni impulsivi

$\mathbf{a} = (a_1, a_2, a_3, \ldots)$ riverbero di un segnale impulsivo unitario

$\mathbf{x} = (x_1, x_2, x_3, \ldots)$ suono originale

$\mathbf{y} = (y_1, y_2, y_3, \ldots)$ suono con riverbero

Vale allora

$$\begin{aligned}
\mathbf{y} &= x_1(a_1, a_2, a_3, \ldots) && \text{riverbero del primo impulso}\\
&\quad + x_2(0, a_1, a_2, a_3, \ldots) && \text{riverbero del secondo impulso}\\
&\quad\quad + x_3(0, 0, a_1, a_2, a_3 \ldots) && \text{riverbero del terzo impulso}\\
&\quad\quad\quad + \cdots
\end{aligned}$$

cioè

$$y_i = \sum_{j=1}^{i} a_{i-j+1} x_j, \quad i = 1, 2, \ldots$$

Calcolare il suono con riverbero comporta calcolare una somma di prodotti

Togliere il riverbero da un segnale comporta la risoluzione di un sistema lineare

Nella musica di un CD, in un secondo ci sono 44.100 “segnali impulsivi” quindi pulire dal riverbero 10 secondi di musica comporta la risoluzione di un sistema di 441.000 equazioni e incognite.

Se voglio fare questo mentre ascolto musica (in tempo reale) devo essere in grado di risolvere questo sistema in meno di 10 secondi

Elaborazioni di immagini

Una immagine, ad esempio una fotografia in bianco/nero, viene rappresentata con una tabella di $m \times n$ numeri. Il numero sulla riga i e sulla colonna j di questa tabella dà l'intensità luminosa del corrispondente puntolino di medesime coordinate nella foto.

Di solito il nero si rappresenta con 0 e il bianco con 255

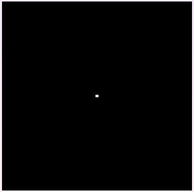
Una foto a colori è rappresentata da tre tabelle di numeri: una per il rosso, una per il verde e una per il blu (codifica RGB).

Infatti il colore di ogni puntolino è il risultato di una parte di rosso, una di verde e una di blu

Ad esempio, nel caso bianco/nero

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 255 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

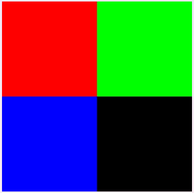
rappresenta una immagine tutta nera con un puntolino bianco al centro



Nel caso a colori, le tre tabelle

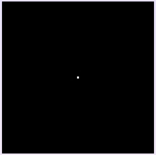
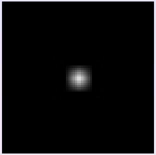
$$R = \begin{bmatrix} 255 & 255 & 255 & 0 & 0 & 0 \\ 255 & 255 & 255 & 0 & 0 & 0 \\ 255 & 255 & 255 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad G = \begin{bmatrix} 0 & 0 & 0 & 255 & 255 & 255 \\ 0 & 0 & 0 & 255 & 255 & 255 \\ 0 & 0 & 0 & 255 & 255 & 255 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix},$$

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 255 & 255 & 255 & 0 & 0 & 0 \\ 255 & 255 & 255 & 0 & 0 & 0 \\ 255 & 255 & 255 & 0 & 0 & 0 \end{bmatrix},$$ danno l'immagine



Modello di sfocatura

Consideriamo l'immagine costituita da un puntolino e la sua sfocata

Chiamiamo con $A = (a_{i,j})$ la tabella di numeri che descrive il punto sfocato. Per convenienza, facciamo scorrere gli indici da $-k$ a k dove $2k + 1$ è l'ampiezza della "patacca"

$$A = \begin{bmatrix} a_{-2,-2} & a_{-2,-1} & a_{-2,0} & a_{-2,1} & a_{-2,2} \\ a_{-1,-2} & a_{-1,-1} & a_{-1,0} & a_{-1,1} & a_{-1,2} \\ a_{0,-2} & a_{0,-1} & a_{0,0} & a_{0,1} & a_{0,2} \\ a_{1,-2} & a_{1,-1} & a_{1,0} & a_{1,1} & a_{1,2} \\ a_{2,-2} & a_{2,-1} & a_{2,0} & a_{2,1} & a_{2,2} \end{bmatrix}$$

Come accade nel caso dei suoni, una immagine la iterpreto come la sovrapposizione di tante immagini fatte di un solo puntolino.

