Algebră liniară numerică

Ionel-Dumitrel Ghiba

dumitrel.ghiba@uaic.ro
https://www.math.uaic.ro/ ghiba/teaching.html

ALEXANDRU IOAN CUZA
UNIVERSITY of IAŞI

Table of contents i

Table of contents ii

Table of contents iii

Bibliografie

Întreaga prezentare se bazează și conține rezultate și calcule din

- MATLAB Lucian Maticiuc, Introducere în MATLAB, Iași, 2019, disponibil on-line la adresa https://www.math.uaic.ro/ maticiuc/didactic/MATLAB_Course.pdf
- Quarteroni, Alfio, Riccardo Sacco, and Fausto Saleri. Numerical mathematics. Vol. 37. Springer Science & Business Media, 2010.
- Gander, Walter, Martin J. Gander, and Felix Kwok. Scientific computing-An introduction using Maple and MATLAB. Vol. 11. Springer Science & Business, 2014.
- Amir Beck, Introduction to nonlinear optimization-Theory, Algorithms, and Applications with MATLAB, SIAM, 2014.

Detalii privind evaluarea

Detalii privind evaluarea

- Evaluare continuă (EC) (N1) (30% din nota finală):
  - teme din patru în patru săptămâni (termen de depunere a temei=patru săptămâni, nota la teme se decide după ce rezolvările sunt explicate în ultima săptămână) (50% din evaluarea continuă),
  - activitatea la seminar (50% din evaluarea continuă);
- Examen final mixt
  - rezolvarea a două exerciții/scriere de programe din care unul foarte asemănător cu cele din fișele de lucru pentru laboratoare și teme și explicarea rezolvării lor (N2) (40% din nota finală);
  - explicarea noțiunilor teoretice prin extragerea a două bilete (N3) (30% din nota finală).

Scopul cursului

Scop general

Ne propunem să găsim algoritmi numerici pentru rezolvarea unui sistem de n ecuații cu n necunoscute, adică

$$\sum_{j=1}^{n} a_{ij} x_j = b_j, \qquad i = 1, 2, ..., m$$

unde $a_{ij} \in \mathbb{R}, b_j \in \mathbb{R}$, $i = 1, 2, ..., m, j = 1, 2, ..., n$.

Definind matricea A având componentele a_{ij}, $i = 1, 2, ..., m, j = 1, 2, ..., n$ (matricea sistemului), vectorul coloană x de componente x_j (necunoscuta) și vectorul coloană b de componente b_j (termenul liber) , sistemul poate fi scris în forma matriceală

$$A\,x = b.$$

A determina soluția înseamnă a determina vectorul $x \in \mathbb{R}^n$ care verifică sistemul de mai sus.

Prezentăm diverse strategii de rezolvare împreună, pe cât posibil, cu aplicații ale lor în practică, însă scopul principal este de argumenta metodele din punct de vedere matematic.

Scop inițial

Ne propunem să găsim algoritmi numerici pentru rezolvarea unui sistem de n ecuații cu n necunoscute, adică

$$\sum_{j=1}^{n} a_{ij} x_j = b_j, \qquad i = 1, 2, ..., n,$$

unde $a_{ij} \in \mathbb{R}, b_j \in \mathbb{R}$, $i, j = 1, 2, ..., n$.

Definind matricea A având componentele a_{ij}, $i, j = 1, 2, ..., n$ (matricea sistemului), vectorul coloană x de componente x_j (necunoscuta) și vectorul coloană b de componente b_j (termenul liber) , sistemul poate fi scris în forma matriceală

$$A\,x = b.$$

Vom presupune pentru început că sistemul este unic determinar, adică $\det A \neq 0$.

Ne reamintim că sistemul admite soluție unică dacă una dintre următoarele condiții este verificată:

- A este inversabilă (atunci $x = A^{-1}b$);
- $\operatorname{rank} A = n$.

Dacă sistemul este omogen ($b = 0$), atunci admite soluția nulă

$$x = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^n.$$

Regula lui Cramer

Dacă A este inversabilă atunci regula lui Cramer ne conduce la soluție

$$x_j = \frac{\Delta_j}{\det A}, \quad j = 1, 2, ..., n,$$

unde Δ_j este determinantul matricei obținute din A prin înlocuirea coloanei j cu coloane termenilor liberi b.

Totuși, această formulă nu este prea indicată în practică pentru că dacă folosim regula lui Laplace pentru calculul determinanților atunci regula lui Cramer necesită $(n+1)!$ operații.

Având în vedere că în practică sistemele sunt mari, aceasta înseamnă timp mare de lucru.

Metode numerice

- DIRECTE (număr finit de pași): se construiește soluția într-un număr finit de pași (calcule cu ajutorul liniilor), folosind factorizări $A = L\,U$, $A = L\,D\,M^T$, ...
- INDIRECTE: se construiește un șir $(x_k) \subset \mathbb{R}^n$ care să conveargă la soluția sistemului. Teoretic am un număr infinit de pași, dar de fapt ne oprim când x_k este "suficient de aproape" de x.

Sisteme mari!!! Ne mai ajută metodele învățate?

Dar ce facem când avem sisteme foarte mari și vrem să găsim o soluție?

Aplicăm teorema Kronecker-Capelli și facem calcule pe hârtie calculând, de exemplu determinanți de matrice 1000×1000?

Matrice-Recapitulare

Matrice pozitiv definite

Pozitiva definire a unei matrice

- Spunem că matricea $A \in \mathbb{R}^{n \times n}$ este simetrică dacă $A = A^T$,
- Spunem că matricea simetrică $A \in \mathbb{R}^{n \times n}$ este **pozitiv semidefinită**, notăm $A \succeq 0$, dacă $\langle Ax, x \rangle \geq 0$ pentru orice $x \in \mathbb{R}^n$, unde $\langle \cdot, \cdot \rangle$ reprezintă produsul scalar standard din $\mathbb{R}^n$.
- Spunem că matricea simetrică $A \in \mathbb{R}^{n \times n}$ este **pozitiv definită**, notăm $A \succ 0$, dacă $\langle Ax, x \rangle > 0$ pentru orice $x \in \mathbb{R}^n$, $x \neq 0$.

Valori proprii și vectori proprii

Fie $A \in \mathbb{R}^{n \times n}$. Un vector nenul $v \in \mathbb{C}z^n$ se numește vector propriu pentru A dacă există $\lambda \in \mathbb{C}$ astfel încât

$$A\,v = \lambda\,v.$$

Scalarul λ se numește valoarea proprie corespunzătoare vectorului propriu v. În general, matricele reale pot avea valori proprii complexe, dar matricele simetrice reale admit doar valori proprii reale. Demonstrați!

Valorile proprii ale unei matrice simetrice $A \in \mathbb{R}^{n \times n}$ vor fi notate cu

$$\lambda_1(A) \geq \lambda_2(A) \geq ... \geq \lambda_n(A).$$

Cea mai mare valoare proprie va fi notată cu $\lambda_{\max}(A) = \lambda_1(A)$ și cea mai mică valoare proprie va fi notată cu $\lambda_{\min}(A) = \lambda_n(A)$.

Pozitiva definire și valori proprii

Fie A o matrice simetrică din $\mathbb{R}^{n\times n}$. Atunci

- A este **pozitiv semidefinită** dacă și numai dacă valorile sale proprii sunt mai mari sau egale cu 0.
- A este **pozitiv definită** dacă și numai dacă valorile sale proprii sunt strict mai mari decât 0.

Criteriul minorilor principali

Fie A o matrice simetrică din $\mathbb{R}^{n\times n}$. Atunci A este **pozitiv definită** dacă și numai dacă minorii principali $\det A(1:i, 1:i)$, $i = 1, 2, \cdots, n$, sunt strict pozitivi.

O matrice $A \in \mathbb{R}^{n\times n}$ se numește diagonal dominantă pe linii dacă

$$|a_{ii}| \geq \sum_{j=1, j\neq i}^{n} |a_{ij}|, \quad \text{with} \quad i = 1, \cdots, n. \tag{1}$$

O matrice $A \in \mathbb{R}^{n\times n}$ se numește diagonal dominantă pe coloane dacă

$$|a_{ii}| \geq \sum_{j=1, j\neq i}^{n} |a_{ji}|, \quad \text{with} \quad i = 1, \cdots, n, \tag{2}$$

Dacă inegalitățile sunt stricte, spunem că A este strict diagonal dominantă (pe linii, respectiv, pe coloane).

Teoremă
O matrice simetrică strict diagonal dominantă cu elemente strict pozitive pe diagonală este pozitiv definită.

Este reciproca valabilă?

Metode directe de rezolvare a sistemelor algebrice liniare unic determinate

Rezolvarea sistemelor inferior triunghiulare

Pentru exemplificare să considerăm sistemul

$$\begin{pmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}, \tag{3}$$

unde $l_{ii} \neq 0,\ i = 1, 2, 3.$

Ultima condiție ne asigură că matricea sistemului este inversabilă, soluția fiind dată de

$$\begin{aligned} x_1 &= \frac{b_1}{l_{11}}, \\ x_2 &= \frac{b_2 - l_{21}x_1}{l_{22}}, \\ x_3 &= \frac{b_3 - l_{31}x_1 - l_{32}x_2}{l_{33}}. \end{aligned}$$

Acest algoritm se numește metoda substituțiilor succesice (forward).

În general pentru $n \geq 2$ avem

$$x_1 = \frac{b_1}{l_{11}},$$

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} l_{ij} x_j}{l_{ii}}, \qquad i = 2, ..., n. \tag{4}$$

Câte operații trebuiesc făcute?

În general pentru $n \geq 2$ avem

$$x_1 = \frac{b_1}{l_{11}},$$

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} l_{ij} x_j}{l_{ii}}, \qquad i = 2, ..., n. \tag{5}$$

Câte operații trebuiesc făcute?

$\frac{n(n+1)}{2}$ înmulțiri și $\frac{n(n-1)}{2}$ adunări și scăderi $= n^2$ operații.

În general pentru $n \geq 2$ avem

$$x_1 = \frac{b_1}{l_{11}},$$

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} l_{ij} x_j}{l_{ii}}, \qquad i = 2, ..., n. \tag{6}$$

Câte operații trebuiesc făcute?

$\frac{n(n+1)}{2}$ înmulțiri și $\frac{n(n-1)}{2}$ adunări și scăderi $= n^2$ operații.

Comparați cu $(n+1)!$ de la regula lui Cramer.

Rezolvarea sistemelor superior triunghiulare

Pentru exemplificare să considerrăm sistemul

$$\begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}, \tag{7}$$

unde $u_{ii} \neq 0$, $i = 1, 2, 3$.

Ultima condiție ne asigură că matricea sistemului este inversabilă, soluția în cazul general fiind dată de

$$\begin{aligned} x_n &= \frac{b_n}{u_{nn}}, \\ x_i &= \frac{b_i - \sum_{j=i+1}^{n} u_{ij} x_j}{u_{ii}}, \qquad i = n-1, ..., 1. \end{aligned} \tag{8}$$

Acest algoritm se numește metoda substituțiilor backward.

Avem tot n^2 operații.

Pentru implementare

Pentru implementare ar fi indicat să stocăm doar elementele nenule atunci când avem de rezolvat sistemele triunghiulare.

Eliminare gaussiană și factorizarea LU

Eliminare gaussianã

Eliminare gaussianã ne ajutã sã reducem un sistem

$$A\,x = b$$

la un sistem (sau douã sisteme) triunghiular prin transformãri succesive ale sistemului în sisteme echivalente

$$A^{(1)}\,x = b^{(1)} \to A^{(2)}\,x = b^{(2)} \to \cdots \to A^{(k)}\,x = b^{(k)}.$$

Presupunem cã la fiecare pas elementul $a_{kk}^{(k)}$ al matricei $A^{(k)}$ este nenul.

Acest element va fi numit pivot.

Presupunem cã A este inversabilã, adicã sistemul admite soluție unicã.

Pasul 1

Plecăm de la sistemul

$$\underbrace{\begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & \cdots & a_{1n}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & \cdots & a_{2n}^{(1)} \\ a_{31}^{(1)} & a_{32}^{(1)} & \cdots & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \cdots & \cdots & a_{nn}^{(1)} \end{pmatrix}}_{\equiv A} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1^{(1)} \\ b_2^{(1)} \\ \vdots \\ b_n^{(1)} \end{pmatrix}$$

Definim multiplicatorii $m_{i1} = \frac{a_{i\mathbf{1}}^{(\mathbf{1})}}{a_{\mathbf{11}}^{(\mathbf{1})}}, \quad i = 2, 3, ..., n$, înmulțim prima linie cu m_{i1}, pe rând, și scădem rezultatele din linia $i = 2, 3, ..., n$, respectiv.

Pasul 1

Se obține astfel un nou sistem, echivalent cu cel inițial

$$\underbrace{\begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & \cdots & a_{2n}^{(2)} \\ 0 & a_{32}^{(2)} & \cdots & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & \cdots & a_{nn}^{(2)} \end{pmatrix}}_{:=A^{(2)}} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1^{(2)} \\ b_2^{(2)} \\ \vdots \\ b_n^{(2)} \end{pmatrix}.$$

Pasul k

Repetând procedeul, la pasul k se obține astfel un nou sistem, echivalent cu cel inițial

$$\underbrace{\begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & \cdots & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & \cdots & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nk}^{(k)} & \cdots & a_{nn}^{(k)} \end{pmatrix}}_{:=A^{(k)}} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \underbrace{\begin{pmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_k^{(k)} \\ \vdots \\ b_n^{(k)} \end{pmatrix}}_{:=b^{(k)}}.$$

Pasul $n-1$

După $n-1$ pași se obține astfel un nou sistem, echivalent cu cel inițial dar **superior triunghiular**

$$\underbrace{\begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & \cdots & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & \cdots & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \cdots & a_{nn}^{(n)} \end{pmatrix}}_{:=A^{(n)}} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \underbrace{\begin{pmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \vdots \\ b_k^{(k)} \\ \vdots \\ b_n^{(n)} \end{pmatrix}}_{:=b^{(n)}}.$$

Algoritm

În concluzie, formulele după care se modifică sistemul de la pasul k în sistemul de la pasul $k+1$ sunt

$$
\begin{aligned}
m_{ik} &= \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, && i = k+1, ..., n, \\
a_{ij}^{(k+1)} &= a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}, && i, j = k+1, ..., n, \\
b_{i}^{(k+1)} &= b_{i}^{(k)} - m_{ik} b_{k}^{(k)}, && i, j = k+1, ..., n.
\end{aligned} \tag{9}
$$

Numărul de operații

- Aplicând GEM avem $\frac{2(n-1)n(n+1)}{3} + n(n-1)$ operații pentru a aduce sistemul la o formă triunghiulară.
- Se adaugă n^2 operații pentru rezolvarea sistemului superior triunghiular.
- În total $\frac{2n^3}{3} + 2\,n^2$ operații.

GEM funcționează dacă $a_{kk}^{(k)} \neq 0$, $k = 1, 2, ..., n-1$.

Din păcate, plecând cu o matrice nenulă pe diagonală nu avem certitudinea că la un pas ulterior k nu vom avea $a_{kk}^{(k)} \neq 0$.

De exemplu, considerând matricea

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 7 & 8 & 9 \end{pmatrix}$$

după primul pas găsim

$$A^{(2)} = \begin{pmatrix} 1 & 2 & 3 \\ 0 & 0 & -1 \\ 0 & -6 & -12 \end{pmatrix}.$$

Ce e de făcut?

Ce e de făcut?

Nu mai rezolvăm sisteme sau mergem cu un algoritm care e posibil să nu funcționeze?

Nu. Constientizăm problema și acoperim toate cazurile construind noi strategii.

Ce e de făcut?

Nu mai rezolvăm sisteme sau mergem cu un algoritm care e posibil să nu funcționeze?

Nu. Constientizăm problema și acoperim toate cazurile construind noi strategii.

Regândim teoretic problema fără a recurge la “peticiri" de moment.

Anticipăm

Poate putem ști de la bun început dacă pentru o matrice este potrivit sau nu să folosim GEM?

Într-adevăr sunt criterii care ne asigură că putem folosi GEM, de exemplu

- Matrice dominate pe linii sau coloane.
- Matrice pozitiv definite.

Însă toate acestea trebuiesc cercetate și argumentate, bănuilile nefiind justificări.

În continuare urmărim să rescriem matricea $A \in \mathbb{R}^{n \times n}$ sub forma $A = L\,U$, unde L este inferior triunghiulară iar U este superior triunghiulară.

Facem acest lucru deoarece după ce vom reuși, sistemul inițial va putea fi rescris sub forma a două sisteme triunghiulare, adică

$$A\,x = b \qquad \leftrightarrow \qquad L\,U\,x = b \qquad \leftrightarrow \qquad \begin{cases} L\,y = b \\ U\,x = y \end{cases} \tag{10}$$

ce se vor rezolva pe rând.

GEM prin multiplicare de matrice

Să remarcăm că operațiile pe care le-am făcut asupra primei coloane se rezumă la a înmulți matricea $A^{(1)} := A$, la stânga, cu matricea

$$M_1 = \begin{pmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ -m_{21} & 1 & 0 & \cdots & \ddots & 0 \\ -m_{31} & 0 & 1 & \cdots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \cdots & 0 \\ -m_{n1} & 0 & 0 & \cdots & \cdots & 1 \end{pmatrix}$$

GEM prin multiplicare de matrice

Adică

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ -m_{21} & 1 & 0 & \cdots & \ddots & 0 \\ -m_{31} & 0 & 1 & \cdots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \cdots & 0 \\ -m_{n1} & 0 & 0 & \cdots & \cdots & 1 \end{pmatrix}}_{=M_1} A^{(1)} := A^{(2)}.$$

GEM prin multiplicare de matrice

Adică

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ -m_{21} & 1 & 0 & \cdots & \ddots & 0 \\ -m_{31} & 0 & 1 & \cdots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \cdots & 0 \\ -m_{n1} & 0 & 0 & \cdots & \cdots & 1 \end{pmatrix}}_{=M_1} \underbrace{\begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & \cdots & a_{1n}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & \cdots & a_{2n}^{(1)} \\ a_{31}^{(1)} & a_{32}^{(1)} & \cdots & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \cdots & \cdots & a_{nn}^{(1)} \end{pmatrix}}_{\equiv A} := A^{(2)}.$$

GEM prin multiplicare de matrice

Adică

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ -\frac{a_{21}^{(1)}}{a_{11}^{(1)}} & 1 & 0 & \cdots & \ddots & 0 \\ -\frac{a_{31}^{(1)}}{a_{11}^{(1)}} & 0 & 1 & \cdots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \cdots & 0 \\ -\frac{a_{n1}^{(1)}}{a_{11}^{(1)}} & 0 & 0 & \cdots & \cdots & 1 \end{pmatrix}}_{=M_1} \underbrace{\begin{pmatrix} a_{11}^{(1)} & a_{12}^{(1)} & \cdots & \cdots & a_{1n}^{(1)} \\ a_{21}^{(1)} & a_{22}^{(1)} & \cdots & \cdots & a_{2n}^{(1)} \\ a_{31}^{(1)} & a_{32}^{(1)} & \cdots & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ a_{n1}^{(1)} & a_{n2}^{(1)} & \cdots & \cdots & a_{nn}^{(1)} \end{pmatrix}}_{\equiv A} := A^{(2)}.$$

GEM prin multiplicare de matrice

Apoi

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ 0 & 1 & 0 & \cdots & \ddots & 0 \\ 0 & -m_{32} & 1 & \cdots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \cdots & 0 \\ 0 & -m_{n2} & 0 & \cdots & \cdots & 1 \end{pmatrix}}_{=M_2} A^{(2)} := A^{(3)}$$

și așa mai departe repetând procedeul de $n-1$ ori până se ajunge la

$$M_{n-1}M_{n-2}\cdots M_2 M_1 A = A^{(n)} =: U \quad \text{matrice superior triunghiulară.}$$

Eliminare gaussiană

Considerăm o matrice $A \in \mathbb{R}^{n\times n}$. Scopul este de a construi o secvență $A^{(k)} = (a_{ij}^{(k)})$ de matrici prin efectuarea de transformări liniare, astfel încât să ajungem la o matrice superior triunghiulară $U = (u_{ij})$ după câțiva pași finiți.

În ultima săptămână am văzut că dacă presupunem că pivoții $a_{11} \neq 0$, $a_{kk}^{(k)} = (M_{k-1}...M_1\, A)_{kk} \neq 0$ la orice pas $k = 2, ..., n-1$, atunci

$$M_{n-1}M_{n-2}...M_1\, A = U, \tag{11}$$

cu U o matrice superior triunghiulară, unde

$$M_k = \begin{pmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & -m_{k+1,k} & 1 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -m_{n,k} & 0 & \cdots & 1 \end{pmatrix} = I_n - m_k\, e_k^T, \tag{12}$$

și

$$m_k = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 0 \\ m_{k+1,k} \\ \vdots \\ m_{n,k} \end{pmatrix} \in \mathbb{R}^n, \qquad e_k = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^n, \quad m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, i = k+1, \cdots, n. \tag{13}$$

Mai mult, deoarece $M_k^{-1} = I_n + m_k\, e_k^T$ (Exercițiu), deducem (cum? Detaliază calculele!)

$$A = (I_n + \sum_{i=1}^{n-1} m_i e_i^T)\, U = \underbrace{\begin{pmatrix} 1 & 0 & \cdots & 0 & \cdots & 0 \\ m_{21} & 1 & 0 & \cdots & \cdots & \vdots \\ m_{31} & m_{32} & 1 & 0 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & 0 \\ m_{n1} & m_{n2} & \cdots & m_{n,n-1} & & 1 \end{pmatrix}}_{:=L} U \tag{14}$$

Deci dacă nu întâlnim pivoți nuli ($a_{kk}^{(k)}$) atunci putem construi factorizarea LU a acelei matrice.

Dar cum știm dacă vom întâlni pivoți nuli fără a începe procesul de construcție al factorizării?

Existența și unicitatea factorizării $L\,U$

Teoremă
Fie $A \in \mathbb{R}^{n\times n}$. Există factorizarea LU a matricei A cu $l_{ii} = 1$ pentru orice $i = 1, \cdots, n$ și este unică dacă și numai dacă submatricele principale $A_i = A(1:i, 1:i)$ ale lui de orice ordin $i = 1, \cdots, n-1$ sunt nesingulare.

Vom demonstra mai întâi implicația "$\Longleftarrow$". Vom demonstra faptul că dacă submatricea principală A_{i-1} admite descumpunere $L\,U$, atunci și A_i admite descompunere $L\,U$.

Pentru $i = 1$: $A_1 = a_{11} = \underbrace{1}_{:=L} \cdot \underbrace{a_{11}}_{:=U}$.

Presupunem că există descompunerea LU pentru A_{i-1}, adică există matricea inferior triunghiulară L_{i-1} având elementele de pe diagonală egale cu 1 și matricea superior triunghiulară U_{i-1} astfel încât

$$A_{i-1} = L_{i-1}\,U_{i-1}.$$

Construim L_i și U_i astfel încât $A_i = L_i\,U_i$.

Construim L_i și U_i astfel încât $A_i = L_i\, U_i$.

Pentru aceasta dorim să determinăm vectorii ℓ și u și scalarul u_{ii} pentru care

$$\begin{pmatrix} A_{i-1} & c \\ d^T & a_{ii} \end{pmatrix} = A_i = \underbrace{\begin{pmatrix} L_{i-1} & 0 \\ \ell^T & 1 \end{pmatrix}}_{:=L_i} \underbrace{\begin{pmatrix} U_{i-1} & u \\ 0^T & u_{ii} \end{pmatrix}}_{:=U_i}.$$

Trebuie să avem

$$\begin{pmatrix} A_{i-1} & c \\ d^T & a_{ii} \end{pmatrix} = \underbrace{\begin{pmatrix} L_{i-1} & 0 \\ \ell^T & 1 \end{pmatrix}}_{:=L_i} \underbrace{\begin{pmatrix} U_{i-1} & u \\ 0^T & u_{ii} \end{pmatrix}}_{:=U_i} = \begin{pmatrix} L_{i-1}\, U_{i-1} & L_{i-1}\, u \\ \ell^T\, U_{i-1} & \ell^T\, u + u_{ii}. \end{pmatrix}$$

$$
\begin{aligned}
L_{i-1}\, u &= c, \\
U_{i-1}^{T} \ell &= d, \\
\ell^{T} u &= a_{ii} - u_{ii}.
\end{aligned}
\tag{15}
$$

Deorece A_{i-1} sunt nesigulare, vom avea că U_{i-1} sunt nesingulare. Matricele L_{i-1} sunt nesingulare, având determinantul egal cu 1.