L'immagine sfocata è allora la somma delle sfocature di ogni singolo puntolino

Se chiamo con X l'immagine originale e con S quella sfocata vale

$$s_{i,j} = \sum_{p,q=-k}^{k} a_{p,q} x_{i-p,j-q}$$

Sfocare significa calcolare la sommatoria

Rimettere a fuoco significa risolvere un sistema lineare, il numero di equazioni e di incognite è dato dal numero di pixel dell'immagine.

Un esempio

Foto originale:

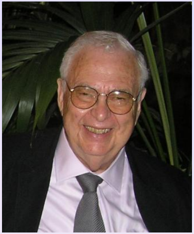


Foto sfocata:

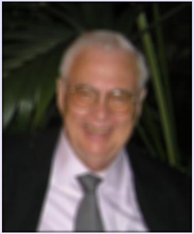
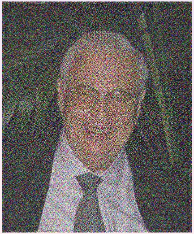
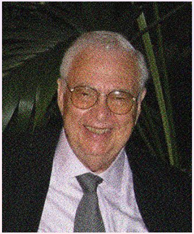


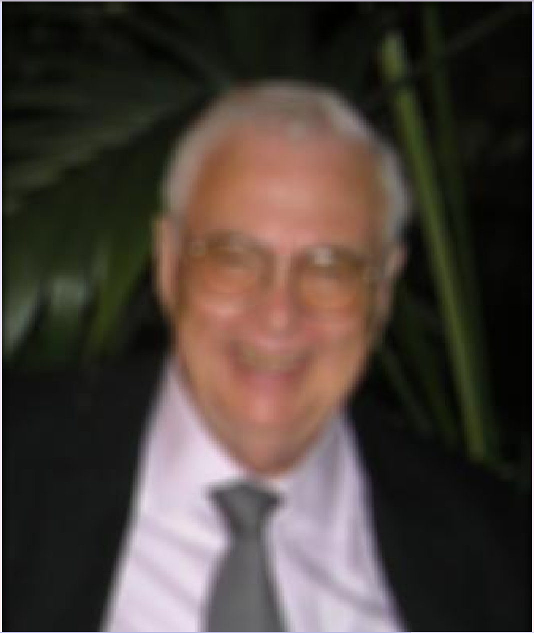
Foto rifocata:

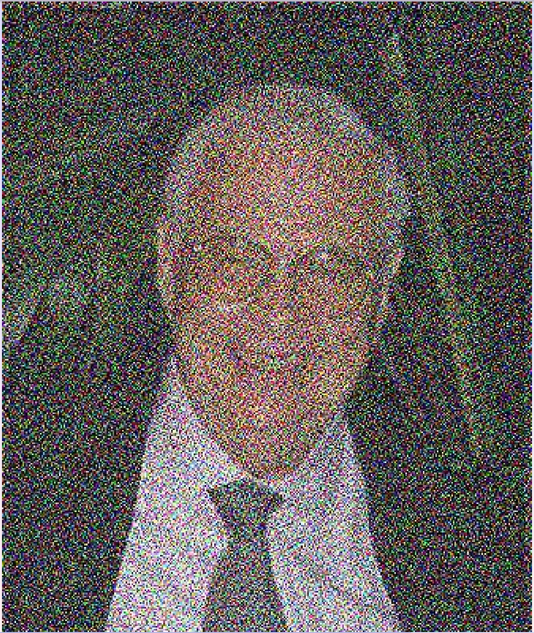
Rifocata con rumore

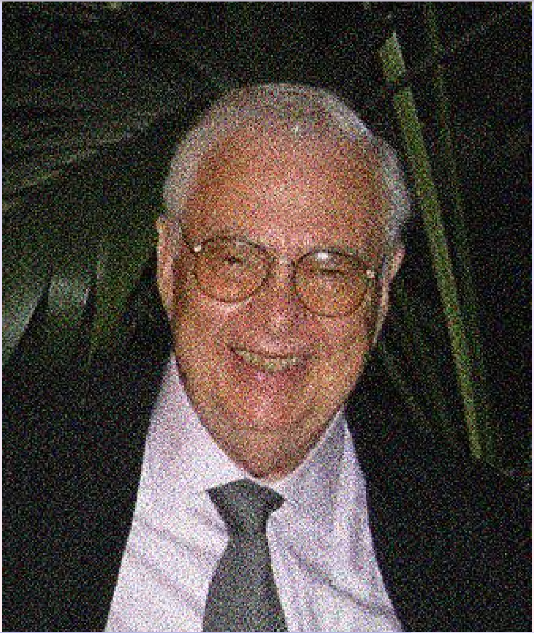


Rifocata con rumore e regolarizzazione









Matematica del Web

Internet costituisce una sorgente di problemi matematici di particolare interesse teorico e applicativo

- Page ranking (Google)
- Information retrieval
- Trust, Reputation and Reliability analysis
- Gestione del flusso delle informazioni sulla rete

Motori di ricerca e Page Rank

Due studenti dell'università di Stanford, Sergey Brin e Larry Page hanno fatto la loro fortuna inventando "Google"

Problema: Ordinare le pagine presenti sul Web in base alla loro importanza

Come si può definire l'importanza (**page rank**) di una pagina?

Motori di ricerca e Page Rank

Due studenti dell'università di Stanford, Sergey Brin e Larry Page hanno fatto la loro fortuna inventando "Google"

Problema: Ordinare le pagine presenti sul Web in base alla loro importanza

Come si può definire l'importanza (**page rank**) di una pagina?

Motori di ricerca e Page Rank

Due studenti dell'università di Stanford, Sergey Brin e Larry Page hanno fatto la loro fortuna inventando "Google"

Problema: Ordinare le pagine presenti sul Web in base alla loro importanza

Come si può definire l'importanza (**page rank**) di una pagina?

Importanza di una pagina

Varie possibilità

- in base al numero di volte che la parola cercata compare
- in base al numero dei link che da essa partono
- in base al numero dei link che ad essa arrivano
- **in base al numero delle pagine importanti che puntano alla pagina**

Importanza di una pagina

Varie possibilità

- in base al numero di volte che la parola cercata compare
- in base al numero dei link che da essa partono
- in base al numero dei link che ad essa arrivano
- **in base al numero delle pagine importanti che puntano alla pagina**

Importanza di una pagina

Varie possibilità

- in base al numero di volte che la parola cercata compare
- in base al numero dei link che da essa partono
- in base al numero dei link che ad essa arrivano
- in base al numero delle pagine importanti che puntano alla pagina

Importanza di una pagina

Varie possibilità

- in base al numero di volte che la parola cercata compare
- in base al numero dei link che da essa partono
- in base al numero dei link che ad essa arrivano
- **in base al numero delle pagine importanti che puntano alla pagina**

L'idea di Page e Brin

Ogni pagina ha una sua propria importanza che deriva dalle connessioni e non dai contenuti

L'importanza di una pagina viene trasferita in parti uguali alle pagine che essa punta

L'importanza di una pagina è data dalla somma delle frazioni di importanza che gli derivano dalle pagine che ad essa puntano

(Una persona importante dà importanza alle persone che frequenta

Una persona è importante se frequenta molte persone importanti)

Modello matematico

Numeriamo le pagine del Web da 1 a n

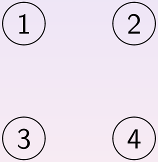
Definiamo la matrice di connettività nel seguente modo:

$$H = \begin{bmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,n} \\ h_{2,1} & h_{2,2} & \dots & h_{2,n} \\ \vdots & \vdots & & \vdots \\ h_{n,1} & h_{n,2} & \dots & h_{n,n} \end{bmatrix}$$

$h_{i,j} = 1$ se c'è un link dalla pagina i alla pagina j
$h_{i,j} = 0$ altrimenti.