Prin urmare există u, ℓ și u_{ii} care verifică sistemul de mai sus și care construiesc matricea L_i.

Să demonstrăm implicația inversă "$\Longrightarrow$".

Avem de demonstrat: Dacă există factorizare LU cu $l_{ii} = 1$ și este unică, atunci primele $n-1$ submatrici principale ale lui A sunt inversabile.

Vom împărți dicuția pe două cazuri.

Cazul 1. A este inversabilă, $\det A \neq 0$:

Presupunem că există factorizare LU cu $l_{ii} = 1$ și este unică

Să remarcăm faptul că din forma factorizării rezultă că $a_{11} \neq 0$.

Deoarece factorizare LU există pentru A, va exista și pentru A_i și $\det A_i = \det L^{(i)} \det U^{(i)} = \det U^{(i)} = u_{11} u_{22} \cdots u_{ii}, i = 1, n-1.$

Deoarece $\det A_n \neq 0$, rezultă că $u_{11} u_{22} \cdots u_{nn} \neq 0$, adică, în particular, $\det A_i = \det L^{(i)} \det U^{(i)} = \det U^{(i)} = u_{11} u_{22} \cdots u_{ii} \neq 0, i = 1, n-1.$

Cazul 2. A nu este inversabilă, $\det A = 0$: Cu alte cuvinte presupunem că măcar un element de pe diagonala lui U este egal cu zero.

Notăm cu u_{kk} elementul nenul de index minim k (pentru că ar putea să fie și alții, dar îl alegem astfel).

În baza procedeului iterativ descris în prima parte a demonstrației va rezulta că factorizarea poate fi calculată până la pasul $k+1$.

De la acest pas, pentru că matricea $U(k) = U(1:k, 1:k)$ este neinversabilă, existența și unicitatea vectorului ℓ se pierde, și, deci întreaga factorizare LU pentru $A(k+1) = U(1:k+1, 1:k+1)$ și pentru matricea A.

Pentru ca acest fapt să nu se întâmple, elementul nul u_{kk} ar trebui să fie de index $k = n-1$. Deoarece
$\det A_i = \det L^{(i)} \det U^{(i)} = \det U^{(i)} = u_{11} u_{22} \cdots u_{ii}$, $i = 1, n-1$, toate matricele principale A_k vor fi inversabile $k = 1, \cdots, n-1$

Example
Considerăm matricele

$$B = \begin{pmatrix} 1 & 2 \\ 1 & 2 \end{pmatrix}, \qquad C = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \qquad D = \begin{pmatrix} 0 & 1 \\ 0 & 2 \end{pmatrix}. \tag{16}$$

- B admite o unică factorizare LU.
- matricea neinversabilă C nu admite factorizare LU.
- matricea neinversabilă D admite o infinitate de factorizări de forma $D = L_\beta U_\beta$, cu

$$L_\beta = \begin{pmatrix} 1 & 0 \\ \beta & 1 \end{pmatrix}, \quad U_\beta = \begin{pmatrix} 0 & 1 \\ 0 & 2-\beta \end{pmatrix} \quad \forall\, \beta \in \mathbb{R}.$$

Există un alt rezultat:

Teoremă
Dacă A este o matrice diagonal dominantă (pe linii sau coloane), atunci există și este unică factorizarea LU. În particular, dacă A este diagonal dominantă pe coloane, atunci $|l_{ij}| \leq 1 \; \forall i, j$.

Forma compactă a factorizării

Forma compactă a factorizării

Variantă remarcabilă a factorizării LU este factorizarea **Doolittle**. Aceasta este cunoscută și ca formă compactă a metodei de eliminare Gauss.

Această denumire se datorează faptului că aceste abordări necesită mai puține rezultate intermediare decât metoda GEM standard pentru a genera factorizarea lui A.

Calcularea factorizării LU a lui A este echivalentă din punct de vedere formal cu rezolvarea următorului sistem neliniar de ecuații n^2

$$a_{ij} = \sum_{r=1}^{\min(i,j)} l_{ir} u_{rj}, i, j = 1, ..., n, \tag{17}$$

necunoscutele fiind intrările $n^2 + n$ ale matricelor triunghiulare L și U.

Dacă stabilim în mod arbitrar n coeficienți (l_{ii}) la 1, ajungem la metoda Doolittle care oferă o cale eficientă sistemului neliniar.

De fapt, presupunând că primele $k-1$ coloane din L și primele rânduri din U sunt disponibile și stabilind $l_{kk} = 1$ (metoda Doolittle), se obțin următoarele ecuații din

$$a_{kj} = \sum_{r=1}^{k-1} l_{kr} u_{rj} + u_{kj}, \quad j = k, \cdots, n, \tag{18}$$

$$a_{ik} = \sum_{r=1}^{k-1} l_{ir} u_{rk} + l_{ik} u_{kk}, \quad i = k+1, \cdots, n. \tag{19}$$

Rețineți că aceste ecuații pot fi rezolvate într-un mod secvențial în ceea ce privește variabilele roșii u_{kj} și l_{ik}.

Din metoda compactă Doolittle obținem astfel mai întâi al k-lea rând al lui U și apoi a k-a coloană a lui L, după cum urmează: pentru $k = 1, \cdots, n$

$$u_{kj} = a_{kj} - \sum_{r=1}^{k-1} l_{kr} u_{rj}, \quad j = k, \cdots, n, \tag{20}$$

$$l_{ik} = \frac{1}{u_{rk}} \left(a_{ik} - \sum_{r=1}^{k-1} l_{ir} u_{rk} \right), \quad i = k+1, \cdots, n. \tag{21}$$

Factorizarea LDM^T

Factorizarea LDM^T

Este posibil să se conceapă și alte tipuri de factorizări ale lui A.

Mai exact, vom aborda unele variante în care factorizarea lui A este de forma

$$A = L\,D\,M^T, \tag{22}$$

unde L, M^T și D sunt matrici inferior triunghiulare, superior triunghiulare și, respectiv, diagonale.

După construirea acestei factorizări, rezolvarea sistemului se poate realiza rezolvând mai întâi sistemul inferior triunghiulară $Ly = b$, apoi cel diagonal $Dz = y$ și în final sistemul superior triunghiulară $M^T x = z$, cu un cost de $n^2 + n$ flop-uri.

În cazul simetric, obținem $M = L$, iar factorizarea LDL^T poate fi calculată cu jumătate din cost, după cum vom vedea când vom vorbi despre cazul matricelor simetrice. Factorizarea LDM^T se bucură de o proprietate analogă cu cea pentru factorizarea LU. În particular, se aplică următorul rezultat.

Teoremă
Dacă toți minorii principali ai unei matrice $A \in \mathbb{R}^{n \times n}$ sunt nenuli, atunci există o matrice diagonală unică D, o matrice inferior triunghiulară unitară unică[1] L și o matrice superior triunghiulară unitară unică M^T, astfel încât $A = LDM^T$.

Demonstrație: Știm deja că există o factorizare unică LU a lui A cu $l_{ii} = 1$ pentru $i = 1, \cdots, n$. Dacă stabilim că intrările diagonale ale lui D sunt egale cu u_{ii} (nu sunt zero deoarece U este nesingulară), atunci $A = LU = LD(D^{-1}U)$. După definirea $M^T = D^{-1}U$, rezultă existența factorizării LDM^T, unde $D^{-1}U$ este o matrice superior triunghiulară unitară. Unicitatea factorizării LDM^T este o consecință a unicității factorizării LU.

[1] Noi numim matrice triunghiulară unitară o matrice triunghiulară care are intrările diagonale egale cu 1.

Pivotare

Pivotare

După cum s-a subliniat anterior, procesul GEM se întrerupe imediat ce se calculează o intrare pivotală zero. Într-un astfel de caz, trebuie să se apeleze la așa-numita tehnică de pivotare, care constă în schimbarea rândurilor (sau a coloanelor) din sistem astfel încât să se obțină pivoți nenuli.

Strategia de pivotare adoptată până în prezent poate fi generalizată prin căutarea, la fiecare pas k al procedurii de eliminare, a unei intrări pivotante care nu este nulă, căutând în interiorul intrărilor din subcoloana $A^{(k)}(k:n,k)$. Din acest motiv, se numește pivotare parțială (pe rânduri).

Se poate observa că o valoare mare a lui $m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, i = k+1, \cdots, n$ (generată, de exemplu, de o valoare mică a pivotului $a_{kk}^{(k)}$) poate amplifica erorile de rotunjire care afectează intrările $a_{kj}^{(k)}$.

Prin urmare, pentru a asigura o mai bună stabilitate, pivotul kj $A^{(k)}(j,k)$ se alege ca fiind cea mai mare intrare (în modul) din coloana $A^{(k)}(k:n,k)$ și, în general, se efectuează o pivotare parțială la fiecare etapă a procedurii de eliminare, chiar dacă nu este strict necesar (adică chiar dacă se găsesc intrări pivotale diferite de zero).

Alternativ, procesul de căutare ar fi putut fi extins la întreaga submatrice $A^{(k)}(k:n,k:n)$, finalizându-se cu o pivotare completă.

Observați, totuși, că în timp ce pivotarea parțială necesită un cost suplimentar de aproximativ n^2 căutări, pivotarea completă necesită aproximativ $2n^3/3$, cu o creștere considerabilă a costului de calcul al GEM.

Matrice de permutare

Schimbul dintre a i-a și j-a linie a unei matrice; acest lucru se poate face prin înmulțirea prealabilă a lui A cu matricea $P^{(i,j)}$ de elemente

$$p_{rs}^{(i,j)} = \begin{cases} 1 & \text{dacă } r = s = 1, \cdots, i-1, i+1, \cdots j-1, j+1, \cdots n, \\ 1 & \text{dacă } r = j, s = i \text{ or } r = i, s = j, \\ 0, & \text{în caz contrar.} \end{cases} \tag{23}$$

Matricele de tipul $P^{(i,j)}$ se numesc matrice de permutare elementară.

Produsul matricelor de permutare elementară se numește matrice de permutare și efectuează schimburile de rânduri asociate fiecărei matrice de permutare elementară.

În practică, o matrice de permutare este o reordonare pe rânduri a matricei identitate.

Să analizăm modul în care pivotarea parțială afectează factorizarea LU indusă de GEM.

La prima etapă a GEM cu pivotare parțială, după ce se află intrarea a_{r1} de modul maxim din prima coloană, se construiește matricea elementară de permutare P_1 care schimbă prima linie cu a r-a linie (dacă $r = 1$, P_1 este matricea identitate).

În continuare, se generează prima matrice de transformare gaussiană M_1 și se stabilește

$$A^{(2)} = M_1 P_1 A^{(1)}. \tag{24}$$

O abordare similară se face acum pentru $A^{(2)}$, căutând o nouă matrice de permutare P_2 și o nouă matrice M_2 astfel încât

$$A^{(3)} = M_2 P_2 A^{(2)} = M_2 P_2 M_1 P_1 A^{(1)}. \tag{25}$$

Executând toate etapele de eliminare, matricea superior triunghiulară U rezultată este acum dată de

$$U = A(n) = \underbrace{M_{n-1} P_{n-1} \cdots M_2 P_2 M_1 P_1}_{:=M} A^{(1)}. \tag{26}$$

Rețineți că $m_{ik}^{(k)}$ utilizat în construcția lui M_k este acum $m_{ik}^{(k)} = \frac{b_{ik}^{(k)}}{b_{kk}^{(k)}}$,

unde $b_{ik}^{(k)}$ sunt intrările matricei $P_k A^{(k)}$.

Obținem că $U = M\,A$ și, astfel, $U = (MP^{-1})PA$, unde $P = P_{n-1} \cdots P_1$.
Afirmăm că $L = PM^{-1}$ este inferior triunghiulară unitară.

Afirmăm că $L = PM^{-1}$ este inferior triunghiulară unitară.

Nu trebuie să fim îngrijorați de prezența inversului lui M, deoarece $M^{-1} = P_1^{-1}M_1^{-1}\cdots P_{n-1}^{-1}M_{n-1}^{-1}$ și $P_i^{-1} = P_i^T$, în timp ce $M_i^{-1} = 2I_n - M_i = I_n + m_k\, e_k^T$.

Prin urmare, avem

$$\begin{aligned} L &= P_{n-1}\cdots P_2 P_1 P_1^{-1}(I_n + m_1\, e_1^T)P_2^{-1}(I_n + m_2\, e_2^T)\cdots P_{n-1}^{-1}(I_n + m_{n-1}\, e_{n-1}^T) \\ &= P_{n-1}\cdots P_2(I_n + m_1\, e_1^T)P_2^{-1}(I_n + m_2\, e_2^T)\cdots P_{n-1}^{-1}(I_n + m_{n-1}\, e_{n-1}^T). \end{aligned}$$

Să discutăm acum

$$\begin{aligned} & P_{n-1}\cdots P_2(I_n + m_1\, e_1^T) \\ &= (P_{n-1}\cdots P_2 + P_{n-1}\cdots P_2 m_1\, e_1^T P_2^T\cdots P_{n-1}^T P_{n-1}\cdots P_2) \\ &= (P_{n-1}\cdots P_2 + P_{n-1}\cdots P_2 m_1\, e_1^T (P_{n-1}\cdots P_2)^T P_{n-1}\cdots P_2) \\ &= (P_{n-1}\cdots P_2 + P_{n-1}\cdots P_2 m_1 (P_{n-1}\cdots P_2 e_1)^T P_{n-1}\cdots P_2) \qquad (27) \\ &= [I_n + P_{n-1}\cdots P_2 m_1 (P_{n-1}\cdots P_2 e_1)^T] P_{n-1}\cdots P_2. \end{aligned}$$

Vom avea

$$\begin{aligned} & P_{n-1}\cdots P_2(I_n + m_1\, e_1^T) \\ &= [I_n + \underbrace{P_{n-1}\cdots P_2 m_1}_{:=\widetilde{m}_1}(P_{n-1}\cdots P_2 e_1)^T]P_{n-1}\cdots P_2. \end{aligned} \tag{28}$$

Dar permutarea $P_{n-1}\cdots P_2$ permută doar intrările de la 2 la n dintr-un vector; intrările 1 rămân neatinse. Aceasta înseamnă că primele intrări ale lui $\widetilde{m}_1$ sunt încă zero și e_1 este neschimbată permutarea, adică $P_{n-1}\cdots P_2 e_1 = e_1$.

Astfel, avem de fapt

$$\begin{aligned} & P_{n-1}\cdots P_2(I_n + m_1\, e_1^T) \\ &= \underbrace{[I_n + \widetilde{m}_1 e_1^T]}_{\text{inferior triunghiulară}} \quad P_{n-1}\cdots P_2 \end{aligned} \tag{29}$$

și

$$L = \underbrace{[I_n + \widetilde{m}_1 e_1^T]}_{\text{inferior triunghiulară}} \quad P_{n-1}\cdots P_2 P_2^{-1}(I_n + m_2\, e_2^T)\cdots P_{n-1}^{-1}(I_n + m_{n-1}\, e_{n-1}^T).$$

Repetând argumentul avem că L este inferior triunghiulară.

Deoarece $L = PM^{-1}$ este inferior triunghiulară unitar, factorizarea LU se citește

$$PA = LU. \tag{30}$$

Odată ce L, U și P sunt disponibile, rezolvarea sistemului liniar inițial se rezumă la rezolvarea sistemelor triunghiulare $Ly = Pb$ și $Ux = y$.

Teoremă
Fie $A \in \mathbb{R}^{n \times n}$ o matrice nesingulară. Atunci există o matrice de permutare P astfel încât $PA = LU$, unde L și U sunt matricile triunghiulare inferioară și superioară obținute prin eliminarea gaussiană.

Proof.
Demonstrația este deja făcută, cu excepția faptului că la orice pas avem

$$\max \operatorname{abs}(A^{(k)}(k:n,k)) \neq 0. \tag{31}$$

Dacă ar fi posibil să avem

$$\max \operatorname{abs}(A^{(k)}(k:n,k)) = 0, \tag{32}$$

atunci $\det A = 0$ ceea ce este evitat de ipoteză. □

Dacă se realizează pivotarea completă, la primul pas al procesului, odată găsit elementul a_{qr} al celui mai mare modul din submatricea $A(1:n, 1:n)$, trebuie să schimbăm prima linie și prima coloană cu a q-a linie și a r-a coloană. Se generează astfel matricea $P_1 A^{(1)} Q_1$, unde P_1 și Q_1 sunt matrici de permutare pe rânduri și, respectiv, pe coloane.

În consecință, acțiunea matricei M_1 este acum astfel încât $A^{(2)} = M_1 P_1 A^{(1)} Q_1$. Repetând procesul, la ultima etapă, obținem

$$U = A^{(n)} = M_{n-1} P_{n-1} \cdots M_1 P_1 A^{(1)} Q_1 \cdots Q_{n-1}. \tag{33}$$

În cazul pivotării complete, factorizarea LU devine

$$PAQ = LU, \tag{34}$$

unde $Q = Q = Q_1 \cdots Q_{n-1}$ este o matrice de permutare care ține cont de toate permutările care au fost operate. Prin construcție, matricea L este tot inferior triunghiulară, cu intrări de modul mai mici sau egale cu 1.

Calculul inversei unei matrice

Calculul explicit al inversei unei matrice poate fi efectuat folosind factorizarea LU după cum urmează.

Notând cu X inversa unei matrice nesingulare în $\mathbb{R}^{n\times n}$, vectorii coloană ai lui X sunt soluțiile sistemelor liniare $Ax_i = e_i$, pentru $i = 1, \cdots, n$.

Presupunând că $PA = LU$, unde P este matricea de permutare cu pivotare parțială, trebuie să rezolvăm $2n$ sisteme triunghiulare de forma $Ly_i = Pe_i$, $Ux_i = y_i$, $i = 1, \cdots, n$, adică o succesiune de sisteme liniare care au aceeași matrice de coeficienți, dar părți drepte diferite.

Aplicarea factorizării LU într-o problemă de deformare elastică 1D

Se consideră o bară elastică de lungime $L = 1\,\text{m}$, cu capetele menținute fixe

$$u(0) = u(1) = 0.$$

Ecuația care descrie deformarea barei este de tip Poisson:

$$-k\,u''(x) = q(x), \qquad 0 < x < 1,$$

unde u este deplasarea, k este un coeficient de elasticitiate, iar $q(x)$ forța care acționează pe acea bară.

Aproximarea numerică a derivatei a doua

Pentru fiecare nod interior x_i, a doua derivată $u''(x_i)$ se poate aproxima prin diferențe finite centrale.

Pornim de la seriile Taylor în jurul nodului x_i:

$$u(x_i + h) = u(x_i) + hu'(x_i) + \frac{h^2}{2}u''(x_i) + \frac{h^3}{6}u'''(x_i) + O(h^4),$$

$$u(x_i - h) = u(x_i) - hu'(x_i) + \frac{h^2}{2}u''(x_i) - \frac{h^3}{6}u'''(x_i) + O(h^4).$$

Adunând cele două ecuații, derivata întâi se elimină:

$$u(x_i + h) + u(x_i - h) = 2u(x_i) + h^2 u''(x_i) + O(h^4),$$

de unde rezultă formula de diferențe finite centrale:

$$\boxed{u''(x_i) \approx \frac{u(x_{i-1}) - 2u(x_i) + u(x_{i+1})}{h^2}}.$$

Această formulă stă la baza discretizării ecuației Poisson pentru fiecare nod interior.

Discretizare numerică a ecuației Poisson

Împărțim bara în $N = 4$ segmente egale ($h = L/(N+1) = 0.2$ m) și notăm deformarea în nodurile interioare u_1, u_2, u_3, u_4.

Înlocuind aproximația derivatei a doua în ecuația Poisson:

$$-\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} = \frac{q(x_i)}{k}, \quad i = 1, \ldots, 4,$$

sau echivalent:

$$2u_i - u_{i-1} - u_{i+1} = h^2 \frac{q(x_i)}{k}.$$

Astfel se generează un sistem liniar tridiagonal:

$$A \cdot \mathrm{u} = \mathrm{b},$$

unde

$$A = \begin{bmatrix} 2 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 2 \end{bmatrix}, \qquad \mathrm{b} = \begin{bmatrix} h^2 q(x_1)/k \\ h^2 q(x_2)/k \\ h^2 q(x_3)/k \\ h^2 q(x_4)/k \end{bmatrix}.$$

Factorizarea LU

Observație: valoarea deplasuarii în fiecare nod interior depinde doar de nodul precedent și următor, ceea ce face sistemul tridiagonal și foarte potrivit pentru factorizarea LU. Pentru rezolvarea eficientă, factorăm matricea A ca:

$$A = L \cdot U,$$

unde L este inferior triunghiulară și U superior triunghiulară:

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{1}{2} & 1 & 0 & 0 \\ 0 & -\frac{2}{3} & 1 & 0 \\ 0 & 0 & -\frac{3}{4} & 1 \end{bmatrix}, \qquad U = \begin{bmatrix} 2 & -1 & 0 & 0 \\ 0 & \frac{3}{2} & -1 & 0 \\ 0 & 0 & \frac{4}{3} & -1 \\ 0 & 0 & 0 & \frac{5}{4} \end{bmatrix}.$$

Matrice de tip bandă

Metodele de discretizare pentru problemele cu valori la frontieră conduc adesea la rezolvarea sistemelor liniare cu matrici care au forme de bandă, bloc sau rare. Exploatarea structurii matricei permite o reducere drastică a costurilor de calcul ale factorizării și ale algoritmilor de substituție.

Vom aborda variante speciale ale factorizării GEM sau LU care sunt concepute în mod corespunzător pentru a trata matrici de acest tip.

Factorizarea Cholesky

Matrice simetrice pozitiv definite: Factorizarea Cholesky

După cum s-a arătat deja, factorizarea LDM^T se simplifică considerabil atunci când A este simetrică, deoarece într-un astfel de caz $M = L$, obținându-se așa-numita factorizare LDL^T. Costul de calcul se înjumătățește, față de factorizarea LU, la aproximativ $(n^3/3)$ flop-uri.

Factorizarea Cholesky

Teoremă
Fie $A \in \mathbb{R}^{n \times n}$ o matrice simetrică și pozitiv definită. Atunci, există o matrice superior triunghiulară unică H cu intrări diagonale pozitive astfel încât

$$A = H^T H. \tag{35}$$

Această factorizare se numește factorizare Cholesky.

Proof: Existența.

Deoarece A este pozitiv definită, avem $\det(A(1:k, 1:k)) > 0$, pentru toate $k \in \{1, 2, ..., n\}$.

Printr-un rezultat anterior, rezultă că există $L, U \in \mathbb{R}$ astfel încât $A = LU$, unde L este inferior triunghiulară cu 1 pe diagonală, iar U este superior triunghiulară.

Fie $D = \mathrm{diag}(\sqrt{u_{11}}, ..., \sqrt{u_{nn}})$. Atunci

$$A = LU = \underbrace{(LD)}_{:=B}\underbrace{(D^{-1}U)}_{:=C}, \tag{36}$$

unde B este inferior triunghiulară și C este superior triunghiulară, ambele cu elemente $\sqrt{u_{11}}, ..., \sqrt{u_{nn}}$ pe diagonală.

Acum vom demonstra că $B = C^T$.

Demonstrație: Existență

Din moment ce $A = A^T$, rezultă

$$BC = C^T B^T \Rightarrow (C^T)^{-1} B = B^T C^{-1}. \tag{37}$$

În partea stângă a ultimei egalități, ambele matrici sunt inferior triunghiulare, adică partea stângă este inferior triunghiulară, în timp ce în partea dreaptă a ultimei egalități, ambele matrici sunt superior triunghiulare, adică partea dreaptă este superior triunghiulară.

În plus, partea stângă are 1 pe diagonală, iar partea dreaptă, la fel.

Dar singura matrice care este inferior triunghiulară-superioară cu 1 pe diagonală este matricea identitate I_n.

Așadar, $(C^T)^{-1} B = I_n$ și $C^T = B$, ceea ce încheie dovada existenței factorizării Cholesky.

Demonstrație: Unicitate

Fie C_1, C_2 superior triunghiulare cu elemente diagonale pozitive astfel încât

$$A = C_1^T C_1 = C_2^T C_2. \tag{38}$$

Fie $D_1 = \mathrm{diag}(C_1), D_2 = \mathrm{diag}(C_2)$.