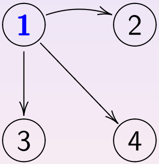
Esempio con $n = 4$

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$



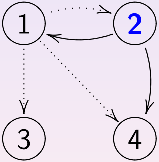
Esempio con $n = 4$

$$H = \begin{bmatrix} \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$



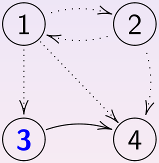
Esempio con $n = 4$

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 \\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$



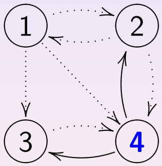
Esempio con $n = 4$

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1} \\ 0 & 1 & 1 & 0 \end{bmatrix}$$



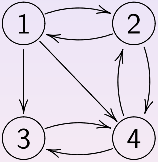
Esempio con $n = 4$

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0} \end{bmatrix}$$



Esempio con $n = 4$

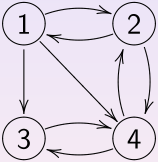
$$H = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$



Sommando i valori sulla riga i si trova il numero di link che partono dalla pagina i. Denotiamo con r_i questo numero

Esempio con $n = 4$

$$H = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$



Sommando i valori sulla riga i si trova il numero di link che partono dalla pagina i. Denotiamo con r_i questo numero

Indichiamo con $\mathbf{x}_j$ l'importanza della pagina j

Allora risulta

$$x_j = h_{1,j}\frac{x_1}{r_1} + h_{2,j}\frac{x_2}{r_2} + \cdots + h_{n,j}\frac{x_n}{r_n}$$

per $j = 1, 2, \ldots, n$.

Questo è un **sistema lineare** di n equazioni in n incognite.

Le soluzioni $x_1, x_2, \ldots, x_n$ danno il livello di importanza delle singole pagine cioè il **page rank**

Indichiamo con $\mathbf{x}_j$ l'importanza della pagina j

Allora risulta

$$x_j = h_{1,j}\frac{x_1}{r_1} + h_{2,j}\frac{x_2}{r_2} + \cdots + h_{n,j}\frac{x_n}{r_n}$$

per $j = 1, 2, \ldots, n$.

Questo è un **sistema lineare** di n equazioni in n incognite.

Le soluzioni $x_1, x_2, \ldots, x_n$ danno il livello di importanza delle singole pagine cioè il **page rank**

Indichiamo con $\mathbf{x}_j$ l'importanza della pagina j

Allora risulta

$$x_j = h_{1,j}\frac{x_1}{r_1} + h_{2,j}\frac{x_2}{r_2} + \cdots + h_{n,j}\frac{x_n}{r_n}$$

per $j = 1, 2, \ldots, n$.

Questo è un **sistema lineare** di n equazioni in n incognite.

Le soluzioni $x_1, x_2, \ldots, x_n$ danno il livello di importanza delle singole pagine cioè il **page rank**

L'equazione usata da Google è leggermente diversa

$$x_j = d(h_{1,j}\frac{x_1}{r_1} + h_{2,j}\frac{x_2}{r_2} + \cdots + h_{n,j}\frac{x_n}{r_n}) + \frac{1}{n}(1-d),$$

dove d è un parametro fra 0 e 1, di solito viene posto $d = 0.85$

I valori di x_j sono compresi fra 0 e 1.

Per calcolare il page rank occorre risolvere un sistema di n equazioni ed n incognite

Attualmente ci sono circa $n = 8.5 \times 10^9$ pagine attive

Come si risolve un sistema lineare?

Se $n = 2$ si ha

$$\begin{cases} ax + by = e \\ cx + dy = f \end{cases}$$

Il metodo di sostituzione detto anche di Eliminazione Gaussiana si applica in generale a sistemi $n \times n$

Esso richiede

$$\frac{2}{3} n^3 + \text{spiccioli}$$

operazioni aritmetiche

Come si risolve un sistema lineare?

Se $n = 8.5$ miliardi il metodo di eliminazione richiede circa

$$\frac{2}{3}(8.5 \times 10^9)^3 \approx 4.1 \times 10^{29}$$

(410 miliardi di miliardi di miliardi) operazioni aritmetiche

sono tante?

Complessità del Page Rank

Il calcolatore più veloce esistente al mondo è attualmente il **Blue Gene** dell'IBM

Ha una velocità massima di 480 teraflops cioè 4.8×10^{14} operazioni al secondo (1 tera = 1000 giga)

Per eseguire 4.1×10^{29} operazioni richiederebbe più di 27 milioni di anni

Un tempo "geologico" .... eppure Larry Page e Sergey Brin **calcolano il page rank ogni mese**

come fanno?

Caso delle immagini

Una macchina fotografica con 6 megapixel produce una fotografia dotata di 6 milioni di puntolini, ciascuno rappresentato da tre numeri (RGB)

Per rimettere a fuoco un'immagine sfocata andrebbero risolti tre sistemi lineari di 6 milioni di equazioni e di incognite

Disponendo del Blue Gene ci vorrebbero almeno 10 giorni. Eppure noi abbiamo rimesso a fuoco una foto usando un vechio *Celeron* in pochi secondi

Caso delle code

Che dire poi del caso dei modelli di code?

Lì il sistema ha infinite equazioni e infinite incognite. Anche se ci accontentassimo di calcolare solo un numero finito di incognite dovremmo comunque imporre un numero infinito di condizioni.

Come si fa?

Tecnologia hardware vs tecnologia matematica

Domanda

Possiamo fare affidamento sulla tecnologia?

Anche se la tecnologia fosse in grado di costruire un computer 1000 volte o un milione di volte più veloce non sarebbe possibile risolvere il problema di Google in tempo reale

Solo sviluppando nuovi metodi matematici è possibile risolvere i sistemi lineari complessi con tempi di calcolo brevi

Metodi iterativi

I **metodi iterativi** generano una successione di approssimazioni che convergono alla soluzione del sistema

Il calcolo di ogni approssimazione costa "poco": tanto quanto costa moltiplicare la matrice del sistema per un vettore

Nel caso di Google il costo è proporzionale al numero di uni presente nella matrice delle connessioni

Mediamente ci sono meno di 10 link per pagina, e quindi meno di 10 uni su ogni riga

Nel caso delle immagini il costo è dato dal numero dei pixel per il suo logaritmo

L'algoritmo Page Rank

Equazione

$$x_j = d(h_{1,j}\frac{x_1}{r_1} + h_{2,j}\frac{x_2}{r_2} + \cdots + h_{n,j}\frac{x_n}{r_n}) + \frac{1}{n}(1-d),$$

Algoritmo

1. Assegna agli x_i dei valori qualsiasi
2. sostituiscili nella parte destra della formula

$$y_j = d(h_{1,j}\frac{x_1}{r_1} + h_{2,j}\frac{x_2}{r_2} + \cdots + h_{n,j}\frac{x_n}{r_n}) + \frac{1}{n}(1-d),$$

3. e ricava i valori di y_j, per $j = 1, 2, \ldots, n$
4. poni $x_j = y_j$ per $j = 1, 2, \ldots, n$ e prosegui dal punto 2

L'algoritmo Page Rank

Equazione

$$x_j = d(h_{1,j}\frac{x_1}{r_1} + h_{2,j}\frac{x_2}{r_2} + \cdots + h_{n,j}\frac{x_n}{r_n}) + \frac{1}{n}(1-d),$$

Algoritmo

1. Assegna agli x_i dei valori qualsiasi
2. sostituiscili nella parte destra della formula

$$y_j = d(h_{1,j}\frac{x_1}{r_1} + h_{2,j}\frac{x_2}{r_2} + \cdots + h_{n,j}\frac{x_n}{r_n}) + \frac{1}{n}(1-d),$$

3. e ricava i valori di y_j, per $j = 1, 2, \ldots, n$
4. poni $x_j = y_j$ per $j = 1, 2, \ldots, n$ e prosegui dal punto 2

L'algoritmo Page Rank

Equazione

$$x_j = d(h_{1,j}\frac{x_1}{r_1} + h_{2,j}\frac{x_2}{r_2} + \cdots + h_{n,j}\frac{x_n}{r_n}) + \frac{1}{n}(1-d),$$

Algoritmo

1. Assegna agli x_i dei valori qualsiasi
2. sostituiscili nella parte destra della formula

$$y_j = d(h_{1,j}\frac{x_1}{r_1} + h_{2,j}\frac{x_2}{r_2} + \cdots + h_{n,j}\frac{x_n}{r_n}) + \frac{1}{n}(1-d),$$