Atunci

$$\underbrace{C_1^T D_1^{-1}}_{\text{inferior triunghiulară cu 1 pe diag}} \qquad \underbrace{D_1 C_1}_{\text{superior triunghiulară}} \qquad = C_2^T D_2^{-1} D_2 C_2. \tag{39}$$

Din unicitatea factorizării LU rezultă că $D_1 C_1 = D_2 C_2$.

Aceasta implică $[(C_1)_{ii}]^2 = [(C_2)_{ii}]^2$, $i = 1, 2, ..., n$, adică $D_1 = D_2$.

Prin urmare, $C_1 = C_2$ și dovada este completă.

Calcularea factorizării Cholesky în practică

Teoremă
Intrările h_{ij} din H^T pot fi calculate după cum urmează:

$$h_{11} = \sqrt{a_{11}} = \sqrt{a_{11}}. \tag{40}$$

și, pentru $i = 2, \cdots, n$,

$$h_{ij} = \left(a_{ij} - \sum_{k=1}^{j-1} h_{ik} h_{jk} \right) / h_{jj}, \quad j = 1, \cdots, i-1, \tag{41}$$

$$h_{ii} = \sqrt{a_{ii} - \sum_{k=1}^{i-1} h_{ik}^2}. \tag{42}$$

Demonstrație:

Să demonstrăm teorema procedând prin inducție asupra mărimii i a matricei, amintind că dacă $A_i \in \mathbb{R}^{i \times i}$ este simetrică pozitiv definită, atunci toate submatricile sale principale se bucură de aceeași proprietate.

Pentru $i = 1$ rezultatul este evident adevărat. Prin urmare, să presupunem că este valabil pentru $i - 1$ și să demonstrăm că este valabil și pentru i. Există o matrice superior triunghiulară H_{i-1} astfel încât $A_{i-1} = H_{i-1}^T H_{i-1}$. Să partiționăm A_i astfel

$$A_i = \begin{pmatrix} A_{i-1} & v \\ v^T & \alpha \end{pmatrix} \tag{43}$$

cu $\alpha \in \mathbb{R}_+$, $v \in \mathbb{R}^{i-1}$ și căutăm o factorizare a lui A_i de forma

$$A_i = H_i^T H_i = \begin{pmatrix} H_{i-1}^T & 0 \\ h^T & \beta \end{pmatrix} \begin{pmatrix} H_{i-1} & h \\ 0^T & \beta \end{pmatrix}. \tag{44}$$

Prin aplicarea egalității cu intrările lui A_i se obțin ecuațiile $H_{i-1}^T h = v$ și $h^T h + \beta^2 = \alpha$.

Astfel, vectorul h este determinat în mod unic, deoarece H_{i-1}^T este nesingulară. În ceea ce privește β, datorită proprietăților determinanților

$$0 < \det(A_i) = \det(H_i^T)\det(H_i) = \beta^2(\det(H_{i-1}))^2, \qquad (45)$$

putem concluziona că trebuie să fie un număr real. Ca urmare, $\beta = \sqrt{\alpha - h^T h}$ intrarea diagonală dorită și astfel se încheie argumentul inductiv.

Să demonstrăm acum restul formulelor.

Faptul că $h_{11} = \sqrt{a_{11}}$ este o consecință imediată a argumentului de inducție pentru $i = 1$. În cazul unui i generic, se obține relații sunt formulele de substituție directă pentru soluția sistemului liniar $H_{i-1}^T h = v$, iar demonstrația este completă.

Folosim Cholesky doar pentru sisteme cu matrice simetrică?

Să presupunem că avem de rezolvat un sistem de forma $Ax = b$, $A \in \mathbb{R}^{n\times n}$, $\det A \neq 0$, $b \in \mathbb{R}^n$.

Matricea A nu este considerată neapărat simetrică, însă prin înmulțire cu A^T, avem sistemul echivalent

$$A^T A x = A^T b, \tag{46}$$

a cărui matrice este simetrică și chiar pozitiv definită. (Demonstrați!)

Prin urmare se poate folosi factorizarea Cholesky care este mai eficientă decât factorizarea LU.

Vom vedea că factorizare Cholesky poate fi folosită și pentru rezolvarea aproximativă a sistemelor supradeterminate (cu aplicații practice în corelarea datelor, probleme de identificare a locației optime, eliminarea zgomotelor din semnale etc.).

Matrice de tip bandă

Matrice de tip bandă

Definiție
Spunem că o matrice $A \in \mathbb{R}^{m \times n}$ are bandă inferioară p dacă $a_{ij} = 0$ când $i > j + p$ și banda superioară q dacă $a_{ij} = 0$ când $j > i + q$.

Matricele diagonale sunt matrici cu benzi pentru care $p = q = 0$, în timp ce matricele trapezoidale au $p = 1$ sau $q = 1$ iar dacă $p = 1$ și $q = 1$ atunci spunedm că avem o matrice tridiagonală.

Rezultatul principal pentru matrici cu benzi este următorul.

Teoremă
Fie $A \in \mathbb{R}^{n \times n}$. Să presupunem că există o factorizare LU a lui A. Dacă A are lățimea de bandă superioară q și lățimea de bandă inferioară p, atunci L are lățimea de bandă inferioară p și U are lățimea de bandă superioară q.

În special, observați că aceeași zonă de memorie utilizată pentru A este suficientă pentru a stoca și factorizarea sa LU.

Să considerăm, într-adevăr, că o matrice A având lățimea de bandă superioară q și lățimea de bandă inferioară p este de obicei stocată într-o matrice $(p+q+1)\times n$, pe care o vom nota cu B, presupunând că

$$b_{i-j+q+1,j} = a_{ij} \tag{47}$$

pentru toți indicii i, j care se încadrează în banda matricei, în rest fiind zero.

De exemplu, în cazul matricei tridiagonale

$$A = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 2 \end{pmatrix} \tag{48}$$

stocarea compactă se citește

$$B = \begin{pmatrix} 0 & -1 & -1 & -1 & -1 \\ 2 & 2 & 2 & 2 & 2 \\ -1 & -1 & -1 & -1 & 0 \end{pmatrix} \tag{49}$$

Același format poate fi utilizat pentru stocarea factorizării LU a lui A.

Este clar că acest format de stocare poate fi destul de incomod în cazul în care doar câteva benzi ale matricei sunt mari.

La limită, dacă doar o coloană și un rând ar fi pline, am avea $p = q = n$ și astfel B ar fi o matrice plină cu multe intrări zero.

În cele din urmă, observăm că inversa unei matrice cu benzi este în general plină (așa cum se întâmplă pentru matricea A considerată mai sus).

Matrice tridiagonale

Considerăm cazul particular al unui sistem liniar cu matrice tridiagonală nesingulară A dată de

$$\begin{pmatrix} a_1 & c_1 & & 0 \\ b_2 & a_2 & \ddots & \\ & \ddots & \ddots & c_{n-1} \\ 0 & & b_n & a_n \end{pmatrix}. \tag{50}$$

În acest caz, matricile L și U din factorizarea LU a lui A sunt matrici bidiagonale de forma

$$\begin{pmatrix} 1 & 0 & & 0 \\ \beta_2 & 1 & \ddots & \\ & \ddots & \ddots & 0 \\ 0 & & \beta_n & 1 \end{pmatrix}, \quad \begin{pmatrix} \alpha_1 & c_1 & & \\ 0 & \alpha_2 & \ddots & 0 \\ & \ddots & \ddots & c_{n-1} \\ 0 & & 0 & \alpha_n \end{pmatrix} \tag{51}$$

Algoritmul Thomas

Coeficienții α_i, β_i pot fi calculați cu ușurință prin următoarele relații

$$\alpha_1 = a_1, \quad \beta_i = \frac{b_i}{\alpha_{i-1}}, \quad \alpha_i = a_i - \beta_i c_{i-1}, \quad i = 2, \cdots, n. \tag{52}$$

Algoritmul Thomas poate fi extins și pentru a rezolva întregul sistem tridiagonal $Ax = f$. Acest lucru înseamnă rezolvarea a două sisteme bidiagonale $Ly = f$ și $Ux = y$, pentru care se aplică următoarele formule:

$$(Ly = f): \quad y_1 = f_1, y_i = f_i - \beta_i y_{i-1}, \qquad i = 2, \cdots, n, \tag{53}$$

$$(Ux = y): \quad x_n = \frac{y_n}{\alpha_n}, \quad x_i = \frac{y_i - c_i x_{i+1}}{\alpha_i}, \qquad i = n-1, \cdots, 1. \tag{54}$$

Sisteme pe blocuri

În această secțiune ne ocupăm de factorizarea LU a matricelor partiționate în blocuri, în care fiecare bloc poate avea o dimensiune diferită.

Obiectivul nostru este dublu: optimizarea ocupării spațiului de stocare prin exploatarea adecvată a structurii matricei și reducerea costului de calcul al soluției sistemului.

Factorizarea LU

Fie $A = \mathbb{R}^{n\times n}$ următoarea matrice partiționată în blocuri

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, \tag{55}$$

unde $A_{11} \in \mathbb{R}^{r\times r}$ este o matrice nesingulară a cărei factorizare $L_{11}D_1R_{11}$ este cunoscută, în timp ce $A_{22} \in \mathbb{R}^{(n-r)\times(n-r)}$.

În acest caz este posibilă factorizarea A folosind doar factorizarea LU a blocului A_{11}. Într-adevăr, este adevărat că

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = \begin{pmatrix} L_{11} & 0 \\ L_{21} & I_{n-r} \end{pmatrix} \begin{pmatrix} D_1 & 0 \\ 0 & D_2 \end{pmatrix} \begin{pmatrix} R_{11} & R_{12} \\ 0 & I_{n-r} \end{pmatrix}, \tag{56}$$

unde

$$\begin{aligned} L_{21} &= A_{21}R_{11}^{-1}D_1^{-1}, \\ R_{12} &= D_1^{-1}L_{11}^{-1}A_{12}, \\ D_2 &= A_{22} - L_{21}D_1R_{12}. \end{aligned} \tag{57}$$

Dacă este necesar, procedura de reducere poate fi repetată pe matricea D_2, obținându-se astfel o versiune în bloc a factorizării LU.

Dacă A_{11} ar fi un scalar, abordarea de mai sus ar reduce cu unu dimensiunea factorizării unei matrice date.

Prin aplicarea iterativă a acestei metode se obține un mod alternativ de efectuare a eliminării Gauss.

Analiza erorii

Algebră liniară: completări

Descompunerii spectrală

Unul dintre cele mai utile rezultate legate de valorile proprii este teorema descompunerii spectrale, care afirmă că orice matrice simetrică A are o bază ortonormală de vectori proprii.

Teorema descompunerii spectrale

Fie A o matrice simetrică în $\mathbb{R}^{n\times n}$. Atunci există o matrice ortogonală $U \in \mathbb{R}^{n\times n}$ ($U^T U = U U^T = I$) și o matrice diagonală $D = \mathrm{diag}(d_1, d_2, ..., d_n)$ pentru care

$$U^T A U = D.$$

Coloanele matricei U din factorizare constituie o bază ortonormată formată din vectorii proprii ai lui A, iar elementele diagonale ale lui D sunt valorile proprii corespunzătoare.

Demonstrați că $\mathrm{tr}(A) = \sum_{i=1}^{n} \lambda_i(A)$ și $\det(A) = \prod_{i=1}^{n} \lambda_i(A)$.

Norme matriceale

Ansamblul valorilor proprii ale lui A se numește spectrul lui A, notat prin $\sigma(A)$.

Se pot demonstra următoarele proprietăți

$$\det(A) = \prod_{i=1}^{n} \lambda_i, \qquad \mathrm{tr}(A) = \sum_{i=1}^{n} \lambda_i \tag{58}$$

și se concluzionează că $\sigma(A) = \sigma(A^T)$, și $\sigma(A^H) = \sigma(\overline{A})$, unde $A^H = \overline{A}^T$.

Modulul maxim al valorilor proprii ale lui A se numește raza spectrală a lui A și se notează cu

$$\rho(A) = \max_{\lambda \in \sigma(A)} |\lambda|. \tag{59}$$

Norme în $\mathbb{R}^n$

Un exemplu de spațiu normat este $\mathbb{R}^n$, echipat, de exemplu, cu norma p (sau norma Hölder); aceasta din urmă se definește pentru un vector x cu componente x_i ca fiind

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}, \qquad \text{for} \quad 1 \leq p < \infty. \tag{60}$$

Observați că limita pe măsură ce p merge la infinit a lui $\|x\|_p$ există, este finită și este egală cu modulul maxim al componentelor lui x. O astfel de limită definește, la rândul său, o normă, numită norma infinit (sau norma maxim), dată de

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|. \tag{61}$$

Când $p = 2$, regăsim definiția standard a normei euclidiene

$$\|x\|_2 = \left(\sum_{i=1}^{n} |x_i|^2 \right)^{\frac{1}{2}} = (x^T x)^{\frac{1}{2}}. \tag{62}$$

Norme matriceale

Definiție
O normă matricială este o funcție $\|\cdot\| : \mathbb{R}^{m\times n} \to \mathbb{R}$ *astfel încât*

1. $\|A\| \geq 0 \ \forall A \in \mathbb{R}^{m\times n}$ *și* $\|A\| = 0$ *dacă și numai dacă* $A = 0$;
2. $\|\alpha A\| = |\alpha| \, \|A\| \ \forall \alpha \in \mathbb{R}, \forall A \in \mathbb{R}^{m\times n}$;
3. $\|A + B\| \leq \|A\| + \|B\| \ \forall A, B \ n\mathbb{R}^{m\times n}$.

Definiție
Spunem că o normă matricială $\|\cdot\|_{\mathbb{R}^{m\times n}}$ *este compatibilă sau consistentă cu normele vectorială* $\|\cdot\|_{\mathbb{R}^m}$ *și* $\|\cdot\|_{\mathbb{R}^n}$ *dacă*

$$\|Ax\|_{\mathbb{R}^m} \leq \|A\|_{\mathbb{R}^{m\times n}} \|x\|_{\mathbb{R}^n} \quad \forall x \in \mathbb{R}^n. \tag{63}$$

Definiție
Spunem că o normă matricială $\|\cdot\|$ este submultiplicativă dacă $\forall\, A \in \mathbb{R}^{n\times m}$, $\forall\, B \in \mathbb{R}^{m\times q}$.

$$\|A\,B\| \leq \|A\|\,\|B\|. \tag{64}$$

În multe lucrări, definiția unei norme matriciale include și submultiplicitatea.

Această proprietate nu este satisfăcută de toate normele matriciale. De exemplu, norma $\|A\|_\Delta = \max_{i=1,\ldots,n,j=1,\ldots,m} |a_{ij}|$ nu îndeplinește condiția submultiplicativă, de exemplu, această condiție nu este îndeplinită pentru pentru $A = B = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$. Prin urmare, este o normă, dar nu este o normă submultiplicativă.

Observați că, dată fiind o anumită normă submultiplicativă $\|\cdot\|_\alpha$, există întotdeauna o normă vectorială compatibilă. De exemplu, dat fiind orice vector fix $y \neq 0$ în $\mathbb{R}^n$, este suficient să se definească norma vectorială consistentă sub forma

$$\|x\| = \|x\, y^T\|_\alpha \quad \forall\, x \in \mathbb{R}^n. \tag{65}$$

Un exemplu de normă matricială este norma Frobenius (sau norma euclidiană în $\mathbb{R}^{n^2}$)

$$\|A\|_F = \sqrt{\sum_{i,j=1}^{n} |a_{ij}|^2} = \sqrt{\mathrm{tr}(A\,A^T)} \tag{66}$$

și este compatibilă cu norma vectorială euclidiană $\|\cdot\|_2$. Într-adevăr,

$$\|A\,x\|_2^2 = \sum_{i=1}^{n} |\sum_{j=1}^{n} a_{ij} x_j|^2 \leq \sum_{i=1}^{n} \left(\sum_{j=1}^{n} |a_{ij}|^2 \sum_{j=1}^{n} |x_j|^2 \right) = \|A\|_F^2 \|x\|_2^2. \tag{67}$$

Observați că pentru o astfel de normă $\|I_n\|_F = \sqrt{n}$. Pentru o normă oarecare, care ar putea fi o așteptare rezonabilă pentru $\|I_n\|$?

Norme matriceale induse

Teoremă
Fie $\|\cdot\|_{\mathbb{R}^m}$ *și* $\|\cdot\|_{\mathbb{R}^n}$ *norme vectoriale. Funcția*

$$\|A\| = \sup_{x \neq 0} \frac{\|Ax\|_{\mathbb{R}^m}}{\|x\|_{\mathbb{R}^n}} \tag{68}$$

este o normă matricială numită normă indusă sau normă matricială naturală.

Cazuri relevante de norme matriceale induse sunt așa-numitele norme p definite astfel

$$\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}. \tag{69}$$

Norma 1 și norma infinit sunt ușor de calculat, deoarece

$$\|A\|_1 = \max_{j=1,\ldots,n} \sum_{i=1}^{m} |a_{ij}|, \qquad \|A\|_\infty = \max_{i=1,\ldots,n} \sum_{i=1}^{n} |a_{ij}| \tag{70}$$

și se numesc norma sumei coloanelor și, respectiv, norma sumei rândurilor.

Mai mult, avem $\|A\|_1 = \|A^T\|_\infty$ și dacă A este simetrică $\|A\|_1 = \|A\|_\infty$.

Teoremă
Fie $\|\cdot\|_{\mathbb{R}^{m\times n}}$ o normă matriceală indusă de normele vectoriale $\|\cdot\|_{\mathbb{R}^m}$ și $\|\cdot\|_{\mathbb{R}^n}$. Atunci, următoarele relații sunt valabile:

1. $\|A\,x\|_{\mathbb{R}^m} \leq \|A\|_{\mathbb{R}^{m\times n}}\,\|x\|_{\mathbb{R}^n}$, *adică norma matriceală indusă este compatibilă cu norma vectorială care o induce;*
2. $\|I_n\| = 1$*;*
3. $\|A\,B\|_{\mathbb{R}^{m\times n}} \leq \|A\|_{\mathbb{R}^{m\times n}}\,\|B\|_{\mathbb{R}^{m\times n}}$, *adică fiecare normă matriceală indusă este submultiplicativă.*

Proof.
TO DO. □

Observați că normele p sunt submultiplicative. Mai mult, observăm că proprietatea de submultiplicativitate ar permite doar să concluzionăm că $\|I_n\| \geq 1$. Într-adevăr, $\|I_n\| = \|I_n\,I_n\| \leq \|I_n\|^2$.

Norma $\|A\|_\Delta = \max_{i=1,\ldots,n, j=1,\ldots,m} |a_{ij}|$ care nu este submultiplicativă, de asemenea, nu este o normă matriceală indusă. O normă care nu este indusă poate fi sau nu submultiplicativă. De exemplu, $\|\cdot\|_\Delta$ nu este submultiplicativă, dar norma Frobenius

$$\|A\|_F = \sqrt{\sum_{i,j=1}^{n} |a_{ij}|^2} = \sqrt{\operatorname{tr}(A\,A^T)} \tag{71}$$

este submultiplicativă, chiar dacă nu este indusă (de ce?), de asemenea.

Norma spectrală pentru matrice simetrice

Teoremă
Fie A o matrice reală simetrică. Atunci

$$\|A\|_2 = \rho(A). \tag{72}$$

Proof.
Deoarece A este simetrică, există matricea unitară U astfel încât $U^T A\, U = \text{diag}(\lambda_1, ..., \lambda_n)$, unde λ_i sunt valorile proprii ale lui A. Fie $y = U^T x$. Atunci

$$\begin{aligned}
\|A\|_2 &= \sup_{x\neq 0} \sqrt{\frac{\|A\,x\|_2}{\|x\|_2}} = \sup_{x\neq 0} \sqrt{\frac{\langle A\,x, A\,x\rangle}{\|x\|_2}} = \sup_{y\neq 0} \sqrt{\frac{\langle A\,Uy, A\,Uy\rangle}{\|Uy\|_2}} \\
&= \sup_{y\neq 0} \sqrt{\frac{\langle U^T A^T A\,Uy, y\rangle}{\|y\|_2}} = \sup_{y\neq 0} \sqrt{\frac{\langle \text{diag}(\lambda_1^2, ..., \lambda_n^2) y, y\rangle}{\|y\|_2}} \\
&= \sup_{y\neq 0} \sqrt{\frac{\sum_{i=1}^n \lambda_i^2 y_i^2}{\sum_{i=1}^n y_i^2}} = \sqrt{\max_{i=1,2,\ldots,n} |\lambda_i^2|} = \max_{i=1,2,\ldots,n} |\lambda_i| = \rho(A).
\end{aligned}$$

□

Matrice nesimetrice

Definiție
Fie $A \in \mathbb{R}^{m \times n}$. Se numesc valori singulare ale matricei A, numerele reale $\sigma_i(A)$ definite prin

$$\sigma_i(A) = \sqrt{\lambda_i(A^T A)}. \tag{73}$$

Dacă A este simetrică, atunci

$$\sigma_i(A) = \sqrt{\lambda)i(A^T A)} = \sqrt{\lambda_i(A^2)} = \sqrt{\lambda_i^2(A)} = |\lambda_i(A)|. \tag{74}$$

Norma spectrală pentru matrice simetrice

Teoremă
Fie $\sigma_1(A)$ cea mai mare valoare singulară a matrice $A \in \mathbb{R}^{n\times n}$. Atunci

$$\|A\|_2 = \sqrt{\rho(A^T A)} = \sigma_1(A). \tag{75}$$

Este clar că a calcula $\|A\|_2$ este mult mai costisitor decât cel al lui $\|A\|_1$ sau $\|A\|_\infty$. Cu toate acestea, dacă este necesară doar o estimare a lui $\|A\|_2$, următoarele relații pot fi utilizate în mod profitabil în cazul matricelor pătrate

$$\begin{aligned}
&\max_{i,j} |a_{ij}|| \leq \|A\|_2 \leq n \max |a_{ij}|, \\
&\frac{1}{\sqrt{n}} \|A\|_\infty \leq \|A\|_2 \leq \sqrt{n}\, \|A\|_\infty, \qquad (76) \\
&\frac{1}{\sqrt{n}} \|A\|_1 \leq \|A\|_2 \leq \sqrt{n}\, \|A\|_1, \qquad \|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty}.
\end{aligned}$$

Relații dintre norme și raza spectrală

Teoremă
Fie $\|\cdot\|$ o normă matriceală consistentă, atunci

$$\rho(A) \leq \|A\| \qquad \forall\, A \in \mathbb{R}^{n\times n}. \tag{77}$$

Proof.
Fie λ o valoare proprie a lui A și $v \neq 0$ un vector propriu asociat acestei valori proprii. Deoarece norma este consistentă, avem

$$|\lambda|\|v\| = \|\lambda\, v\| = \|A\, v\| \leq \|A\|\, \|v\|, \tag{78}$$

și deci $|\lambda| \leq \|A\|$. □

În restul prelegerilor noastre, dacă nu specificăm altceva, considerăm norma matricei spectrale și o vom nota cu $\|\cdot\|$.

Relații dintre norme și raza spectrală

Teoremă
Fie $A \in \mathbb{R}^{n\times n}$ și $\varepsilon > 0$. Atunci, există o normă matricealăindusă notată $\|\cdot\|_{A,\varepsilon}$ (depinzând de ε) astfel încât

$$\|\cdot\|_{A,\varepsilon} \leq \rho(A) + \epsilon. \tag{79}$$

Deci, fixând o toleranță arbitrară, mereu există o normă matricealăcare este apropiată de norma spectrală a matricei A.

Analiza senzivității soluției

La fiecare pas al GEM în urma rotunjirilor numerelor se rezolvă un sistem perturbat

$$(A + \delta A)(x + \delta x) = b + \delta b, \tag{80}$$

soluția acestui sistem perturbat fiind perturbată față de solutția sistemului de start

$$Ax = b. \tag{81}$$

Ne dorim să caracterizăm perturbarea δx în funcție de pertubările δA și δb.

Un rol important va fi jucat de *numărul de condiționare*.

Numărul de condiționare

Numărul de condiționare al unei matrice $A \in \mathbb{R}^{n\times n}$ este definit prin

$$K(A) = \|A\| \, \|A^{-1}\|, \tag{82}$$

unde $\|\cdot\|$ este o normă indusă.

Se poate observa că numărul de condiționare depinde de norma aleasă.