3. e ricava i valori di y_j, per $j = 1, 2, \ldots, n$
4. poni $x_j = y_j$ per $j = 1, 2, \ldots, n$ e prosegui dal punto 2

L'algoritmo Page Rank

Equazione

$$x_j = d(h_{1,j}\frac{x_1}{r_1} + h_{2,j}\frac{x_2}{r_2} + \cdots + h_{n,j}\frac{x_n}{r_n}) + \frac{1}{n}(1-d),$$

Algoritmo

1. Assegna agli x_i dei valori qualsiasi
2. sostituiscili nella parte destra della formula

$$y_j = d(h_{1,j}\frac{x_1}{r_1} + h_{2,j}\frac{x_2}{r_2} + \cdots + h_{n,j}\frac{x_n}{r_n}) + \frac{1}{n}(1-d),$$

3. e ricava i valori di y_j, per $j = 1, 2, \ldots, n$
4. poni $x_j = y_j$ per $j = 1, 2, \ldots, n$ e prosegui dal punto 2

Viene generata una successione di approssimazioni che converge alla soluzione del sistema **qualunque** siano le approssimazioni iniziali $x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}$.

$$x_j^{(1)},\ x_j^{(2)},\ x_j^{(3)},\ \ldots \quad \to x_j, \qquad \text{per} \quad j = 1, 2, \ldots, n$$

Quanto veloce è la convergenza?

L'errore di approssimazione $e^{(k)} = \max_i |x_i^{(k)} - x_i|$ è tale che

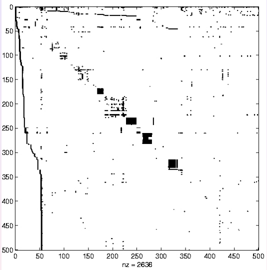
$$e^{(k)} \le \lambda^k \quad \text{con} \quad 0 < \lambda < 1$$

Purtroppo per valori di d vicini a 1 il valore di λ è molto vicino ad 1 e quindi la convergenza è lenta.

λ coincide col modulo del secondo autovalore più grande in modulo di una opportuna matrice.

Complessità

Per fare un passo dell'algoritmo del Page Rank bisogna eseguire tante moltiplicazioni quanti sono gli elementi non nulli di H e all'incirca altrettante addizioni
Su ogni riga della matrice H ci sono "pochi" elementi diversi da zero.



Se mediamente ci fossero 50 elementi non nulli su ogni riga, un passo del metodo iterativo eseguito col Blue Gene impiegherebbe 2 millesimi di secondo.

Se anche fossero necessari 1000 passi iterativi basterebbero 2 secondi per approssimare la soluzione di Google.

Note storiche

I primi tentativi documentati di risolvere sistemi di equazioni lineari si trovano nel libro cinese Chiu-chang Suan-shu (nove capitoli sull'aritmetica) che si stima fosse stato scritto intorno al 200 AC.

All'inizio dell'ottavo capitolo è descritto il problema:

tre fasci di raccolto di buona qualità, due di media qualità e uno di cattiva qualità sono venduti a 39 dou

due fasci di buona qualità, tre di media e uno di cattiva qualità sono venduti a 34 dou

un fascio di buona qualità, due di media e tre di cattiva qualità sono venduti a 26 dou

Quali sono i costi di ciascun fascio di buona, media e cattiva qualità?

In chiave moderna il problema si formula con il sistema:

$$\begin{aligned} 3x + 2y + z &= 39, \\ 2x + 3y + z &= 34, \\ x + 2y + 3z &= 26, \end{aligned}$$

Nel libro il problema viene affrontato mettendo in un tavoliere dei bastoncini colorati di bambù che rappresentano i coefficienti; le righe del tavoliere vengono poi manipolate secondo delle regole opportune

Documentazioni successive si hanno dopo quasi due millenni, quando il matematico giapponese Seki Kowa (1642-1708) migliorò la tecnica cinese introducendo il concetto attualmente noto come **determinante**.

Circa nello stesso periodo il matematico tedesco Gottfried W. Leibniz (1646-1716) sviluppava in modo indipendente il suo concetto di determinante.