Se observă însă că indiferent de norma aleasă $K(A) \geq 1$ deoarece

$$1 = \|A\, A^{-1}\| \leq \|A\| \, \|A^{-1}\| = K(A).$$

Mai mult, $K(A) = K(A^{-1})$ și $K(\alpha\, A) = K(A)$, $\forall \alpha \neq 0$.

Pentru norma $\|\cdot\|_2$ pe $\mathbb{R}^{n\times n}$, $K_2(A) = \|A\|_2 \, \|A^{-1}\|_2$ este dat de

$$K_2(A) = \frac{\sigma_1(A)}{\sigma_n(A)}, \tag{83}$$

iar în cazul matricelor pozitiv definite

$$K_2(A) = \frac{\lambda_1(A)}{\lambda_n(A)} = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}. \tag{84}$$

$K_2(A)$ se numește numărul de condiționare spectral.

Theorem
Fie $A \in \mathbb{R}^{n\times n}$ o matrice inversabilă și $\delta A \in \mathbb{R}^{n\times n}$ astfel ca

$$\|A^{-1}\|\,\|\delta A\| < 1 \tag{85}$$

este verificată într-o normă indusă. Atunci dacă $x \in \mathbb{R}^n$ este soluție a sistemului $Ax = b$ cu $b \in \mathbb{R}^n$ $(b \neq 0)$ și $\delta x \in \mathbb{R}^n$ verifică

$$(A + \delta A)(x + \delta x) = b + \delta b, \tag{86}$$

atunci

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{K(A)}{1 - K(A)\|\delta A\|/\|A\|}\left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|}\right) \tag{87}$$

Câteva observații

Condiția $\|A^{-1}\| \, \|\delta A\| < 1$ asigură faptul că $(A + \delta A)$ rămâne inversabilă.

Dacă $\|A^{-1}\| \, \|\delta A\| < 1$, atunci $\rho(A^{-1}\delta A) < 1$.

Lemma
Fie $A \in \mathbb{R}^{n\times n}$, Atunci

$$\lim_{k\to\infty} A^k = 0 \qquad \Leftrightarrow \qquad \rho(A) < 1. \tag{88}$$

În plus, seria geometrică $\sum_{k=0}^{\infty} A^k$ este convergentă dacă și numai dacă $\rho(A) < 1$. În acest caz

$$\sum_{k=0}^{\infty} A^k = (I - A)^{-1}. \tag{89}$$

Prin urmare, dacă $\rho(A) < 1$, matricea $I - A$ este inversabilă și au loc inegalitățile

$$\frac{1}{1 + \|A\|} \leq \|(I - A)^{-1}\| \leq \frac{1}{1 - \|A\|}, \tag{90}$$

unde $\|\cdot\|$ este o matrice indusă astfel încât $\|A\| < 1$.

Demonstrația lemei

Dacă $\rho(A) < 1$ atunci $\exists\ \varepsilon > 0$ astfel încât $\rho(A) < 1 - \varepsilon$ și va rezulta că există o normă indusă astfel încât $\|A\| \leq \rho(A) + \epsilon < 1$.

Din $\|A^k\| \leq \|A\|^k < 1$ și din definiția convergenței rezultă că $\lim_{k\to\infty} A^k = 0$.

Invers. Presupunem că $\lim_{k\to\infty} A^k = 0$. Fie λ o valoare proprie a lui A. Atunci $A^k x = \lambda^k x$. Atunci $\lambda^k \to 0$. Deci avem $|\lambda| < 1$. Atunci, pentru că λ a fost considerată o valoare proprie generică, vom avea $\rho(A) < 1$.

Pentru următoarea parte din teoremă, să remarcăm pentru început că valorile proprii ale lui $I - A$ sunt $1 - \lambda(A)$, $\lambda(A)$ fiind valoare proprie a lui A. Pe de altă parte, deoarece $\rho(A) < 1$ deducem că $I - A$ este inversabilă.

Demonstrația lemei

Atunci, din identitatea

$$(I - A)(I + A + ... + A^n) = I - A^{n+1}, \tag{91}$$

și considerând limita $n \to \infty$ vom avea

$$(I - A)\sum_{k=0}^{\infty} A^k = I. \tag{92}$$

În final, deoarece pentru o normă indusă $\|I\| = 1$, avem

$$1 = \|I\| \leq \|(I - A)\| \, \|(I - A)^{-1}\| \leq (1 + \|A\|)\|(I - A)^{-1}\|, \tag{93}$$

adică prima inegalitate pe care noi o aveam de demonstrat.

Legat de ce-a de a doua inegalitate, din $I = I - A + A$ și prin multiplicare cu $(I - A)^{-1}$ avem

$$(I - A)^{-1} = I + A(I - A)^{-1}. \tag{94}$$

Demonstrația lemei

Trecând la normă în

$$(I - A)^{-1} = I + A(I - A)^{-1}, \tag{95}$$

găsim

$$\|(I - A)^{-1}\| \leq 1 + \|A\| \, \|(I - A)^{-1}\|, \tag{96}$$

adică inegalitatea a doua pentru că avem $\|A\| < 1$.

Revenim torema de demonstrat: Analiza a priori a erorii

Theorem
Fie $A \in \mathbb{R}^{n\times n}$ o matrice inversabilă și $\delta A \in \mathbb{R}^{n\times n}$ astfel ca

$$\|A^{-1}\| \, \|\delta A\| < 1 \tag{97}$$

este verificată într-o normă indusă. Atunci dacă $x \in \mathbb{R}^n$ este soluție a sistemului $Ax = b$ cu $b \in \mathbb{R}^n$ $(b \neq 0)$ și $\delta x \in \mathbb{R}^n$ verifică

$$(A + \delta A)(x + \delta x) = b + \delta b, \tag{98}$$

atunci

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{K(A)}{1 - K(A)\|\delta A\|/\|A\|} \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right) \tag{99}$$

Revenim la demonstrația teoremei

Deoarece $\|A^{-1}\delta A\| < 1$, avem că că $I + A^{-1}\delta A$ este inversabilă și din lema precedentă rezultă că

$$\|(I + A^{-1}\delta\, A)^{-1}\| \leq \frac{1}{1 - \|A^{-1}\delta A\|} \leq \frac{1}{1 - \|A^{-1}\|\,\|\delta A\|}. \tag{100}$$

Pe de altă parte, din

$$(A + \delta A)(x + \delta x) = b + \delta b, \tag{101}$$

și $A\,x = b$ găsim

$$\delta x = (I + A^{-1}\delta A)^{-1}A^{-1}(\delta b - \delta A\,x), \tag{102}$$

iar trecând la normă deducem

$$\|\delta x\| \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\|\,\|\delta A\|}(\|\delta b\| + \|\delta A\|\,\|x\|). \tag{103}$$

În final, împărțind prin $\|x\|$ (care nu e zero pentru că $b \neq 0$ și A este inversabilă), apoi folosinf că $\|x\| \geq \frac{\|b\|}{\|A\|}$ se deduce inegalitatea dorită.

Îmbunătățirea acurateței GEM

Îmbunătățirea acurateței GEM

După cum s-a menționat anterior, dacă matricea sistemului este prost condiționată, soluția generată de GEM ar putea fi inexactă, chiar dacă reziduul său la pasul i, adică $r^{(i)} = b^{(i)} - A^{(i)}x^{(i)}$, este mic. În această secțiune, menționăm două tehnici de îmbunătățire a acurateței soluției calculate de GEM.

Scalarea problemei

În cazul în care intrările din A variază foarte mult ca mărime, este probabil ca în timpul procesului de eliminare intrările mari să fie însumate cu intrările mici, având drept consecință apariției erorilor de rotunjire. Un remediu constă în efectuarea unei redimensionări a matricei A înainte de a se efectua eliminarea.

Scalarea pe rând a lui A constă în găsirea unei matrice diagonale nesingulare D_1 astfel încât intrările diagonale ale lui $D_1 A$ să aibă același ordin de mărime (aceeași dimensiune). Sistemul liniar $Ax = b$ se transformă în

$$D_1 A\, x = D_1\, b. \tag{104}$$

Atunci când atât liniile cât și coloanele lui A trebuie să fie scalate, versiunea scalată a sistemului devine

$$(D_1 A\, D_2) y = D_1\, b \qquad \text{cu} \qquad y = D_2^{-1} x, \tag{105}$$

presupunând, de asemenea, că D_2 este inversabil. Matricea D_1 redimensionează ecuațiile, în timp ce D_2 redimensionează necunoscutele.

Scalarea problemei

Observați că, pentru a preveni erorile de rotunjire, matricile de scalare sunt alese sub forma

$$D_1 = \text{diag}(\beta^{r_1}, ..., \beta^{r_n}), D_2 = \text{diag}(\beta^{c_1}, ..., \beta^{c_n}) \tag{106}$$

unde β este baza aritmeticii în virgulă mobilă utilizată, iar exponenții $r_1, \cdots, r_n, c_1, \cdots, c_n$ trebuie determinați.

Rafinare iterativă

Rafinarea iterativă este o tehnică de îmbunătățire a acurateței unei soluții obținute printr-o metodă directă. Să presupunem că sistemul liniar $AX = b$ a fost rezolvat cu ajutorul factorizării LU (cu pivotare parțială sau completă) și să notăm cu $x(0)$ soluția calculată. După ce s-a fixat o toleranță de eroare, tol, rafinarea iterativă se desfășoară astfel: pentru $i = 0, 1, \cdots$, până la convergență:

1. se calculează rezidualul $r^{(i)} = b^{(i)} - A^{(i)}x^{(i)}$;
2. rezolvă sistemul liniar $A^{(i)}z^{(i)} = r^{(i)}$ folosind factorizarea LU a lui $A^{(i)}$;
3. actualizați soluția stabilind $x^{(i+1)} = x^{(i)} + z^{(i)}$;
4. dacă $\|z^{(i)}\|/\|x^{(i+1)}\| < tol$, atunci încheiem procesul returnând soluția $x^{(i+1)}$. În caz contrar, algoritmul reîncepe de la pasul 1.

În absența erorilor de rotunjire, procesul s-ar opri la primul pas, producând soluția exactă.

Metode iterative de rezolvare a sistemelor

Despre convergența metodelor iterative

Ideea de bază a metodelor iterative este de a construi o secvență de vectori $x^{(k)}$ care se bucură de proprietatea de convergență

$$x = \lim_{k \to \infty} x^{(k)}, \tag{107}$$

unde x este soluția pentru

$$A\,x = b. \tag{108}$$

În practică, procesul iterativ se oprește la valoarea minimă a lui n astfel încât $\|x^{(n)} - x\| < \varepsilon$, unde ε este o toleranță fixă și $\|\cdot\|$ este orice normă vectorială convenabilă.

Cu toate acestea, deoarece soluția exactă nu este, evident, disponibilă, este necesar să se introducă criterii de oprire adecvate pentru a monitoriza convergența iterației.

Clasificarea metodelor iterative

Iterațiile introduse mai sus sunt cazuri speciale de metodelor iterative de forma

$$x^{(0)} = f_0(A, b), \tag{109}$$

$$x^{(n+1)} = f_{n+1}(x^{(n)}, x^{(n-1)}, \cdots, x^{(n-m)}, A, b), \qquad \text{pentru} \quad n \geq m, \tag{110}$$

unde f_i și $x^{(m)}, \cdots, x^{(1)}$ sunt funcții și, respectiv, vectori dați.

Numărul de pași de care depinde iterația curentă se numește **ordinul metodei**.

Dacă funcțiile f_i sunt independente de indicele de pas i, metoda se numește **staționară**, în caz contrar este **nestaționară**.

În sfârșit, dacă f_i depinde liniar de $x^{(0)}, \cdots, x^{(m)}$, metoda se numește **liniară**, în caz contrar este **neliniară**.

În lumina acestor definiții, metodele pe care le vom considera sunt metode iterative liniare staționare de ordinul întâi.

Luăm în considerare metodele iterative de forma

$$\text{dată} \quad x^{(0)}, \qquad x^{(k+1)} = Bx^{(k)} + f, \qquad k \geq 0, \tag{111}$$

având notată cu B o matrice pătrată de $n \times n$ numită matrice de iterație și cu f un vector care se obține din partea dreaptă b.

Definiție
Se spune că o metodă iterativă de forma (111) *este conformă cu* $Ax = b$ *dacă* f *și* B *sunt astfel încât* $x = Bx + f$ *. În mod echivalent,*

$$f = (I - B)A^{-1}b. \tag{112}$$

După ce am notat cu

$$e^{(k)} = x^{(k)} - x \tag{113}$$

eroarea la al k-lea pas al iterației, condiția de convergență se rezumă la a cere ca $\lim_{k\to\infty} e^{(k)} = 0$ pentru orice alegere a datelor inițiale. $x^{(0)}$ (adesea numită presupunere inițială).

Consistența nu este suficientă pentru a asigura convergența iterației.

Teoremă
Fie (111) *o metodă consistentă. Atunci, secvența de vectori* $x^{(k)}$ *converge către soluția lui* $Ax = b$ *pentru orice alegere a lui* $x^{(0)}$ *dacă și numai dacă* $\rho(B) < 1$.

Demonstrație: Din (111) și din ipoteza de consistență, rezultă relația recursivă

$$e^{(k+1)} = Be^{(k)}. \tag{114}$$

Prin urmare,

$$e^{(k)} = B^k e^{(0)}, \forall\ k = 0, 1, \cdots . \tag{115}$$

Rezultă că $\lim_{k\to\infty} B^k e^{(0)} = 0$ pentru orice $e^{(0)}$ dacă $\rho(B) < 1$.

Invers, să presupunem că $\rho(B) > 1$, atunci există cel puțin o valoare proprie $\lambda(B)$ cu modul mai mare de 1. Fie $e^{(0)}$ un vector propriu asociat cu λ; atunci $Be^{(0)} = \lambda\, e^{(0)}$ și, prin urmare, $e^{(k)} = \lambda^k\, e^{(0)}$. În consecință, $e^{(k)}$ nu poate tinde la 0 pe măsură ce $k \to \infty$, deoarece $|\lambda| > 1$.

Metode iterative liniare

O tehnică generală de concepere a unor metode iterative liniare consistente se bazează pe o *scriere aditivă* a matricei A de forma $A = P - N$, unde P și N sunt două matrici adecvate și P este nesingulară.

Din motive care vor fi clarificate în secțiunile ulterioare, P se numește **matrice de precondiționare**.

Mai exact, dat fiind $x^{(0)}$, se poate calcula $x^{(k)}$ pentru $k \geq 1$, rezolvând sistemele

$$Px^{(k+1)} = Nx^{(k)} + b \quad \Leftrightarrow \quad x^{(k+1)} = P^{-1}Nx^{(k)} + P^{-1}b, \quad k \geq 0. \tag{116}$$

Alternativ, iterația poate fi scrisă sub forma

$$x^{(k+1)} = x^{(k)} + P^{-1}r^{(k)} \quad k \geq 0. \tag{117}$$

unde

$$r^{(k)} = b - Ax^{(k)} \tag{118}$$

$$x^{(k+1)} = x^{(k)} + P^{-1} r^{(k)} \quad k \geq 0. \tag{119}$$

unde

$$r^{(k)} = b - Ax^{(k)} \tag{120}$$

reprezintă **vectorul rezidual** la pasul k.

Relația de mai sus evidențiază faptul că un sistem liniar, cu matricea coeficienților P, trebuie rezolvat pentru a actualiza soluția la pasul $k + 1$.

Prin urmare, P, pe lângă faptul că nu este singulară, trebuie să fie ușor de inversat, pentru a menține costul total de calcul scăzut. (Observați că, dacă P ar fi egal cu A și $N = 0$, metoda ar converge într-o singură iterație, dar la același cost ca și o metodă directă).

Menționăm (fără a demonstra) două rezultate care asigură convergența iterației

Teoremă
Fie $A = P - N$, cu A și P simetrice și pozitiv definite. Dacă matricea $2P - A$ este pozitiv definită, atunci metoda iterativă definită mai sus este convergentă pentru orice alegere a datelor inițiale $x^{(0)}$ și[2]

$$\rho(B) = \|B\|_A = \|B\|_P < 1. \tag{121}$$

Mai mult, convergența iterației este monotonă în raport cu normele $\|\cdot\|_P$ și $\|\cdot\|_A$ (adică, $\|e^{(k+1)}\|_P < \|e^{(k)}\|_P$ și $\|e^{(k+1)}\|_A < \|e^{(k)}\|_A$ $k = 0, 1, 2, \cdots$).

[2]Aici, $\|B\|_A = \|A^{1/2}B\|_2$, unde $A^{1/2}$ este soluția ecuației $X^2 = A$.

Teoremă

Fie $A = P - N$, A fiind simetrică și pozitiv definită. Dacă matricea $P + P^T - A$ este pozitiv definită, atunci P este inversabilă, metoda iterativă definită mai sus este convergentă monoton în raport cu norma $\|\cdot\|_A$ și $\rho(B) \leq \|B\|_A < 1$.

Jacobi, Gauss-Seidel și metode relaxate

Jacobi, Gauss-Seidel și metode relaxate

În această secțiune vom considera câteva metode iterative liniare clasice. Dacă intrările diagonale ale lui A sunt diferite de zero, putem extrage în fiecare ecuație necunoscuta corespunzătoare, obținând sistemul liniar echivalent

$$x_i = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1, j\neq i}^{n} a_{ij} x_j \right], \qquad i = 1, \cdots, n. \tag{122}$$

În metoda Jacobi, odată ce a fost aleasă o valoare inițială arbitrară $x^{(0)}$, $x^{(k+1)}$ se calculează prin formulele

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1, j\neq i}^{n} a_{ij} x_j^{(k)} \right], \qquad i = 1, \cdots, n. \tag{123}$$

Acest lucru înseamnă să efectuăm următoare scriere pentru A.

$$P = D, \qquad N = D - A = E + F, \tag{124}$$

unde D este matricea diagonală a intrărilor diagonale din A, E este matricea triunghiulară inferioară a intrărilor $e_{ij} = -a_{ij}$ dacă $i > j$, $e_{ij} = 0$ dacă $i \leq j$, iar F este matricea triunghiulară superioară a intrărilor $f_{ij} = -a_{ij}$ dacă $j > i$, $f_{ij} = 0$ dacă $j \leq i$. Ca o consecință,

$$A = D - (E + F), \tag{125}$$

iar matricea de iterație a metodei Jacobi este astfel dată de

$$B_J = D^{-1}(E + F) = I_n - D^{-1}A. \tag{126}$$

Metoda Jacobi relaxată

O generalizare a metodei Jacobi este metoda relaxată (sau JOR), în care, după introducerea unui parametru de relaxare ω, iterația Jacobi este înlocuită cu

$$x_i^{(k+1)} = \frac{\omega}{a_{ii}} \left[b_i - \sum_{j=1, j\neq i}^{n} a_{ij} x_j^{(k)} \right] + (1-\omega) x_i^{(k)}, \qquad i = 1, \cdots, n. \tag{127}$$

Matricea de iterație corespunzătoare este

$$B_{J_\omega} = \omega B_J + (1-\omega) I_n. \tag{128}$$

Scrisă în termenii vectorului rezidual, metoda JOR corespunde la

$$x^{(k+1)} = x^{(k)} + \omega D^{-1} r^{(k)}. \tag{129}$$

Această metodă este conformă pentru orice $\omega = 0$, iar pentru $\omega = 1$ coincide cu metoda Jacobi.

Gauss-Seidel method

Metoda Gauss-Seidel se deosebește de metoda Jacobi prin faptul că la al $k+1$-lea pas se folosesc valorile disponibile ale lui $x^{(k+1)}$ pentru a actualiza soluția, astfel încât, se are

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right], \qquad i = 1, \cdots, n. \tag{130}$$

Această metodă echivalează cu efectuarea următoarei împărțiri pentru A în

$$P = D - E, \qquad N = F, \tag{131}$$

iar matricea de iterație asociată este

$$B_{GS} = (D - E)^{-1} F. \tag{132}$$

Gauss-Seidel over-relaxation method

Pornind de la metoda Gauss-Seidel, prin analogie cu ceea ce s-a făcut pentru iterațiile Jacobi, introducem metoda relaxării succesive (sau metoda SOR)

$$x_i^{(k+1)} = \frac{\omega}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right] + (1-\omega) x_i^{(k)}, \quad i = 1, \cdots, n. \tag{133}$$

Metoda SOR poate fi scrisă în formă vectorială sub forma

$$(I - \omega D^{-1} E) x^{(k+1)} = [(1-\omega) I_n + \omega D^{-1} F] x^{(k)} + \omega D^{-1} b, \tag{134}$$

din care matricea de iterație este

$$B(\omega) = (I - \omega D^{-1} E)^{-1} [(1-\omega) I + \omega D^{-1} F]. \tag{135}$$

Înmulțind cu D ambele părți ale lui

$$(I - \omega D^{-1}E)x^{(k+1)} = [(1-\omega)I_n + \omega D^{-1}F]x^{(k)} + \omega D^{-1}b, \qquad (136)$$

și amintind că $A = D - (E + F)$, rezultă următoarea formă a metodei SOR

$$x^{(k+1)} = x^{(k)} + \left(\frac{1}{\omega}D - E\right)^{-1} r^{(k)}. \qquad (137)$$

Este compatibilă pentru orice $\omega \neq 0$, iar pentru $\omega = 1$ coincide cu metoda Gauss-Seidel. În special, dacă $\omega \in (0, 1)$, metoda se numește sub-relaxare, în timp ce dacă $\omega > 1$ se numește supra-relaxare.

Convergența metodelor Jacobi și Gauss-Seidel

Convergența metodelor Jacobi și Gauss-Seidel

Există clase speciale de matrici pentru care este posibil să se stabilească a priori unele rezultate de convergență pentru metodele examinate în secțiunea anterioară. Primul rezultat în această direcție este următorul.

Teoremă
Dacă A este o matrice dominantă strict diagonală pe linii, metodele Jacobi și Gauss-Seidel sunt convergente.

Proof: Să demonstrăm doar partea teoremei referitoare la metoda Jacobi. Deoarece A este strict diagonal dominantă pe rânduri, $|a_{ii}| > \sum_{j=1}^{n} |a_{ij}|$ pentru $j \neq i$ și $i = 1, \cdots, n$. În consecință,
$\|B_J\|_\infty = \max_{i=1,\cdots,n} |a_{ij}|/|a_{ii}| < 1$, astfel încât metoda Jacobi este convergentă.

Teoremă
Dacă A și $2D - A$ sunt matrici simetrice și pozitiv definite, atunci metoda Jacobi este convergentă și $\rho(B_J) = \|B_J\|_A = \|B_J\|_D$.

Demonstrație: Teorema rezultă din primul rezultat general luând $P = D$.

Teoremă
Dacă A este simetrică și pozitiv definită, metoda Gauss-Seidel este convergentă monotonic în raport cu norma $\|\cdot\|_A$.

Proof: Putem aplica al doilea rezultat general pentru matricea $P = D - E$, după ce verificăm că $P + P^T - A$ este definită pozitiv. Într-adevăr,

$$P + P^T - A = 2D - E - F - A = D, \tag{138}$$

după ce am observat că $(D - E)T = D - F$. Încheiem observând că D este pozitiv definită.

În cele din urmă, dacă A este tridiagonală (sau tridiagonală în bloc), se poate demonstra că

$$\rho(B_{GS}) = \rho^2(B_J) \tag{139}$$

De aici putem concluziona că ambele metode converg sau nu converg în același timp. În primul caz, metoda Gauss-Seidel converge mai repede decât metoda Jacobi, iar rata de convergență asimptotică a metodei Gauss-Seidel este dublă față de cea a metodei Jacobi. În special, dacă A este tridiagonală și simetrică pozitiv definită, teorema de mai sus implică convergența metodei Gauss-Seidel, iar $\rho(B_{GS}) = \rho^2(B_J)$ asigură convergența și pentru metoda Jacobi.

Convergența metodelor relaxate

Teoremă
Dacă A este simetrică și pozitiv definită, atunci metoda JOR este convergentă dacă $0 < \omega < 2/\rho(D^{-1}A)$.

Demonstrație: Demonstrația rezultă din $B_{J_\omega} = \omega B_J + (1-\omega)I_n$ și $B_J = D^{-1}(E+F) = I_n - D^{-1}A$ și observând că A are toate valorile proprii reale.

Convergența metodelor relaxate

Teoremă
Dacă metoda Jacobi este convergentă, atunci metoda JOR converge dacă $0 < \omega \leq 1$.

Demonstrație: Obținem că valorile proprii ale lui B_{J_ω} sunt

$$\mu_k = \omega\lambda_k + 1 - \omega, \qquad k = 1, ..., n, \tag{140}$$

unde λ_k sunt valorile proprii ale lui B_J. Apoi, reamintind formula lui Euler pentru reprezentarea unui număr complex, lăsăm $\lambda_k = r_k e^{i\theta k}$ și obținem

$$|\mu_k|^2 = \omega^2 r_k^2 + 2\omega r_k \cos(\theta_k)(1-\omega) + (1-\omega)^2 \leq (\omega r_k + 1 - \omega)^2, \tag{141}$$

care este mai mică decât 1 dacă $0 < \omega \leq 1$.

Convergența metodelor relaxate

Teoremă
Pentru orice $\omega \in \mathbb{R}$ *avem* $\rho(B(\omega) \geq |\omega - 1|$*; prin urmare, metoda SOR nu converge dacă* $\omega \leq 0$ *sau* $\omega \geq 2$*.*

Demonstrație: Dacă $\{\lambda_i\}$ reprezintă valorile proprii ale matricei de iterație SOR, atunci

$$\prod_{i=1}^{n} \lambda_i = |\det[(1-\omega)I + \omega D^{-1}F]| = |1-\omega|^n \tag{142}$$

Prin urmare, trebuie să existe cel puțin o valoare proprie λ_i astfel încât $|\lambda_i| \geq |1-\omega|$ și, astfel, pentru a asigura convergența, trebuie să avem $|1-\omega| < 1$, adică $0 < \omega < 2$.

Sisteme nedeterminate

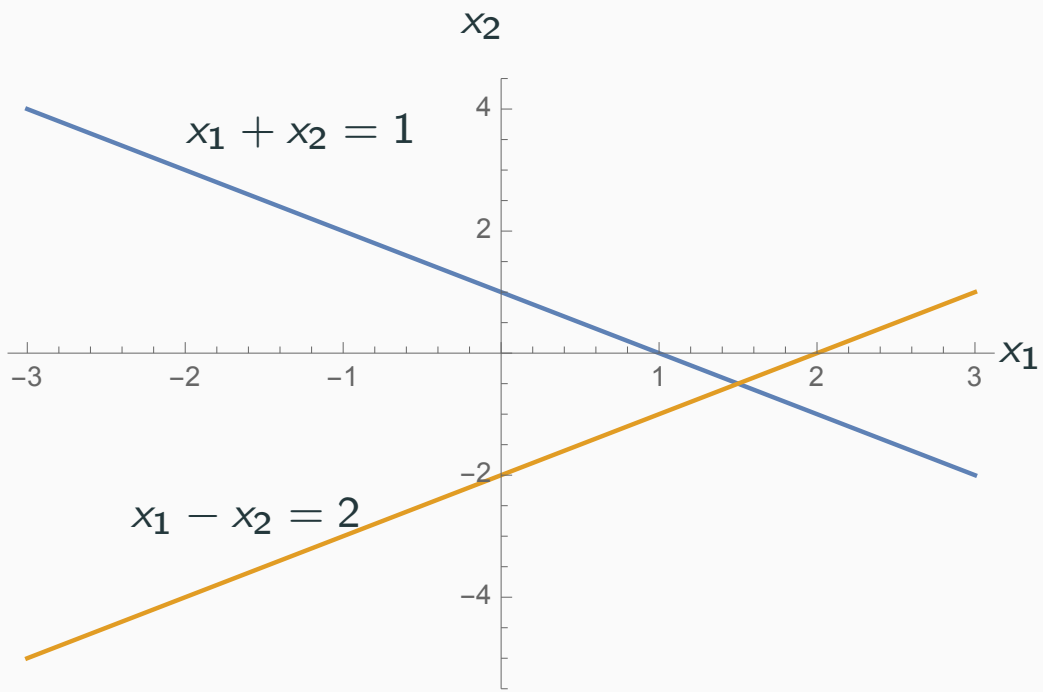
“Soluția” sistemelor supradeterminate

Sisteme algebrice liniare

Să considerăm următorul sistem algebric de ecuații liniare:

$$x_1 + x_2 = 1,$$
$$x_1 - x_2 = 2.$$

Semnificația geometrică a acestui sistem este că se caută un punct de intersecție a două drepte, vezi figura.



Eliminarea gaussiană

În mod clar, efectuând o eliminare gaussiană, înmulțim prima ecuație cu (-1) și o adăugăm la a doua pentru a obține următorul sistem echivalent

$$\begin{aligned} x_1 + x_2 &= 1, \\ -2\,x_2 &= 1. \end{aligned}$$

A doua ecuație ne dă $x_2 = -\frac{1}{2}$ și, împreună cu prima ecuație, găsim și $x_1 = \frac{3}{2}$.

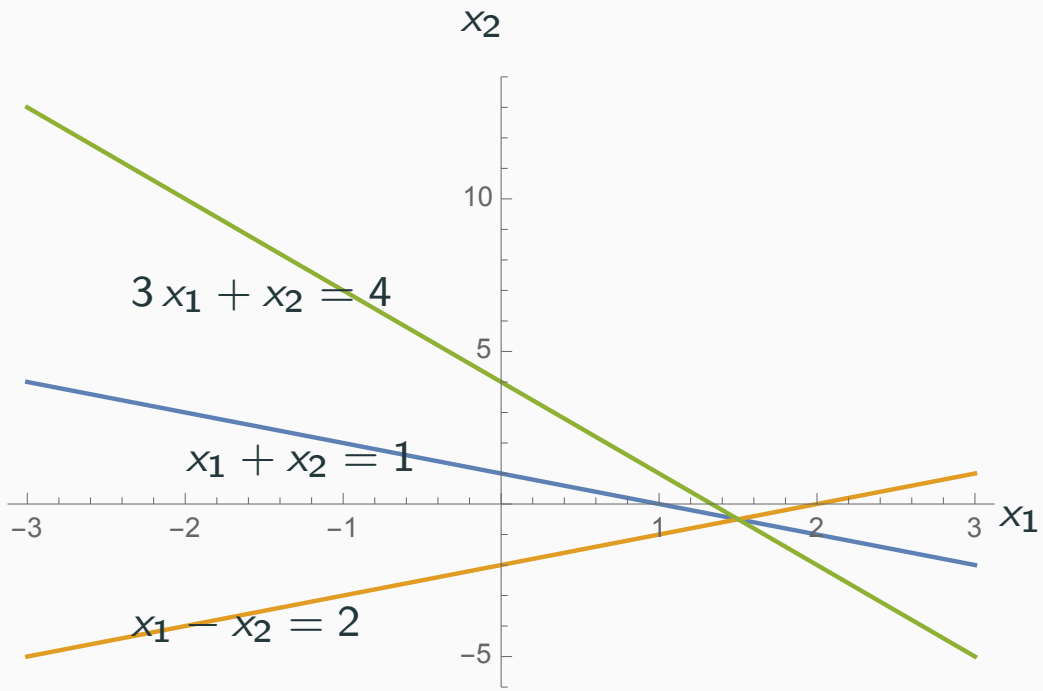
Prin urmare, punctul de intersecție al celor două drepte este punctul $(x_1, x_2) = (\frac{3}{2}, -\frac{1}{2})$.

Sisteme supradeterminate

Acum, să luăm în considerare următorul sistem algebric de ecuații liniare:

$$x_1 + x_2 = 1,$$
$$x_1 - x_2 = 2,$$
$$3\,x_1 + x_2 = 4.$$

Punctul de intersecție al acestor trei drepte este același cu cel din exemplul anterior, deoarece ultima ecuație este redundantă.

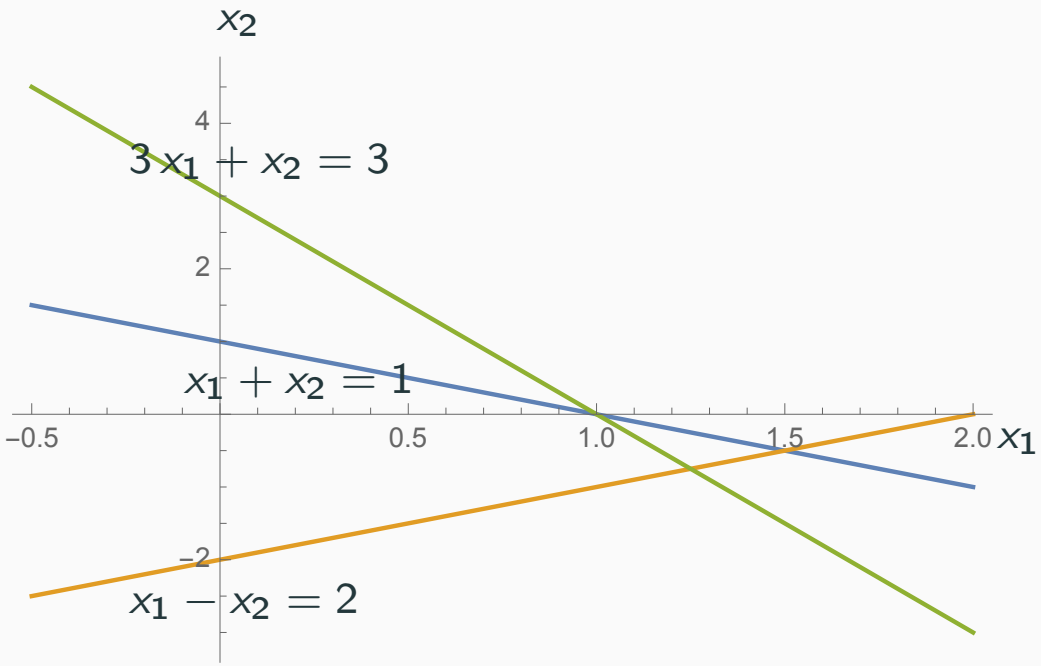


Sisteme supradeterminate

Dar ce se întâmplă dacă a treia ecuație nu este redundantă? Spunem că sistemul este supradeterminat. De exemplu

$$\begin{aligned} x_1 + x_2 &= 1, \\ x_1 - x_2 &= 2, \\ 3\,x_1 + x_2 &= 3. \end{aligned}$$

Nu există un punct de intersecție a acestor trei drepte.



"Soluția" sistemelor supradeterminate

Deci, sistemul algebric

$$\begin{aligned} x_1 + x_2 &= 1, \\ x_1 - x_2 &= 2, \\ 3\,x_1 + x_2 &= 3. \end{aligned}$$

nu are o soluție.

Cu toate acestea, suntem în continuare interesați să găsim un punct (x_1, x_2) care este **"o soluție aproximativă"**.

"Soluția" sistemelor supradeterminate

Deci, sistemul algebric

$$\begin{aligned} x_1 + x_2 &= 1, \\ x_1 - x_2 &= 2, \\ 3\,x_1 + x_2 &= 3 \end{aligned}$$

nu are o soluție.

Cu toate acestea, suntem în continuare interesați să găsim un punct (x_1, x_2) care este **"'o soluție aproximativă"**.

De ce?

“Soluția” sistemelor supradeterminates

De ce?

Pentru a răspunde la această întrebare, trebuie să remarcăm că sistemul algebric

$$\begin{aligned} x_1 + x_2 &= 1, \\ x_1 - x_2 &= 2, \\ 3\,x_1 + x_2 &= 4 \end{aligned}$$

are o soluție unică, în timp ce perturbată sistemul algebric perturbat

$$\begin{aligned} x_1 + x_2 &= 1, \\ x_1 - x_2 &= 2, \\ 3\,x_1 + x_2 &= 4.0001 \end{aligned}$$

nu are o soluție.

De ce?

Pentru că aceste sisteme provin din practică și este posibil să avem nu există valori exacte (corecte) ale coeficienților. O mică eroare în măsurători ar putea conduce la un sistem algebric nedeterminat și ne interesează să vedem care punct (x_1, x_2) satisface "mai bine" sistemul.

"Soluția" sistemelor supradeterminate

Mai întâi de toate, să remarcăm că sistemul algebric

$$\begin{aligned} x_1 + x_2 &= 1, \\ x_1 - x_2 &= 2, \\ 3\,x_1 + x_2 &= 3. \end{aligned}$$

poate fi scris sub formă de matrice sub forma

$$A\,x = b,$$

unde $A = \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ 3 & 1 \end{pmatrix} \in \mathbb{R}^{3\times 2}$, $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2$ și $b = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \in \mathbb{R}^3$.

“Soluția” sistemelor supradeterminate

Printr-o “soluție” a sistemului supradeterminat

$$A\,x = b$$

înțelegem un vector $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2$ astfel încât $A\,x$ să nu fie "atât de departe" de b. Cu alte cuvinte, un vector $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \in \mathbb{R}^2$ astfel încât $A\,x - b$ să nu fie “departe” de $\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$.

Dar ce înseamnă că un vector nu este “atât de departe” de un alt vector?

Problema celor mai mici pătrate: “Soluția” sistemelor supra-determinate, Data Fitting

Am văzut că soluția sistemului liniar $Ax = b$ există și este unică dacă $n = m$ și A este nesingulară.

În această secțiune dăm un sens soluției unui sistem liniar în cazul supradeterminat, $m > n$.

"Soluția" sistemelor supradeterminate

Aproximarea soluției sistemelor supradeterminate

Să presupunem că ni se dă un sistem liniar de forma

$$Ax = b, \quad \text{unde} \quad A \in \mathbb{R}^{m \times n} \quad \text{și} \quad b \in \mathbb{R}^m \quad \text{cu} \quad m > n.$$

Presupunem, de asemenea, că $\operatorname{rank}(A) = n$.

În aceste condiții, sistemul de ecuații liniare considerat poate fi incompatibil (nu are soluție).

Observăm că un sistem nedeterminat nu are, în general, soluție decât dacă partea dreaptă b este un element al lui $\operatorname{Range}(A) := \{y \in \mathbb{R}^m | \exists\, x \in \mathbb{R}^n \text{ a. î. } Ax = y\}$.

Aproximarea soluției sistemelor supradeterminate

O abordare obișnuită pentru găsirea unei soluții aproximative constă în alegerea găsirea acelui x pentru care se realizează valoarea minimă a normei reziduului $r = Ax - b$, pe $\mathbb{R}^m$, adică

$$\text{(LS)} \qquad \min_{x \in \mathbb{R}^n} \|Ax - b\|^2.$$

Aceasta este o problemă de minimizare a unei funcții pătratice pe întreg spațiu, funcția obiectiv pătratică fiind dată de

$$f(x) = \langle A^T Ax, x \rangle - 2\langle b, Ax \rangle + \|b\|^2.$$

Problema celor mai mici pătrate

Dat fiind $A \in \mathbb{R}^{m \times n}$ cu $m \geq n$, $b \in \mathbb{R}^m$, spunem că $x^* \in \mathbb{R}^n$ este o soluție a sistemului liniar $Ax = b$ în sensul celor mai mici pătrate dacă

$$f(x^*) = \|Ax^* - b\|_2^2 \leq \min_{x \in \mathbb{R}^n} \underbrace{\|Ax - b\|_2^2}_{:= f(x)} = \min_{x \in \mathbb{R}^n} f(x). \tag{143}$$

Astfel, problema constă în minimizarea normei euclidiene a reziduului. Soluția problemei de minimizare poate fi găsită prin impunerea condiției ca gradientul funcției f să fie egal cu zero la x^*.

Din

$$f(x) = (Ax - b)^T (Ax - b) = x^T A^T Ax - 2x^T A^T b + b^T b, \qquad (144)$$

aflăm că

$$\nabla f(x^*) = 2A^T Ax^* - 2A^T b = 0, \qquad (145)$$

de unde rezultă că x^* trebuie să fie soluția sistemului pătratic

$$A^T Ax^* = A^T b \qquad (146)$$

cunoscut sub numele de *ecuația normală*.

Sistemul este nesingular dacă A are rang complet și, în acest caz, soluția celor mai mici pătrate există și este unică.

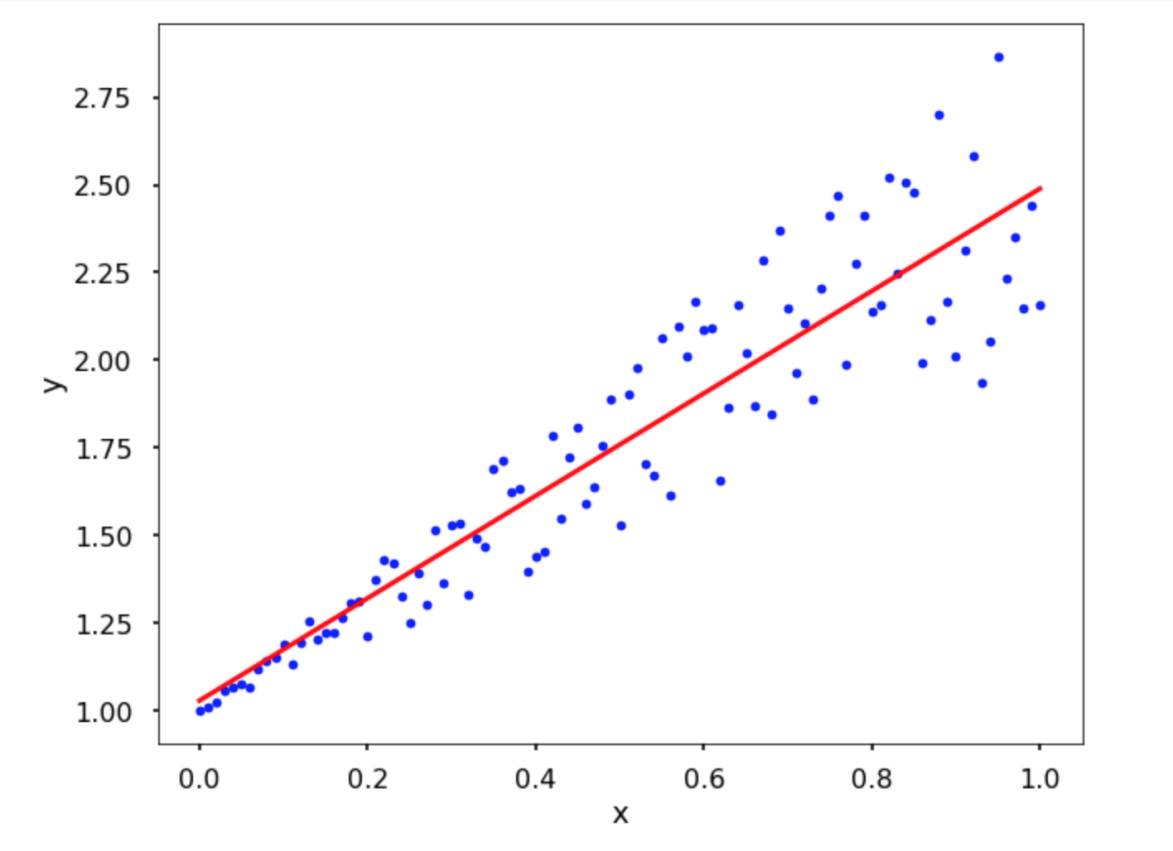
Observăm că $B = A^T A$ este o matrice simetrică și pozitiv definită. Astfel, pentru a rezolva ecuațiile normale, se poate calcula mai întâi factorizarea Cholesky $B = H^T H$ și apoi se pot rezolva cele două sisteme $H^T y = A^T b$ și $Hx^* = y$. Cu toate acestea, din cauza erorilor de rotunjire, calculul lui $A^T A$ poate fi afectat de o pierdere de cifre semnificative, cu o pierdere consecventă a definiției pozitive sau a nesingularității matricei, așa cum se întâmplă în următorul exemplu (implementat în MATLAB) în care, pentru o matrice A cu rang complet, matricea corespunzătoare $fl(ATA)$ se dovedește a fi singulară

$$A = \begin{pmatrix} 1 & 1 \\ 2^{-27} & 0 \\ 0 & 2^{-27} \end{pmatrix}, \qquad fl(A^T A) = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}. \tag{147}$$

Prin urmare, în cazul matricelor prost condiționate, este mai convenabil să se utilizeze o metodă alternativă bazată pe factorizarea QR.

A 2D picture

Un domeniu în care se utilizează problema cel mai mici pătrate este corelarea datelor.



Linear Data Fitting in 2D

Date n puncte în $\mathbb{R}^n$, obiectivul este de a găsi o dreaptă de forma

$$y = a\,x + b$$

care se potrivește cel mai bine cu acestea. Aceasta înseamnă că trebuie să găsim a și b care să definească această dependență liniară.

Corespondențele liniare corespunzătoare care trebuiesc corelate sunt

$$y_i = a\,x_i + b, \qquad i = 1, 2, ..., n,$$

adică, sistemul care trebuie "rezolvat" este

$$\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \\ x_n & 1 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}.$$

Deci, găsiți a și b "soluție" pentru

$$\underbrace{\begin{pmatrix} x_1 & 1 \\ x_2 & 1 \\ \vdots & \\ x_n & 1 \end{pmatrix}}_{:=X} \begin{pmatrix} a \\ b \end{pmatrix} = \underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}}_{:=y}.$$

Soluția problemei celor mai mici pătrate este

$$\begin{pmatrix} a \\ b \end{pmatrix} = (X^T X)^{-1} X^T y.$$

Data Fitting

Unul dintre domeniile în care se utilizează problema celor mai mici pătrate este corelarea datelor.

Linear Data Fitting

Să presupunem că ni se dă un set de date (s_i, t_i), $i = 1, 2, ..., m$, unde $s_i \in \mathbb{R}^n$ și $t_i \in \mathbb{R}$, și să presupunem că o relație liniară de forma

$$t_i = \langle s_i, x \rangle, \qquad i = 1, 2, ..., m,$$

este căutată. Găsiți x pentru a putea aproxima această dependență liniară!

Aplicații?