Sembra che sia nel lavoro di Kowa che in quello di Leibniz fosse contenuta quella che poi venne chiamata la **regola di Cramer** per risolvere sistemi scoperta da Gabriel Cramer (1704–1752).

Tra il 1750 e il 1900 fu scritto molto sul concetto di determinante, esso diventò lo strumento più importante per risolvere sistemi lineari.

Ma è solo col lavoro del matematico inglese Arthur Cayley (1821-1895) che il concetto di **matrice** fu introdotto come entità a sé stante distinto dal concetto di determinante, assieme alle operazioni algebriche fra matrici. Il lavoro *A Memoir on the Theory of Matrices* segna la nascita della **teoria delle matrici** e dell'**algebra lineare**.

Grandi contributi furono dati dal matematico inglese James J. Sylvester (1814-1897) che aveva introdotto il concetto di **rango**, termine coniato dal matematico tedesco Ferdinand Georg Frobenius (1849-1917).

Frobenius fu il primo a dare la dimostrazione del teorema di **Cayley-Hamilton**. Egli diede anche l'estensione del teorema di Perron (noto appunto come torema di **Perron-Frobenius**) sulle matrici non negative.

Dopo che la teoria delle matrici si consolidò verso la fine del XIX secolo, prese campo la consapevolezza che differenti entità matematiche di natura diversa avevano forti analogie con le matrici, in particolare erano accomunate dalle stesse proprietà algebriche di composizione.

Lo studio delle proprietà comuni di entità diverse ha portato alla formalizzazione assiomatica del concetto di **spazio vettoriale**.

L'idea di una definizione assiomatica degli spazi vettoriali, è contenuta in una teoria generale pubblicata nel 1844 da Hermann Grassmann (1808-1887),

Giuseppe Peano (1858-1932) nel 1888 pubblicò una interpretazione sintetica del lavoro di Grassmann in cui dava una definizione assiomatica di spazio vettoriale simile a quella moderna.

La definizione attuale di spazio vettoriale è quella data dal matematico tedesco Hermann Weyl (1885-1955).

Gli aspetti computazionali dell'algebra lineare ebbero inizio essenzialmente con John von Neumann (1903–1957), matematico ungherese emigrato in America negli anni '40 e uno dei 6 professori di matematica alla prestigiosa università di Princeton.

L'avvento dei calcolatori diede un grande impulso alla ricerca di metodi di risoluzione efficienti

Nel 1943 Harold Hotelling (1895–1973) dimostrò che il metodo di Gauss era inaffidabile se usato con una aritmetica approssimata come è quella dei computer a causa della grande amplificazione degli errori.

Nel 1947 von Neumann e Goldstein dimostrarono che il metodo di Gauss era affidabile se applicato a matrici definite positive

Fu James H. Wilkinson (1919–1986) a fare per primo una analisi sistematica e rigorosa degli errori e a dimostrare che il metodo di Gauss, accompagnato da opportune strategie di *pivoting* è comunque affidabile ed efficiente.

Gli aspetti computazionali dell'algebra lineare hanno avuto un grandissimo sviluppo a partire dal secolo scorso. Un ruolo fondamentale in questo senso l'ha avuto il matematico americano Gene H. Golub (1932–2007) che può essere definito il padre fondatore dell'**Algebra lineare numerica**.

Situazione attuale

La ricerca scientifica in algebra lineare è intensa e vivace. Molti gruppi sono in forte competizione. Nuovi metodi, nuovi problemi e nuovi teoremi vengono introdotti in continuazione.

C'è un grande fermento e attività internazionale con congressi che si svolgono più volte all'anno in ogni parte del mondo e con una forte attività editoriale: i risultati delle ricerche sono pubblicati su diverse riviste internazionali dedicate all'argomento e su libri di carattere avanzato.

I problemi del mondo reale richiedono in continuazione metodi di risoluzione sempre più efficienti e strumenti di analisi sempre più sofisticati

La matematica diventa sempre più sofisticata e fornisce strumenti sempre più potenti per le applicazioni

Le applicazioni si estendono sempre di più a nuovi campi

È questa una forte sinergia che alimenta il lavoro dei matematici e produce avanzamenti importanti nella società tecnologica e industriale con conseguenze sociali di grossa rilevanza.