Deci, problema este de a găsi vectorul de parametri $x \in \mathbb{R}^n$ care rezolvă problema

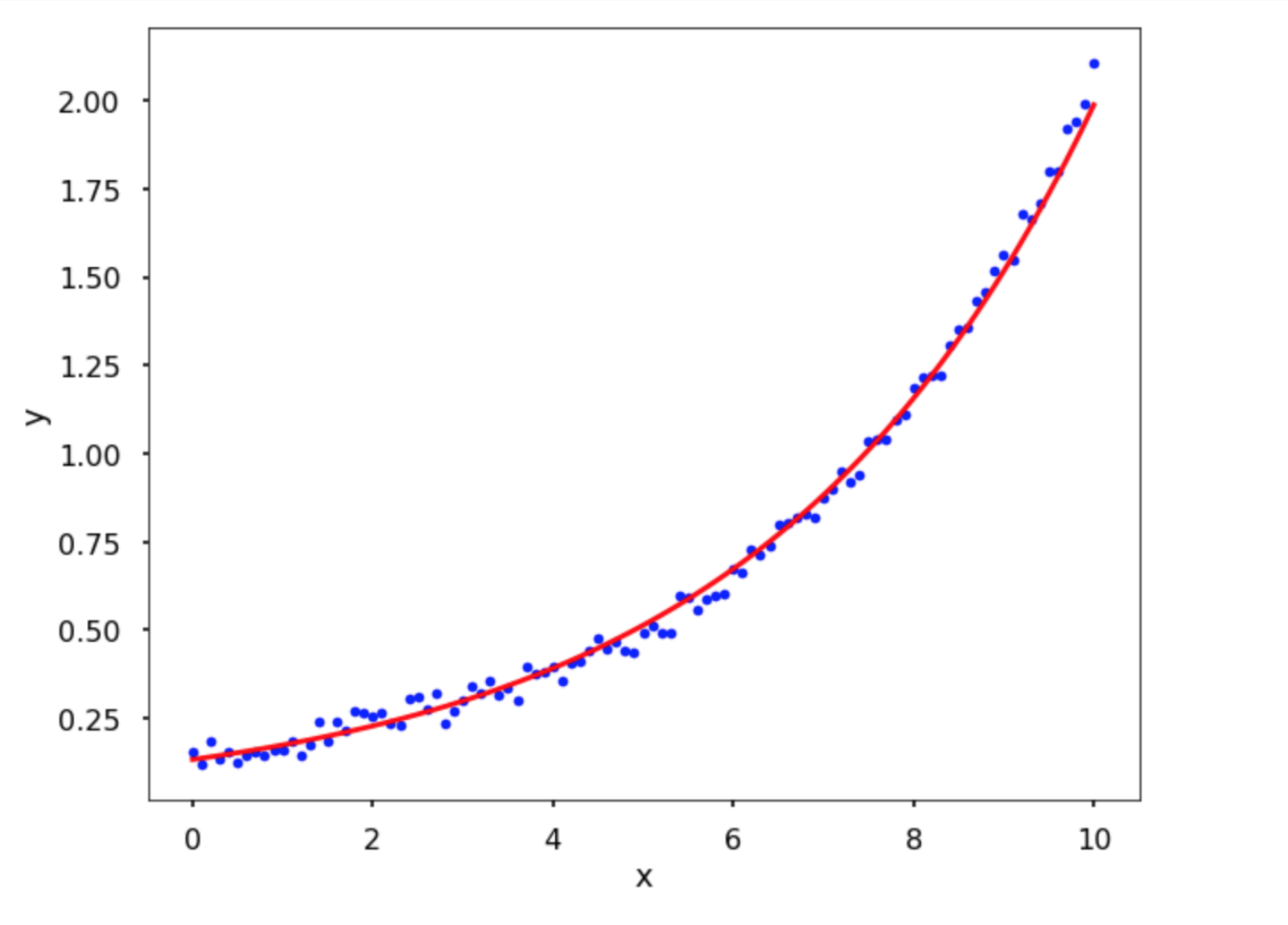
$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^{m} (\langle s_i, x \rangle - t_i)^2.$$

Aceasta este o problemă (LS) scrisă ca

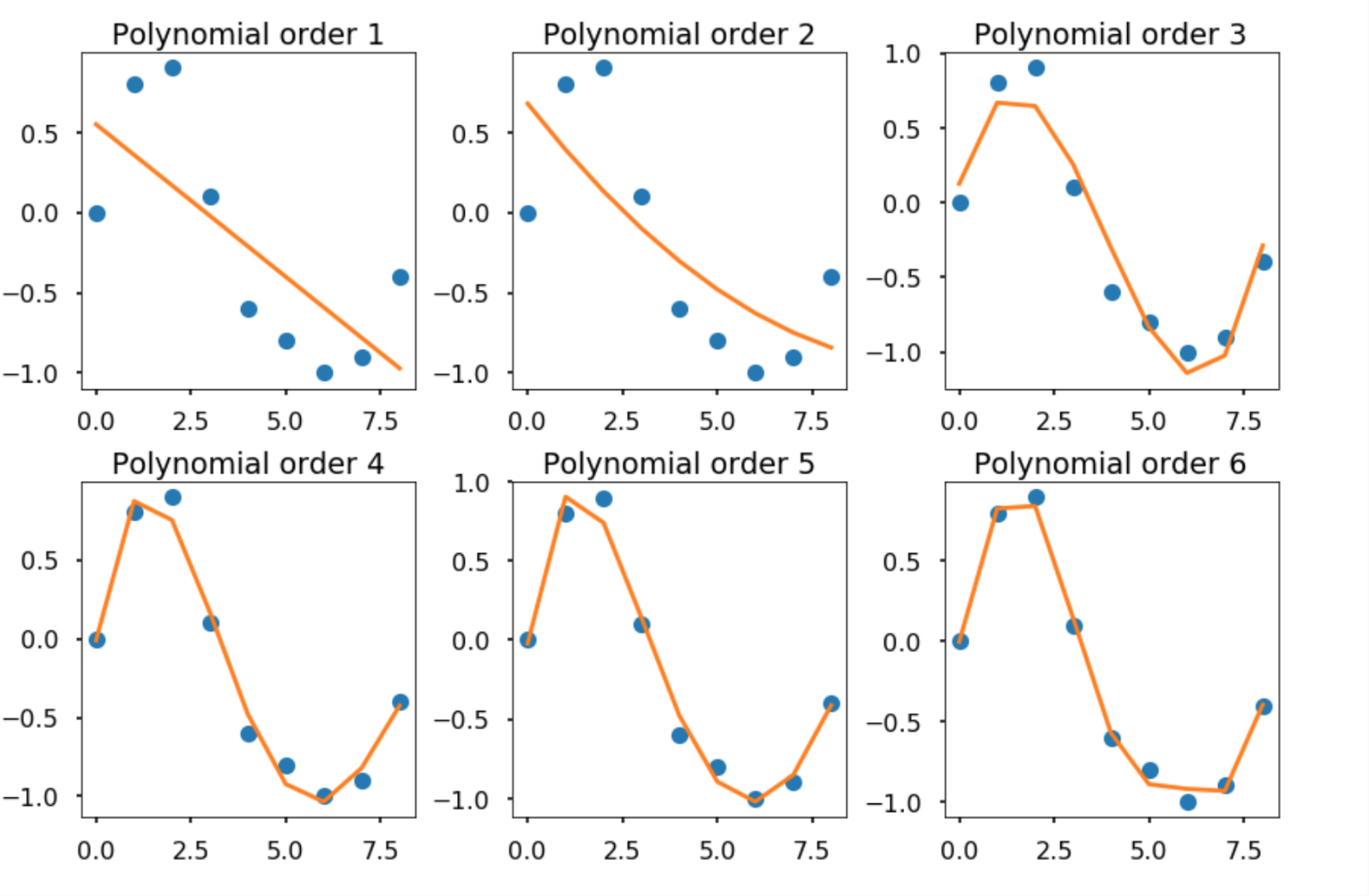
$$\min_{x \in \mathbb{R}^n} \|S\,x - t\|^2,$$

unde $S = \begin{pmatrix} - - s_1^T - - \\ - - s_2^T - - \\ \vdots \\ - - s_m^T - - \end{pmatrix}$, $t = \begin{pmatrix} t_1 \\ t_2 \\ \vdots \\ t_m \end{pmatrix}$.

Alte situații



Alte situații



Mai multe despre corelarea datelor

Abordarea celor mai mici pătrate poate fi utilizată și în cazul ajustărilor neliniare. Să presupunem, de exemplu, că ni se dă un set de puncte în $\mathbb{R}^2$: (u_i, y_i), $i = 1, 2, ..., m$, și că știm a priori că aceste puncte sunt aproximativ legate prin intermediul unui polinom de grad cel mult d; adică există $a_0, a_1, ..., a_d$ astfel încât

$$\sum_{j=0}^{d} a_j u_i^j \approx y_i, \qquad i = 1, ..., m.$$

Abordarea prin metoda celor mai mici pătrate a acestei probleme este: caută $a_0, a_1, ..., a_d$ care să fie soluția celor mai mici pătrate a sistemului liniar

$$\begin{pmatrix} 1 & u_1 & u_1^2 & \cdots & u_1^d \\ 1 & u_2 & u_2^2 & \cdots & u_2^d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & u_m & u_m^2 & \cdots & u_m^d \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_d \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix}.$$

(LS) este, desigur, bine definită dacă $m \geq d + 1$. Matricea este așa-numita matrice Vandermonde, despre care se știe că este de rang $d + 1$ dacă $d + 1$ din u_i-uri sunt diferite între ele.

Regularizarea Problemei celor mai mici pătrate, Eliminarea zgomotului dintr-un semnal

Regularizarea Problemei celor mai mici pătrate

Atunci când A este subdeterminată, adică atunci când există mai puține ecuații decât variabile, există mai multe soluții optime pentru problema celor mai mici pătrate și nu este clar care dintre aceste soluții optime este cea care trebuie luată în considerare.

În modelul de optimizare ar trebui încorporat un anumit tip de informații prealabile despre x.

O modalitate de a face acest lucru este de a lua în considerare o problemă penalizată în care o funcție de regularizare $R(\cdot)$ este adăugată la funcția obiectiv.

Regularized Least Squares

RLS

Problema regularizată a celor mai mici pătrate (RLS) are forma

$$(\mathrm{RLS}) \qquad \min_{x \in \mathbb{R}^n} \|A\,x - b\|^2 + \lambda\, R(x),$$

unde $\lambda > 0$ este parametrul de regularizare. Pe măsură ce λ devine mai mare, funcția de regularizare primește o pondere mai mare.

În multe cazuri, se consideră că regularizarea este pătratică. În special $R(x) = \|D\,x\|^2$, cu $D \in \mathbb{R}^{p \times n}$ dat. Funcția de regularizare pătratică urmărește să controleze norma lui $D\,x$ și este formulată după cum urmează:

$$(\mathrm{RLS}) \qquad \min_{x \in \mathbb{R}^n} \|A\,x - b\|^2 + \lambda\, \|D\,x\|^2,$$

sau, echivalent ca

$$(\mathrm{RLS}) \qquad \min_{x \in \mathbb{R}^n} \{f_{\mathrm{RLS}}(x) = \langle (A^T\,A + \lambda\, D^T\,D)\,x, x\rangle - 2\langle b, A\,x\rangle + \|b\|^2\},$$

$$\text{(RLS)} \qquad \min_{x \in \mathbb{R}^n} \{ f_{\text{RLS}}(x) = \big\langle (A^T A + \lambda D^T D)\, x, x \big\rangle - 2 \big\langle b, A\, x \big\rangle + \|b\|^2 \},$$

Deoarece D și λ sunt căutați a.î. matricea hessiană a funcției obiectiv dată de $\nabla^2 f_{\text{RLS}}(x) = 2\,(A^T A + \lambda D^T D) \succ 0$ să fie positiv definită, rezultă că orice punct staționar este un punct minim global.

Punctele staționare sunt cele care satisfac

$$\nabla f_{\text{RLS}}(x) = 0$$

adică

$$(A^T A + \lambda D^T D)\, x = A^T b.$$

Prin urmare, dacă D și λ sunt astfel încât $A^T A + \lambda D^T D \succ 0$, atunci soluția RLS este dată de

$$x_{RLS} = (A^T A + \lambda D^T D)^{-1} A^T b.$$

Other situations

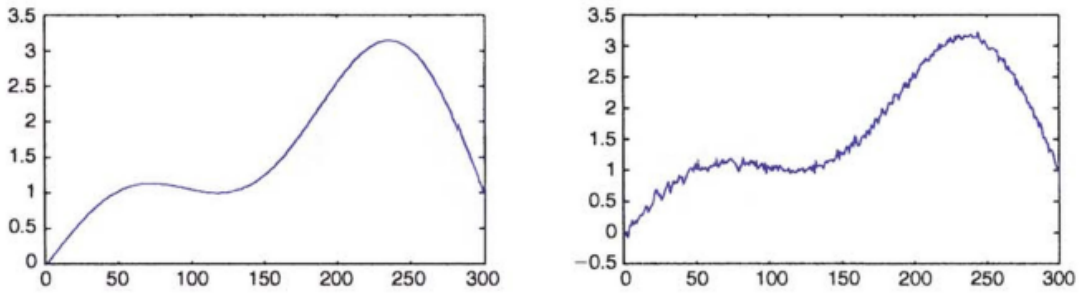


Figure 3.2. *A signal (left image) and its noisy version (right image).*

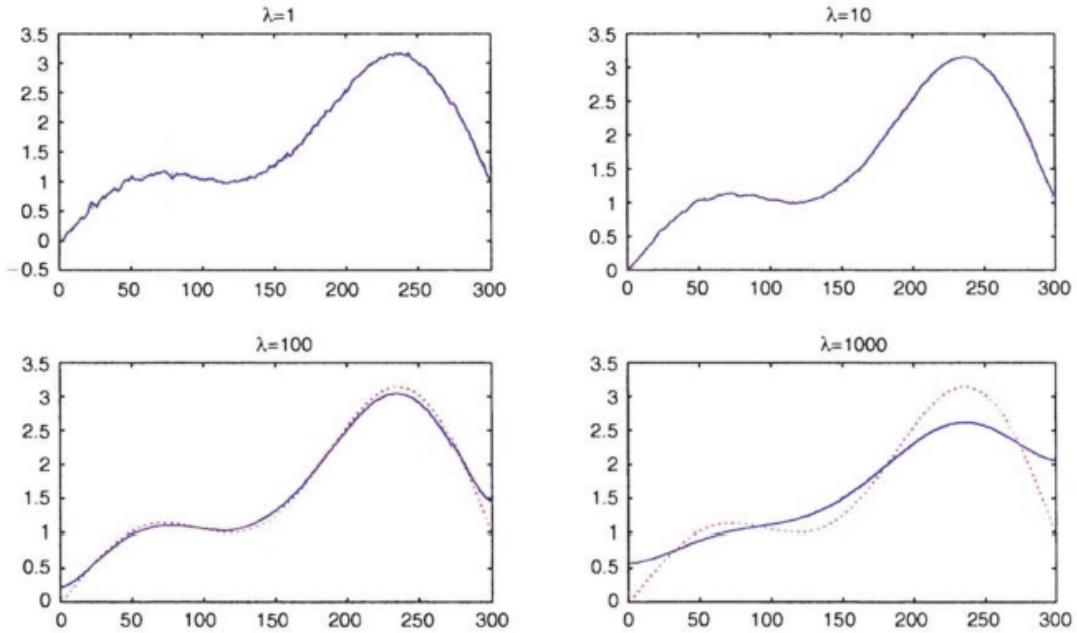


Figure 3.3. *Four reconstructions of a noisy signal by RLS solutions.*

Eliminarea zgomotului dintr-un semnal

Denoising problem

Să presupunem că este dat un semnal bruiat a unui semnal $x \in \mathbb{R}^n$:

$$b = x + w.$$

Aici x este un semnal necunoscut, w este un vector de zgomot necunoscut, iar b este vectorul măsurătorilor cunoscute.

Problema eliminării zgomotului este următoarea: Având în vedere b, găsiți o estimare "bună" a lui x.

Aplicații?

Problema celor mai mici pătrate vă va da soluția $x = b$.

Pentru a găsi o problemă mai relevantă, vom adăuga un termen de regularizare. Pentru aceasta, trebuie să exploatăm unele informații a priori despre semnal.

De exemplu, am putea ști că semnalul este neted într-un anumit sens. În acest caz, este foarte natural să adăugăm o penalizare pătratică, care este suma pătratelor diferențelor dintre componentele consecutive ale vectorului; adică funcția de regularizare este

$$R(x) = \sum_{i=1}^{n-1} (x_i - x_{i+1})^2.$$

Această funcție pătratică poate fi scrisă și sub forma $R(x) = \|L\,x\|^2$, unde $L \in \mathbb{R}^{(n-1)\times n}$ este dată de

$$L = \begin{pmatrix} 1 & -1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & -1 \end{pmatrix}$$

Problema rezultată a celor mai mici pătrate regularizate este

$$\min_{x \in \mathbb{R}^n} \|x - b\|^2 + \lambda \|L\,x\|^2,$$

iar soluția sa optimă este dată de

$$x_{\mathrm{RLS}}(\lambda) = (I_n + \lambda L^T L)^{-1} b,$$

unde $\lambda > 0$ este un parametru de regularizare dat (bun).

Am putea găsi un astfel de $\lambda > 0$?

Alte situații

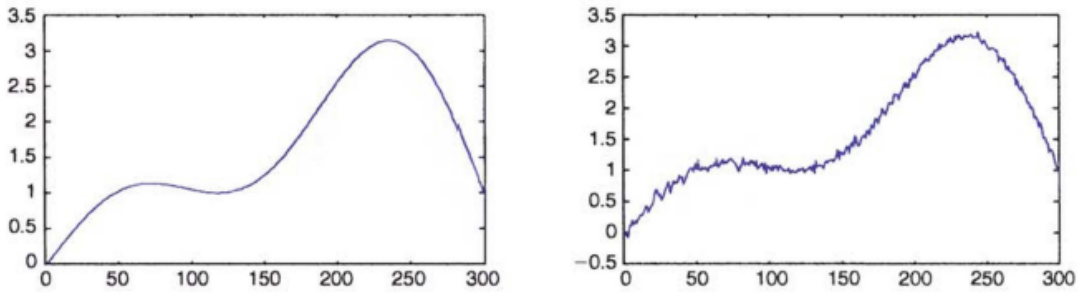


Figure 3.2. *A signal (left image) and its noisy version (right image).*

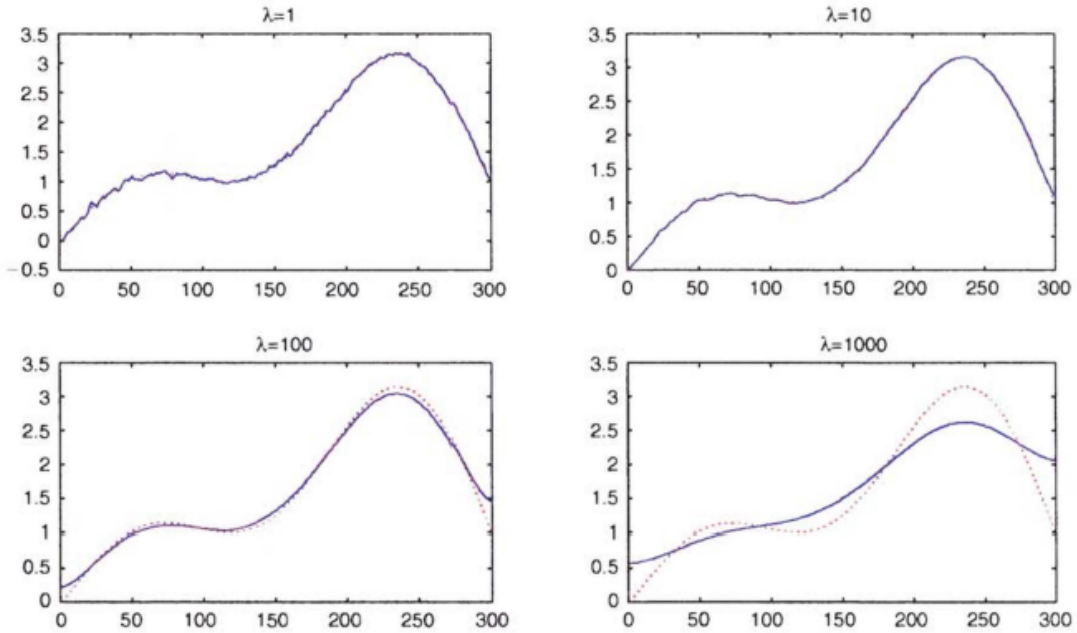


Figure 3.3. *Four reconstructions of a noisy signal by RLS solutions.*

Aplicație de pe telefonul dumneavoastră

Transmiteți pe Whatsapp sau alt canal de comunicații un mesaj vocal făcând un zgomot de fundal, de exemplu, sunetul cheilor sau al unui clopoțel. Salvați fișierul cu numele `sunet_bruiat.ogg` și folosiți codul de mai jos.

Pentru rezolvarea sistemului $(I + \lambda L^T L)\, x = b$ folosiți diverse metode de rezolvare a sistemelor (de exemplu, Cramer).

Comparați timpul în care se reconstruiește semnalul. Acesta indicua rapiditatea sau lentoarea metodei de rezolvare a sistemului.

Observați că pentru unele metode apare la final un bruiaj în plus, ceva nou! Aceasta indică eroarea cu care se rezolvua sistemul prin metoda aleasă. Practic, "auziți eroarea"! Eroarea reiese în produsul final.

Folosiți atât metode directe cât și metode iterative!

```
[b,fs] = audioread('sunet_bruiat.ogg');
%incarcam sunetul cu zgomot
%Verificam sunetul si afisam semnalul descris de acesta
sound(b, fs);
plot(b);
hold on
% Afisam lungimea pentru a intelege magnitudinea datelor
n = length(b)
% Construim matricele In si L folosind matrici rare (sparse)
%deoarece au dimensiuni
% foarte mari, dar majoritatea elementelor sunt nule
idn = speye(n);
e = ones(n-1,1);
L = spdiags([-e e], [0 1], n-1, n);
% Definim lambda (putem incepe de la o valoare mica
%si sa tot crestem)
lambda = 5;
```

```
% Folosim metoda celor mai mici patrate cu regularizare
%patratica (vezi curs) si ajungem la format
% x = inv(In + lambda*L'*L) *b, dar nu folosim aceasta forma,
%deoarece matricea este prea mare.
% Fie folosim operatorul \ care implementeaza deja cea mai
%eficienta metoda de rezolvare a unui
% sistem liniar, luand in calcul si cazul matricelor rare:
x = (speye(n) + lambda*(L'*L)) \ b;
% Ascultam semnalul filtrat:
sound(x, fs);
plot(x);
hold on
```

```
% Fie LU pe blocuri rezolvat in laboratoarele anterioare, dar
%se pierde putin din acuratete:
A = speye(n) + lambda*(L'*L);
db = floor(n/500) % numarul de blocuri (se poate modifica)
x = [];
for k = 1:499
[L,U]= FactorizareLU(A(1+(k-1)*db:k*db,1+(k-1)*db:k*db));
y = lowerSolve(L,b(1+(k-1)*db:k*db));
x1 = upperSolve(U,y);
x = [x x1'];
end
% Ascultam semnalul filtrat:
sound(x, fs);
plot(x);
legend('Original', 'Filtrat \', 'Filtrat LU blocuri');
xlabel('Sample');
ylabel('Amplitudine');
```

Problema neliniară a celor mai mici pătrate: Circle Fitting

Există situații în care ni se dă un sistem de ecuații neliniare

$$f_i(x) = c_i, \qquad i = 1, 2, ..., m,$$

unde $f_i : \mathbb{R}^n \to \mathbb{R}$, $c_i \in \mathbb{R}$ sunt date și x trebuie finanțat.

În acest caz, problema de aproximare este cea a celor mai mici pătrate neliniare (NLS), care se formulează astfel

$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^{m} (f_i(x) - c_i)^2.$$

Nu există o modalitate ușoară de a rezolva problemele NLS. Metoda Gauss-Newton este o modalitate, dar aceasta converge numai către un punct staționar.

Circle fitting

Să presupunem că ne sunt date m puncte $a_1, a_2, ..., a_m \in \mathbb{R}^n$. Problema adaptării cercului urmărește să găsească un cerc

$$C(x, r) = \{y \in \mathbb{R}^n : \|y - x\| = r\}$$

care se potrivește cel mai bine punctelor m.

Aplicație?

Ecuațiile neliniare asociate cu această problemă sunt

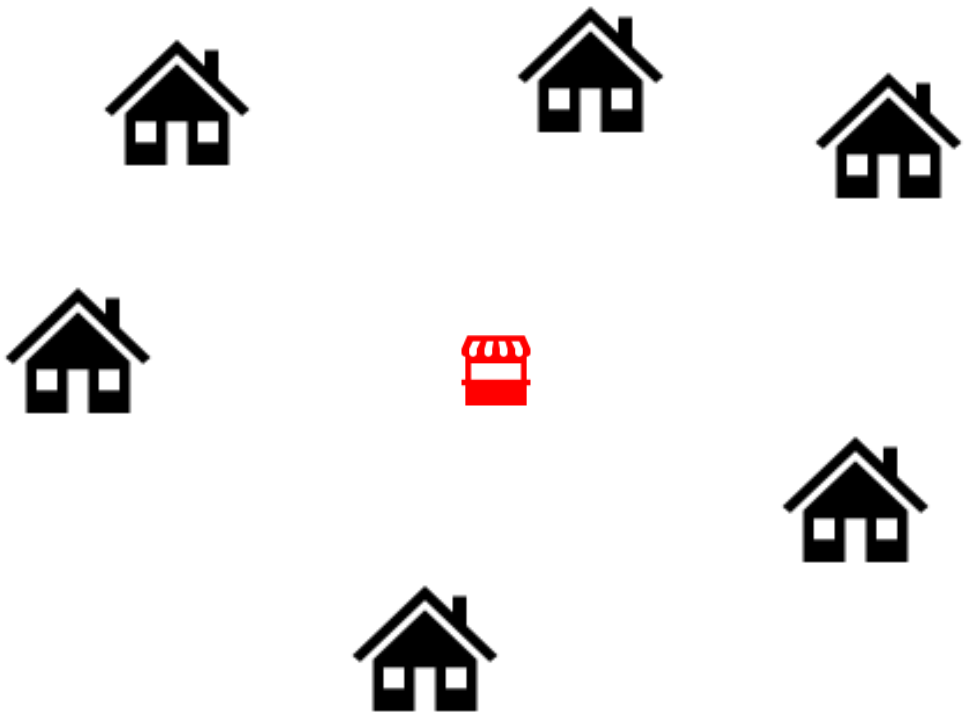
$$\|x - a_i\| = r, \qquad i = 1, 2, ..., m.$$

Deoarece dorim să avem de-a face cu funcții diferențiabile, iar funcția normă nu este diferențiabilă, vom considera versiunea pătratică a acesteia:

$$\|x - a_i\|^2 = r^2, \quad i = 1, 2, ..., m.$$

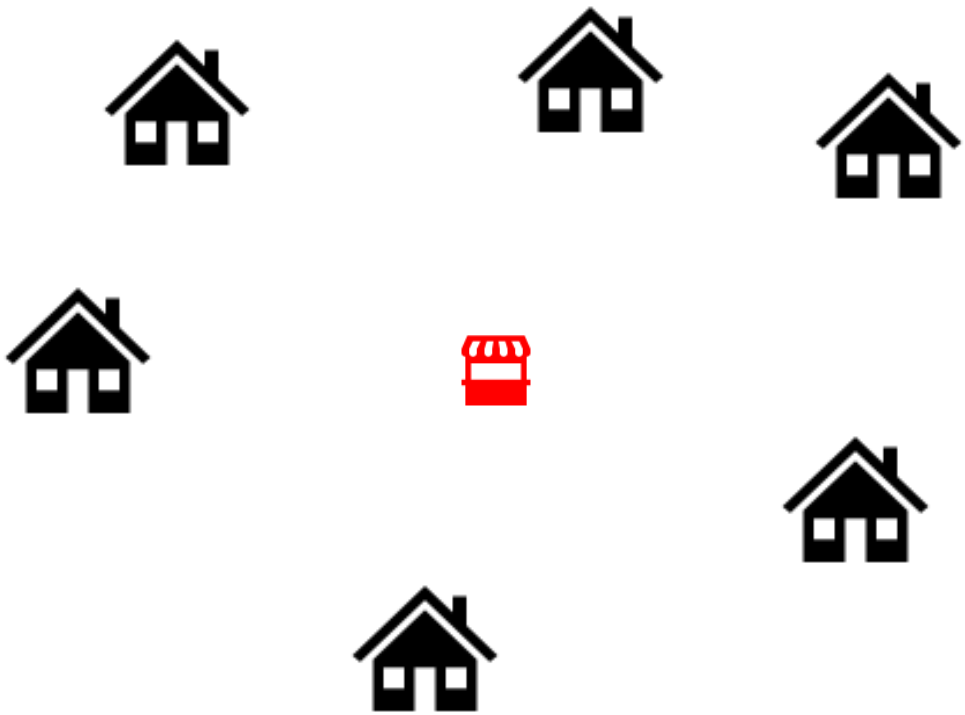
Problema locației egal depărtate

Problema locației egal depărtate



$$\min_{x \in \mathbb{R}^n, r \in \mathbb{R}} \sum_{i=1}^{m} (\|x - a_i\|^2 - r^2)^2.$$

Problema locației egal depărtate cu ponderi de importanță



$$\min_{x \in \mathbb{R}^n, r \in \mathbb{R}} \sum_{i=1}^{m} \omega_i (\|x - a_i\|^2 - r^2)^2.$$

NLS for Circle fitting

Problema NLS asociată cu aceste ecuații este

$$\min_{x\in\mathbb{R}^n, r\in\mathbb{R}_+} \sum_{i=1}^{m} (\|x - a_i\|^2 - r^2)^2.$$

Observație: În această formă nu avem o problemă de optimizare fără constrângeri!

Dar, de fapt, problema este echivalentă cu

$$\min_{x\in\mathbb{R}^n, r\in\mathbb{R}} \sum_{i=1}^{m} (-2\langle a_i, x\rangle + \|x\|^2 + \|a_i\|^2 - r^2)^2.$$

Efectuând schimbarea de variabile $R = \|x\|^2 - r^2$, problema de mai sus se reduce la

$$\min_{x\in\mathbb{R}^n, \|x\|^2 \geq R} f(x, R) := \sum_{i=1}^{m} (-2\langle a_i, x\rangle + R + \|a_i\|^2)^2.$$

De fapt, orice soluție optimă $(\widehat{x}, \widehat{R})$ a problemei

$$\min_{x\in\mathbb{R}^n, R\in\mathbb{R}} f(x,R) := \sum_{i=1}^{m}(-2\langle a_i, x\rangle + R + \|a_i\|^2)^2$$

satisface în mod automat $\|\widehat{x}\|^2 \geq \widehat{R}$, deoarece altfel

$$-2\langle a_i, \widehat{x}\rangle + \widehat{R} + \|a_i\|^2 > -2\langle a_i, \widehat{x}\rangle + \|\widehat{x}\|^2 + \|a_i\|^2 = \|\widehat{x} - a_i\|^2 \geq 0,\ i = 1, 2, ..., m.$$

Prin ridicarea la pătrat a ambelor părți ale primei inegalități din ecuația de mai sus și adunarea la i rezultă

$$\begin{aligned} f(\widehat{x}, \widehat{R}) &= \sum_{i=1}^{m}(-2\langle a_i, \widehat{x}\rangle + \widehat{R} + \|a_i\|^2)^2 \\ &> \sum_{i=1}^{m}(-2\langle a_i, \widehat{x}\rangle + \|\widehat{x}\|^2 + \|a_i\|^2)^2 = \|\widehat{x} - a_i\|^2 = f(\widehat{x}, \|\widehat{x}\|^2), \end{aligned}$$

arătând că $(\widehat{x}, \|\widehat{x}\|^2)$ conduce o valoare a funcției mai mică decât $(\widehat{x}, \widehat{R})$, în contradicție cu optimimalitatea lui $(\widehat{x}, \widehat{R})$.

În concluzie, problema NLS

$$\min_{x\in\mathbb{R}^n, \|x\|^2\geq R} f(x,R) := \sum_{i=1}^{m}(-2\langle a_i, x\rangle + R + \|a_i\|^2)^2$$

este de fapt echivalentă cu problema LS

$$\min_{(x,R)\in\mathbb{R}^{n+1}} f(x,R) := \|A\begin{pmatrix} x \\ R \end{pmatrix} - b\|^2,$$

unde $A = \begin{pmatrix} 2\,a_1^T & -1 \\ 2\,a_2^T & -1 \\ \vdots & \vdots \\ 2\,a_m^T & -1 \end{pmatrix}$ și $b = \begin{pmatrix} \|a_1\|^2 \\ \|a_2\|^2 \\ \vdots \\ \|a_m\|^2 \end{pmatrix}$.

Dacă A este de rang maxim, atunci soluția unică a problemei liniare a celor mai mici pătrate este

$$\begin{pmatrix} x \\ R \end{pmatrix} = (A^T A)^{-1} A^T b.$$

Optimul x este dat de primele n componente, iar raza r este dată de

$$r = \sqrt{\|x\|^2 - R}.$$

Matrici dreptunghiulare: Factorizarea QR

Factorizarea QR

Definiție
O matrice $A \in \mathbb{R}^{m \times n}$, cu $m \geq n$, admite o factorizare QR dacă există o matrice ortogonală $Q \in \mathbb{R}^{m \times m}$ și o matrice trapezoidală superior $R \in \mathbb{R}^{m \times n}$, cu rânduri nule începând de la al $(n+1)$-lea, astfel încât $A = QR$.

Este de asemenea posibil să se genereze o versiune redusă a factorizării QR, așa cum este afirmat în rezultatul următor.

Teoremă
Fie $A \in \mathbb{R}^{m \times n}$, cu $m \geq n$, o matrice de rang n pentru care este cunoscută o factorizare QR. Atunci există o factorizare unică a lui A de forma

$$A = \widetilde{Q}\widetilde{R},$$

unde $\widetilde{Q}$ și $\widetilde{R}$ sunt submatrice ale lui Q și R, date respectiv de

$$\widetilde{Q} = Q(1:m, 1:n), \qquad \widetilde{R} = R(1:n, 1:n).$$

Mai mult, $\widetilde{Q}$ are coloane vectoriale ortonormale și $\widetilde{R}$ este triunghiulară superior și coincide cu factorul Cholesky H al matricei simetrice definite pozitiv $A^T A$, adică, $A^T A = \widetilde{R}^T \widetilde{R}$.

Demonstrație

Putem demonstra direct existența factorizării speciale apoi să consideruam cua mulțimea vectorilor liniari independenți dați de coloanele lui $\widetilde{Q}$ este completatua până la o bazua iar R se obține adăgând linii nule.

Presupunem aplicarea algoritmului Gram–Schmidt asupra coloanelor matricei

$$A = (a_1 \mid \cdots \mid a_n).$$

Proiecția unui vector a pe direcția unui vector q se notează

$$\operatorname{proj}_q(a) = \frac{\langle q, a \rangle}{\langle q, q \rangle}\, q, \qquad \langle v, w \rangle = v^T w,$$

unde $\langle \cdot, \cdot \rangle$ reprezintă produsul scalar standard.

Demonstrație

Procedura începe cu:

$$u_1 = a_1, \qquad q_1 = \frac{u_1}{\|u_1\|},$$

iar vectorii următori se obțin prin eliminarea componentelor pe direcțiile deja determinate:

$$\begin{aligned}
u_2 &= a_2 - \operatorname{proj}_{q_1} a_2, & q_2 &= \frac{u_2}{\|u_2\|}, \\
u_3 &= a_3 - \operatorname{proj}_{q_1} a_3 - \operatorname{proj}_{q_2} a_3, & q_3 &= \frac{u_3}{\|u_3\|}, \\
&\vdots \\
u_k &= a_k - \sum_{j=1}^{k-1} \operatorname{proj}_{q_j} a_k, & q_k &= \frac{u_k}{\|u_k\|}.
\end{aligned}$$

Demonstrație

Rearanjarea acestor relații astfel încît vectorii a_i să fie pe partea stîngă conduce la:

$$
\begin{aligned}
a_1 &= q_1 \|u_1\|, \\
a_2 &= \operatorname{proj}_{q_1} a_2 + q_2 \|u_2\|, \\
a_3 &= \operatorname{proj}_{q_1} a_3 + \operatorname{proj}_{q_2} a_3 + q_3 \|u_3\|, \\
&\vdots \\
a_k &= \sum_{j=1}^{k-1} \operatorname{proj}_{q_j} a_k + q_k \|u_k\|.
\end{aligned}
$$

Demonstrație

Rescriem

$$
\begin{aligned}
a_1 &= q_1\|u_1\|,\\
a_2 &= \langle q_1, a_2\rangle q_1 + q_2\|u_2\|,\\
a_3 &= \langle q_1, a_3\rangle q_1 + \langle q_2, a_3\rangle q_2 + q_3\|u_3\|,\\
&\vdots\\
a_k &= \sum_{j=1}^{k-1}\langle q_j, a_k\rangle q_j + q_k\|u_k\|.
\end{aligned}
$$

Toate aceste relații pot fi exprimate matriceal:

$$
(q_1 \mid \ldots \mid q_n)\begin{pmatrix}
\|u_1\| & \langle q_1, a_2\rangle & \langle q_1, a_3\rangle & \cdots\\
0 & \|u_2\| & \langle q_2, a_3\rangle & \cdots\\
0 & 0 & \|u_3\| & \cdots\\
\vdots & \vdots & \vdots & \ddots
\end{pmatrix}.
$$

Demonstrație

Produsul matricial reproduce exact matricea originală A, astfel încît:

$$A = \widetilde{Q}\widetilde{R},$$

unde $\widetilde{Q} = (q_1 \mid \ldots \mid q_n)$ este ortogonală, iar $\widetilde{R}$ este triangulară superior.

Cum $\widetilde{R} = \widetilde{Q}^T A$, obținem explicit:

$$\widetilde{R} = \begin{pmatrix} \langle q_1, a_1 \rangle & \langle q_1, a_2 \rangle & \langle q_1, a_3 \rangle & \cdots \\ 0 & \langle q_2, a_2 \rangle & \langle q_2, a_3 \rangle & \cdots \\ 0 & 0 & \langle q_3, a_3 \rangle & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

Observăm că:

$$\langle q_j, a_j \rangle = \|u_j\|, \qquad \langle q_j, a_k \rangle = 0 \text{ pentru } j > k,$$

iar din relația $\widetilde{Q}\widetilde{Q}^T = I$ deducem

$$\widetilde{Q}^T = \widetilde{Q}^{-1}.$$

Dacă A are rangul n (adică, rang complet), atunci vectorii coloană ai lui $\widetilde{Q}$ formează o bază ortonormală pentru spațiul vectorial

$$\mathrm{range}(A) = \{y \in \mathbb{R}^m \,:\, y = A\,x \quad \text{pentru} \quad x \in \mathbb{R}^n\}.$$

Ca o consecință, construirea factorizării QR poate fi interpretată și ca o procedură pentru generarea unei baze ortonormale pentru un set dat de vectori.

Dacă A are rangul $r < n$, factorizarea QR nu conduce neapărat la o bază ortonormală pentru $\mathrm{range}(A)$. Totuși, se poate obține o factorizare de forma

$$Q^T A\,P = \begin{pmatrix} R_{11} & R_{12} \\ 0 & 0 \end{pmatrix},$$

unde Q este ortogonală, P este o matrice de permutare și R_{11} este o matrice triunghiulară superior nesingulară de ordin r.

În general, când folosim factorizarea QR, ne vom referi întotdeauna la forma sa redusă, deoarece are aplicație în rezolvarea sistemelor supra-determinate.

Factorii matriciali $\widetilde{Q}$ și $\widetilde{R}$ pot fi calculați utilizând ortogonalizarea Gram-Schmidt.

În continuare, impunând ca $A = \widetilde{Q}\widetilde{R}$ și folosind faptul că $\widetilde{Q}$ este ortogonală (adică, $\widetilde{Q}^T\widetilde{Q} = I_n$), elementele lui $\widetilde{R}$ pot fi calculate cu ușurință.

De asemenea, este de remarcat faptul că, dacă A are rang complet, matricea A^TA este simetrică și pozitiv definită, și, prin urmare, admite o factorizare Cholesky unică de forma H^TH. Pe de altă parte, deoarece ortogonalitatea lui $\widetilde{Q}$ implică

$$H^TH = A^TA = \widetilde{R}^T\widetilde{Q}^T\widetilde{Q}\widetilde{R} = \widetilde{R}^T\widetilde{R}, \tag{148}$$

concluzionăm că $\widetilde{R}$ este, de fapt, factorul Cholesky H al lui A^TA.

Astfel, elementele diagonale ale lui $\widetilde{R}$ sunt nenule doar dacă A are rang complet.

Metoda Gram-Schmidt modificată

Metoda Gram-Schmidt are o utilitate practică redusă, deoarece vectorii generați își pierd independența liniară din cauza erorilor de rotunjire. Deși este corectă din punct de vedere matematic, ea prezintă o dificultate majoră în practică: *este instabilă numeric*, în special atunci când vectorii sunt aproape depedenți liniar sau când erorile de rotunjire se acumulează.

Pentru a îmbunătăți stabilitatea, se utilizează *Metoda Gram–Schmidt modificată (Modified Gram–Schmidt, MGS)*, o variantă echivalentă teoretic, dar mult mai robustă.

Metoda Gram-Schmidt modificată

În forma clasică, vectorul nou

$$u_k = a_k - \sum_{j=1}^{k-1} \langle q_j, a_k \rangle q_j$$

se obține prin scăderea dintr-o singură operație a tuturor proiecțiilor pe vectorii ortonormali anteriori. Această operație implică scăderi între valori apropiate, ceea ce conduce la pierderea de precizie în aritmetica flotantă.

Metoda modificată reorganizează calculele astfel încît proiecțiile sunt eliminate *una câte una*, actualizând vectorul la fiecare pas:

$$\begin{aligned} u_k^{(0)} &= a_k, \\ u_k^{(j)} &= u_k^{(j-1)} - \langle q_j, u_k^{(j-1)} \rangle q_j, \qquad j = 1, 2, \ldots, k-1, \\ u_k &= u_k^{(k-1)}. \end{aligned}$$

Această procedură reduce riscul de anulări numerice și limitează propagarea erorilor.

Observați că nu este posibilă rescrierea factorizării QR pe matricea A. În general, matricea $\widetilde{R}$ este rescrisă pe A, în timp ce $\widetilde{Q}$ este stocată separat.

Sisteme nedeterminate cu factorizarea QR

Sisteme nedeterminate cu factorizarea QR

Teoremă
Fie $A \in \mathbb{R}^{m\times n}$, cu $m \geq n$, o matrice de rang complet. Atunci soluția unică a problemei

$$\min_{x\in\mathbb{R}^n} \underbrace{\|Ax - b\|_2^2}_{:=\Phi(x)}$$

este dată de

$$x^* = \widetilde{R}^{-1}\widetilde{Q}^T b, \tag{149}$$

unde $\widetilde{R} \in \mathbb{R}^{n\times n}$ și $\widetilde{Q} \in \mathbb{R}^{m\times n}$ sunt matricile obținute din factorizarea QR a lui A. Mai mult, minimul lui Φ este dat de

$$\Phi(x^*) = \sum_{i=n+1}^{m} [(Q^T b)_i]^2. \tag{150}$$

Demonstrație

Factorizarea QR a lui A există și este unică, deoarece A are rang complet. Astfel, există două matrici, $Q \in \mathbb{R}^{m \times m}$ și $R \in \mathbb{R}^{m \times n}$, astfel încât $A = QR$, unde Q este ortogonală.

Deoarece matricile ortogonale păstrează produsul scalar euclidian, rezultă că

$$\|Ax - b\|_2^2 = \|Rx - Q^T b\|_2^2. \tag{151}$$

Aducându-ne aminte că R este trapezoidală superior, avem

$$\|Rx - Q^T b\|_2^2 = \|\widetilde{R}x - \widetilde{Q}^T b\|_2^2 + \sum_{i=n+1}^{m} [(Q^T b)_i]^2, \tag{152}$$

așa încât minimul este atins când $x = x^*$.

Dacă A nu are rang complet, tehnicile de rezolvare de mai sus eșuează, deoarece, în acest caz, dacă x^* este o soluție a problemei

$$\min_{x \in \mathbb{R}^n} \underbrace{\|Ax - b\|_2^2}_{:=\Phi(x)},$$

atunci vectorul $x^* + z$, cu $z \in \ker(A)$, este de asemenea o soluție. Prin urmare, trebuie introdusă o constrângere suplimentară pentru a impune unicitatea soluției.

De obicei, se impune ca x^* să aibă norma euclidiană minimă, astfel încât problema celor mai mici pătrate poate fi formulată astfel:

$$\begin{aligned} &\text{găsiți } \; x^* \in \mathbb{R}^n \; \text{ cu norma euclidiană minimă astfel încât} \\ &\|A\,x^* - b\|_2^2 \leq \min_{x \in \mathbb{R}^n} \underbrace{\|Ax - b\|_2^2}_{:=\Phi(x)}. \end{aligned} \tag{153}$$

Instrumentul pentru rezolvarea acestei probleme noi este descompunerea prin valori singulare (sau SVD).

Reflectori Householder

Definiție
Fie $u \in \mathbb{R}^m$ normat, adică $\|u\| = 1$. O matrice $U \in \mathbb{R}^{m \times m}$ de forma

$$U = I_m - 2\, u\, u^T$$

se numește reflector elementat Householder de ordin m.

Matricea U are proprietățile

- $U^T U = (I_m - 2\, u\, u^T)^T (I_m - 2\, u\, u^T) = I_m - 4\, u\, u^T + 4\, u (u^T u)\, u^T = I_m$.
- $U^T = U$
- $U\, x = (I_m - 2\, u\, u^T)\, x = x - 2\, u \underbrace{(u^T\, x)}_{=\langle u, x \rangle} = x - 2 \underbrace{(u^T\, x)}_{=\langle u, x \rangle} u$

Reflectori Householder

Dacă u nu are norma 1, putem defini totuși reflectorul Householder

$$U = I_m - \frac{1}{\beta} u\ u^T, \qquad \text{unde} \qquad \beta = \frac{\|u\|^2}{2}.$$

Vom obține $U\,x = (I_m - \frac{1}{\beta}\, u\, u^T)\, x = x - \frac{1}{\beta}\, u\, \underbrace{(u^T\, x)}_{=\langle u,x\rangle} = x - \underbrace{\frac{1}{\beta}\,(u^T\, x)}_{:=\nu}\, u$

- pentru $u = (0\ \ldots\ 0\ u_k\ \ldots\ u_m)^T$, reflectorul Houseldorf corespunzător este

  $U_k = \begin{pmatrix} I_{k-1} & 0 \\ 0 & \tilde{U} \end{pmatrix}$, $\tilde{U}$ fiind reflectorul asociat vectorului $\widetilde{u} = (u_k\ \ldots\ u_m)^T$

Aplicații ale reflectorilor Householder pentru factorizarea QR

Considerăm o matrice $A \in \mathbb{R}^{m\times n}$ și dorim să construim o matrice $Q \in \mathbb{R}^{m\times m}$ ortogonală și o matrice $R \in \mathbb{R}^{m\times n}$ astfel ca $A = Q\,R$.

Dacă notăm cu a_1 prima coloană a matricei A, și definim (De ce u_1 este nenul?)

$$u_1 = a_1 - \|a_1\|\, e_1, \quad \text{unde} \quad e_1 \quad \text{este primul element al bazei canonice,}$$

construind reflectorul Householder

$$U_1 = I_m - \frac{1}{\beta_1} u_1\ u_1^T, \qquad \text{unde} \qquad \beta_1 = \frac{\|u_1\|^2}{2},$$

vom avea

$$U_1\, a_1 = \|a_1\|\, e_1.$$

Prin urmare obținem

$$A^{(1)} := U_1\, A = \begin{pmatrix} \|a_1\| & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & \cdots & \vdots \\ 0 & a_{m2}^{(1)} & \cdots & a_{mn}^{(1)} \end{pmatrix}$$

Aplicații ale reflectorilor Householder pentru factorizarea QR

La următorul pas definim

$$a_2 = \begin{pmatrix} 0 \\ A^{(1)}(2:m, 2) \end{pmatrix}, \tag{154}$$

apoi definim

$u_2 = a_2 - \|a_2\| e_2$, unde e_2 este al doilea element al bazei canonice și

construind reflectorul Householder

$$U_2 = I_m - \frac{1}{\beta_2} u_2 \; u_2^T, \qquad \text{unde} \qquad \beta_2 = \frac{\|u_2\|^2}{2}$$

vom avea

$$U_2 \, a_2 = \|a_2\| \, e_2.$$

În plus, deoarece prima linie și prima coloană din $u_2 \; u_2^T$ sunt zero, prima linie a lui $A^{(1)}$ dar și prima coloană a lui nu se modifică $A^{(1)}$ prin înmulțirea cu U_2.

Aplicații ale reflectorilor Householder pentru factorizarea QR

Prin urmare

$$A^{(2)} := U_2 A^{(1)} = U_2 U_1 A = \begin{pmatrix} \|a_1\| & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & \|a_2\| & a_{23}^{(2)} & \cdots & a_{2n}^{(2)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3n}^{(2)} \\ \vdots & \vdots & \cdots & \vdots & \\ 0 & 0 & a_{m3}^{(2)} & \cdots & a_{mn}^{(2)} \end{pmatrix}$$

Aplicații ale reflectorilor Householder pentru factorizarea QR

La pasul k definim

$$a_k = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ A^{(k-1)}(k:m,k) \end{pmatrix}, \tag{155}$$

apoi definim

$$u_k = a_k - \|a_k\| \, e_k, \quad \text{unde } e_k \text{ este elementul } k \text{ al bazei canonice}i$$

construind reflectorul Householder

$$U_k = I_m - \frac{1}{\beta_k} u_k \; u_k^T, \qquad \text{unde} \qquad \beta_k = \frac{\|u_k\|^2}{2}$$

vom avea

$$U_k \, a_k = \|a_k\| \, e_k.$$

În plus, deoarece primele $k-1$ linii și primele $k-1$ coloană din $u_k \; u_k^T$ sunt zero, primele linii ale lui $A^{(k-1)}$ dar și primele coloane ale lui nu se modifică $A^{(k-1)}$ prin înmulțirea cu U_k.

Aplicații ale reflectorilor Householder pentru factorizarea QR

Prin urmare

$$A^{(k)} := U_k A^{(k-1)} = U_k \ldots U_1 A = \begin{pmatrix} \|a_1\| & a_{12}^{(1)} & \cdots & a_{1k}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & \|a_2\| & \cdots & a_{2k}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \cdots & \cdots & \vdots & \\ 0 & 0 & \cdots & \|a_k\| & \cdots & a_{3n}^{(k)} \\ \vdots & \vdots & \cdots & \vdots & \vdots & \\ 0 & 0 & \cdots & a_{mk}^{(k)} & \cdots & a_{mn}^{(k)} \end{pmatrix}$$

Aplicații ale reflectorilor Householder pentru factorizarea QR

Repetând procedeul de n ori găsim

$$A^{(n)} := U_n A^{(n-1)} = U_n \ldots U_1 \, A = \begin{pmatrix} \|a_1\| & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & \|a_2\| & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \cdots & \\ 0 & 0 & \cdots & \|a_n\| \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \\ 0 & 0 & \cdots & 0 \end{pmatrix}.$$

Factorizarea este găsită

Deci, Q și R din factorizarea QR a lui A sunt

$$Q = U_1 .... U_{n-1} U_n \text{ și } R = \begin{pmatrix} \|a_1\| & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\ 0 & \|a_2\| & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \cdots & \\ 0 & 0 & \cdots & \|a_n\| \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \cdots & \\ 0 & 0 & \cdots & 0 \end{pmatrix}$$

Având factorizarea QR vom putea deduce factorizarea $\widetilde{Q}\widetilde{R}$ și apoi putem să le folosim pentru rezolvarea problemei celor mai mici pătrate, dacă $\operatorname{rang} A = n$.

Metoda gradientului folosită pentru rezolvarea sistemelor

Scop

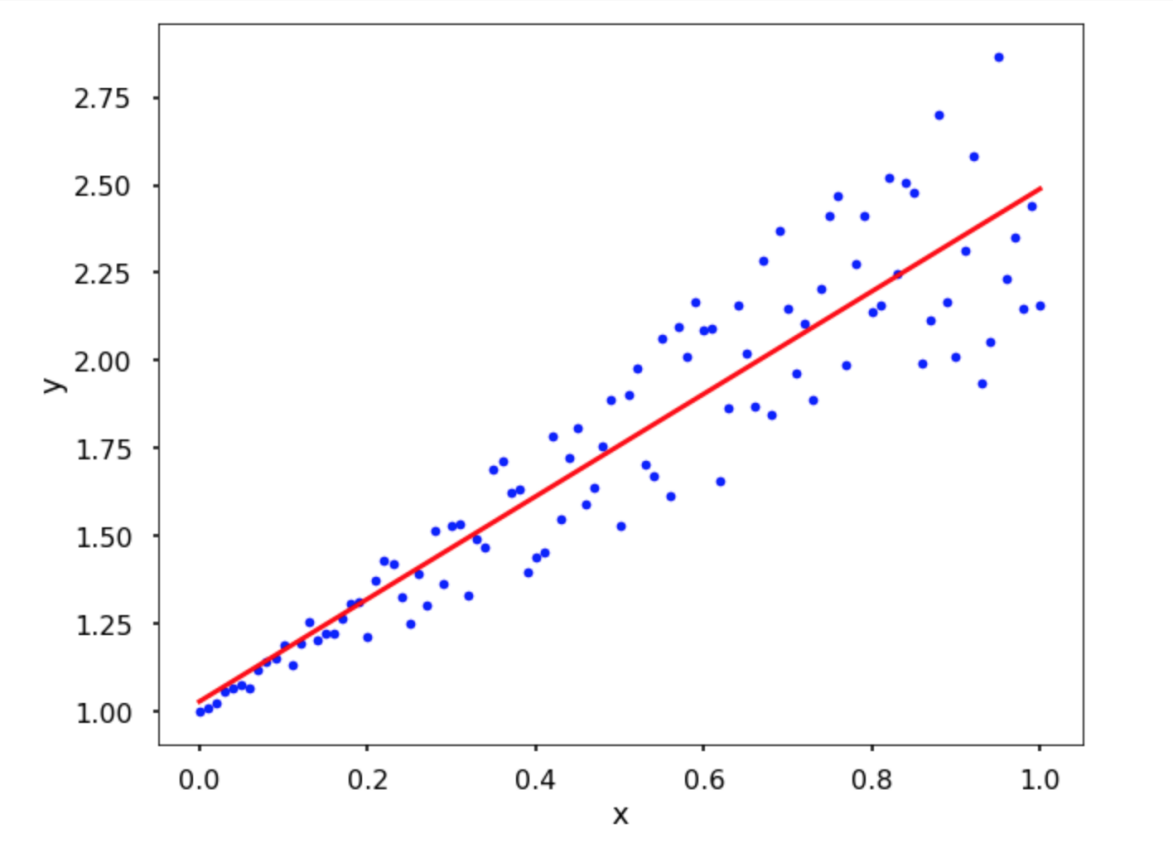
Dorim să construim o soluție aproximativă pentru problema de minim

$$\min_{x \in \mathbb{R}^n} f(x), \quad \text{unde} \quad f : \mathbb{R}^n \to \mathbb{R} \ \text{ este de clasă } \ C^1 \ \text{ pe } \ \mathbb{R}^n.$$

De ce?

Pentru că după cum am v ăzut, rezolvarea aproximativă a unui sistem de ecuații se reduce la o problemă de optimizare, adică găsirea unei valori minime a unei funcții pă tratice.

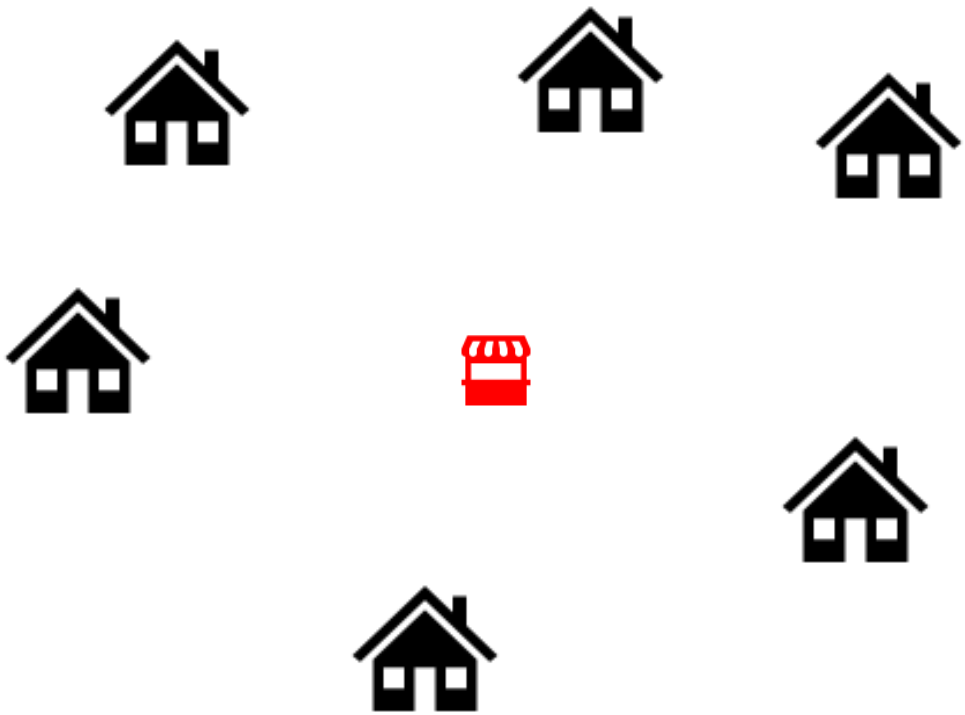
Corelarea datelor



$$f : \mathbb{R}^2 \to \mathbb{R}, f(z) = \langle Az, z \rangle + 2\langle b, z \rangle + c, \qquad A \in \mathbb{R}^{2\times 2}, \quad b \in \mathbb{R}^2, \quad c \in \mathbb{R}.$$

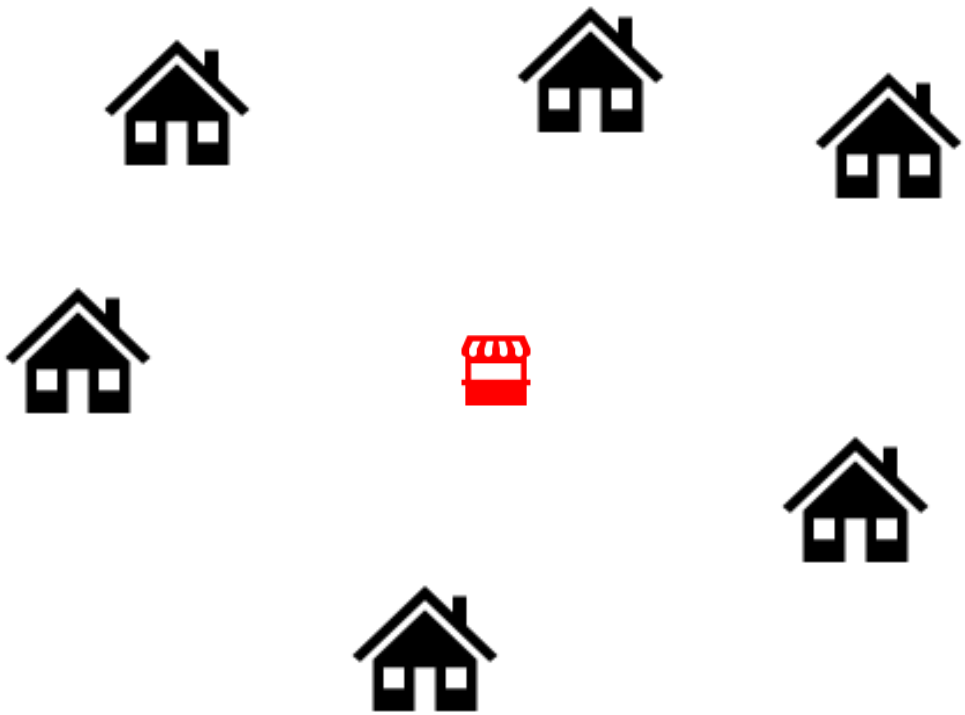
Alte probleme

Alte probleme



$$\min_{x\in\mathbb{R}^n, r\in\mathbb{R}} \sum_{i=1}^{m} (\|x - a_i\|^2 - r^2)^2.$$

Alte probleme



$$\min_{x\in\mathbb{R}^n, r\in\mathbb{R}} \sum_{i=1}^{m} \omega_i (\|x - a_i\|^2 - r^2)^2.$$

De ce aproximativ?

În majoritatea problemelor, abordările analitice obișnuite nu pot fi aplicare în practică din următoarele motive:

- ar putea fi o sarcină foarte dificilă să se rezolve sistemul de ecuații (de obicei neliniare) $\nabla f(x) = 0$;
- chiar dacă este posibilă găsirea tuturor punctelor staționare, s-ar putea să existe un număr infinit de puncte staționare, iar sarcina de a-l găsi pe cel care corespunde valorii minime a funcției este o problemă de optimizare care, prin ea însăși, ar putea fi la fel de dificilă ca și problema inițială.

Direcții de descreștere

Considerăm problema de minim

$$\min_{x \in \mathbb{R}^n} f(x), \quad \text{unde} \quad f : \mathbb{R}^n \to \mathbb{R} \ \text{ este o funcție de clasă } C^1 \ \text{ pe } \ \mathbb{R}^n.$$

În loc să încercăm să găsim expresia analitică a unui punct staționar, vom construi un algoritm iterativ pentru găsirea punctelor staționare.

Algoritmii iterativi pe care îi vom lua în considerare în această secțiune au forma $x_{k+1} = x_k + t_k\, d_k, \quad k = 0, 1, 2, ...$, unde d_k este așa-numita direcție, iar t_k este mărimea pasului.

Definiție [Direcția de descreștere]

Fie $f : \mathbb{R}^n \to \mathbb{R}$ de clasă C^1 pe $\mathbb{R}^n$. Un vector $0 \neq d \in \mathbb{R}^n$ se numește direcție de descreștere a lui f în x dacă derivata direcțională $f'(x; d)$ este negativă, ceea ce înseamnă că

$$f'(x; d) = \left\langle \nabla f(x), d \right\rangle < 0.$$

Lemma [Proprietatea de descreștere pe direcțiilor de descreștere]

Fie $f : \mathbb{R}^n \to \mathbb{R}$ o funcție de clasă C^1 pe $\mathbb{R}^n$. Să presupunem că d este o direcție de descreștere a lui f în x. Atunci există $\varepsilon > 0$ astfel încât

$$f(x + t\,d) < f(x) \qquad \text{pentru orice} \qquad t \in (0, \varepsilon].$$

Proof.

Deoarece $f'(x; d) < 0$, din definiția derivatei direcționale rezultă că

$$\lim_{t \to 0^+} \frac{f(x + t\,d) - f(x)}{t} = f'(x; d) < 0.$$

Prin urmare, există $\varepsilon > 0$ astfel încât

$$\frac{f(x + t\,d) - f(x)}{t} < 0 \qquad \text{pentru orice} \qquad t \in (0, \varepsilon],$$

ceea ce implică cu ușurință rezultatul dorit. □

Metoda direcțiilor de descreștere schematică

Algoritmul metodei direcțiilor de descreștere

- **Initializare:** Alegem $x_0 \in \mathbb{R}^n$ arbitrar;
- **Etapa generală:** Pentru orice $k = 0, 1, 2, ...,$ se
  - alege o direcție de descreștere d_k;
  - găsește o mărime a pasului t_k care să satisfacă $f(x_k + t_k d_k) < f(x_k)$;
  - Setați $x_{k+1} = x_k + t_k d_k$;
  - se verifică dacă un criteriu de oprire este satisfăcut, atunci STOP și x_{k+1} este ieșirea.

Atât de frumos și atât de neclar (doar conceptual)!

Multe detalii lipsesc din descrierea schematică de mai sus a algoritmului :

- Care este punctul de plecare?
- Cum se alege direcția de descreștere?
- Ce mărime a pasului trebuie să fie luată?
- Care este criteriul de oprire?

Clarificări

- Punctul de pornire poate fi ales arbitrar, în absența unei informații deja știute despre soluția optimă.
- Un exemplu de criteriu de oprire popular este $\|\nabla f(x_{k+1})\| \leq \varepsilon$.
- Principala diferență între diferitele metode este alegerea direcției de descreștere .
- Vom presupune că mărimea pasului este aleasă astfel încât $f(x_{k+1}) < f(x_k)$. Încă nu este clar!
  - mărime constantă a pasului: $t_k = \overline{t}, k = 0, 1, 2, ...$; Util pentru probleme simple.
    - O constantă mare poate face ca algoritmul să nu fie descrescător;
    - O constantă mică poate cauza o convergență lentă a metodei.
  - căutarea exactă pe linie: t_k este un minimizator al lui f de-a lungul razei $x_k + t\, d_k$, adică $t_k = \operatorname{argmin}_{t \geq 0} f(x_k + t\, d_k)$.
    - Pare mai atractivă la prima vedere;
    - Dar, nu este întotdeauna posibil să găsim minimizatorul exact.
  - backtracking este un compromis între ultimele două abordări;

Backtracking

Metoda necesită trei parametri: $s > 0$, $\alpha \in (0,1)$ $\beta \in (0,1)$. Alegerea lui t_k se face prin următoarea procedură. În primul rând, t_k se stabilește ca fiind egal cu presupunerea inițială s. Apoi, atâta timp cât

$$f(x_k) - f(x_k + t_k d_k) < -\alpha\, t_k \big\langle \nabla f(x_k), d_k \big\rangle,$$

se stabilește $t_k = \beta\, t_k$.

Mărimea pașilor se alege ca $t_k = s\, \beta^{i_k}$, unde i_k este cel mai mic număr întreg nenegativ pentru care se îndeplinește condiția

$$f(x_k) - f(x_k + s\, \beta^{i_k} d_k) \geq -\alpha\, s\, \beta^{i_k} \big\langle \nabla f(x_k), d_k \big\rangle$$

este satisfăcută.

Validitatea condiției suficiente de scădere

Fie $f : \mathbb{R}^n \to \mathbb{R}$ o funcție de clasă C^1 pe $\mathbb{R}^n$ și fie $x \in \mathbb{R}^n$. Să presupunem că $0 \neq d \in \mathbb{R}^n$ este o direcție de descreștere a lui f în x și fie $\alpha \in (0, 1)$. Atunci există $\varepsilon > 0$ astfel încât

$$f(x) - f(x + t\,d) \geq -\alpha\, t \left\langle \nabla f(x), d \right\rangle \quad \text{pentru orice} \qquad t \in (0, \varepsilon].$$

Proof.

Deoarece f de clasă C^1 pe $\mathbb{R}^n$ rezultă că

$$f(x + t\,d) = f(x) + t\left\langle \nabla f(x), d\right\rangle + o(t\|d\|),$$

și, prin urmare

$$f(x) - f(x + t\,d) = -\alpha\, t\left\langle \nabla f(x), d\right\rangle - (1-\alpha)\, t\left\langle \nabla f(x), d\right\rangle - o(t\|d\|),$$

Deoarece d este o direcție de descreștere a lui f la x avem

$$\lim_{t\to 0^+} \frac{(1-\alpha)\, t\left\langle \nabla f(x), d\right\rangle + o(t\|d\|)}{t} = (1-\alpha)\left\langle \nabla f(x), d\right\rangle < 0.$$

Prin urmare, există $\varepsilon > 0$ astfel încât pentru toate $t \in (0, \varepsilon]$ să existe inegalitatea $(1-\alpha)\, t\left\langle \nabla f(x), d\right\rangle + o(t\|d\|) < 0$, ceea ce conduce la rezultatul dorit.

□

Metoda gradientului

În metoda gradientului, direcția de descreștere este aleasă ca fiind

$$d_k = -\nabla f(x_k), \quad k = 0, 1, 2\ldots.$$

Aceasta este o alegere bună, deoarece

$$f'(x_k; -\nabla f(x_k)) = -\langle \nabla f(x_k), \nabla f(x_k) \rangle = -\|\nabla f(x_k)\|^2 < 0.$$

Derivata direcțională minimă între toate direcțiile normalizate

Fie $f : \mathbb{R}^n \to \mathbb{R}$ o funcție de clasă C^1 pe $\mathbb{R}^n$ și fie $x \in \mathbb{R}^n$ un punct nestaționar. Atunci o soluție optimă a

$$\min_{d \in \mathbb{R}^n} \{f'(x; d) \, : \, \|d\| = 1\}$$

este $d = -\dfrac{\nabla f(x)}{\|\nabla f(x)\|}$.

Proof.
Din moment ce $f'(x;d) = \langle \nabla f(x), d \rangle$, problema este aceeași ca și în cazul în care

$$\min_{d \in \mathbb{R}^n} \{ \langle \nabla f(x), d \rangle \ : \ \|d\| = 1 \}$$

Din inegalitatea Cauchy-Schwarz avem

$$\langle \nabla f(x), d \rangle \geq -\|\nabla f(x)\| \, \|d\| = -\|\nabla f(x)\|.$$

Astfel, $-\|\nabla f(x)\|$ este o limită inferioară a valorii optime a problemei de minimizare. Pe de altă parte, introducând $d = -\dfrac{\nabla f(x)}{\|\nabla f(x)\|}$ în funcția obiectiv obținem că

$$f'(x; -\frac{\nabla f(x)}{\|\nabla f(x)\|}) = \langle \nabla f(x), -\frac{\nabla f(x)}{\|\nabla f(x)\|} \rangle = -\|\nabla f(x)\|,$$

și astfel ajungem la concluzia că limita inferioară $-\|\nabla f(x)\|$ se atinge la $d = -\dfrac{\nabla f(x)}{\|\nabla f(x)\|}$, ceea ce implică rezultatul dorit. □

Metoda Gradient

- **Input**: $\varepsilon > 0$ ca parametru de toleranță.
- **Initializare**: Se alege $x_0 \in \mathbb{R}^n$ în mod arbitrar.
- **Etapa generală**: Pentru orice $k = 0, 1, 2, ...$ se execută următorii pași:
  - Se alege o mărime a pasului t_k printr-una dintre procedurile menționare mai sus pentru

$$g(t) = f(x_k - t\,\nabla f(x_k)).$$

  - Setați $x_{k+1} = x_k - t_k \nabla f(x_k)$.
  - Dacă $\|\nabla f(x_{k+1})\| \leq \varepsilon$, STOP și x_{k+1} este valoarea de OUTPUT

Metoda gradientului se poate comporta destul de rău. Ca un exemplu, să considerăm problema de minimizare

$$\min_{x,y\in\mathbb{R}^n} x^2 + \frac{1}{100}\, y^2,$$

și să presupunem că utilizăm metoda gradientului cu vectorul inițial $(\frac{1}{100}, 1)^T$.

Aceasta este o problemă importantă, un răspuns parțial poate fi găsit folosind noțiunea de număr de condiționare.

Se consideră problema de minimizare pătratică

$$\min_{x\in\mathbb{R}^n}\{f(x) := \langle Ax, x\rangle\}, \quad \text{unde} \quad A \succ 0.$$

Soluția optimă este, evident, $x^* = 0$. Metoda gradientului cu pas exact are forma

$$x_{k+1} = x_k + t_k d_k,$$

unde $d_k = -2\,A\,x_k$ este gradientul lui f în x_k, iar pasul t_k este găsit ca fiind

$$t_k = \frac{\|d_k\|^2}{2\langle A\,d_k, d_k\rangle}.$$

Deci,

$$\begin{aligned} f(x_{k+1}) &= \langle A\,x_{k+1}, x_{k+1}\rangle = \langle A\,(x_k + t_k d_k), (x_k + t_k d_k)\rangle \\ &= \langle A\,x_k, x_k\rangle + 2\,t_k\langle A\,x_k, d_k\rangle + t_k^2\langle A\,d_k, d_k\rangle \\ &= \langle A\,x_k, x_k\rangle - 2\,t_k\langle d_k, d_k\rangle + t_k^2\langle A\,d_k, d_k\rangle. \end{aligned}$$

Introducând în ultima relație expresia pentru t_k dată mai sus, obținem că

$$\begin{aligned} f(x_{k+1}) &= \langle A\,x_k, x_k\rangle - \frac{1}{4}\frac{\langle d_k, d_k\rangle^2}{\langle A\,d_k, d_k\rangle} = \langle A\,x_k, x_k\rangle\left(1 - \frac{1}{4}\frac{\langle d_k, d_k\rangle^2}{\langle A\,d_k, d_k\rangle\langle A\,x_k, x_k\rangle}\right) \\ &= f(x_k)\left(1 - \frac{\langle d_k, d_k\rangle^2}{\langle A\,d_k, d_k\rangle\langle A^{-1}\,d_k, d_k\rangle}\right) \end{aligned}$$

Inegalitatea lui Kantorovici

Fie A o matrice $n \times n$ pozitiv definită. Atunci pentru orice $0 \neq x \in \mathbb{R}^n$ are loc inegalitatea

$$\frac{\langle x, x\rangle^2}{\langle Ax, x\rangle\langle A^{-1}x, x\rangle} \geq \frac{4\,\lambda_{\max}\lambda_{\min}}{(\lambda_{\max}+\lambda_{\min})^2}.$$

Se notează $m = \lambda_{\min}$ și $M = \lambda_{\max}$. Valorile proprii ale matricei $A + M\,m\,A^{-1}$ sunt $\lambda_i + \frac{M\,m}{\lambda_i}$, $i = 1, ..., n$. Este ușor de demonstrat că maximul funcției unidimensionale $\varphi(t) = t + \frac{M\,m}{t}$ pe $[m, M]$ este atins în punctele m și M cu o valoare corespunzătoare a funcției φ de $M + m$ și, prin urmare, din moment ce $m \leq \lambda_i(A) \leq M$, rezultă că valorile proprii ale lui $A + M\,m\,A^{-1}$ sunt mai mici decât $(M + m)$. Astfel,

$A + M\,m\,A^{-1} \preceq (M + m)I_n$ în sensul că $A + M\,m\,A^{-1} - (M + m)I_n \preceq 0$

Adică.

$$\langle A\,x, x\rangle + M\,m\,\langle A^{-1}\,x, x\rangle \leq (M+m)\langle x, x\rangle,$$

care, combinată cu inegalitatea simplă $\alpha\,\beta \leq \frac{1}{4}(\alpha+\beta)^2\ \forall\,\alpha, \beta \in \mathbb{R}$ rezultă

$$\begin{aligned}\langle A\,x, x\rangle\,M\,m\,\langle A^{-1}\,x, x\rangle &\leq \frac{1}{4}[\langle A\,x, x\rangle + M\,m\,\langle A^{-1}\,x, x\rangle]^2\\ &\leq \frac{(M+m)^2}{4}\langle x, x\rangle^2,\end{aligned}$$

care, după o simplă rearanjare a termenilor, conduce la rezultatul dorit.

Revenind la analiza ratei de convergență a metodei gradientului, rezultă, folosind inegalitatea lui Kantorovici, că

$$f(x_{k+1}) \leq \left(1 - \frac{4\,m\,M}{(M+m)^2}\right) f(x_k) = \left(\frac{M-m}{M+m}\right)^2 f(x_k),$$

unde $M = \lambda_{\max}(A)$, $m = \lambda_{\min}(A)$.

Rezumând, avem

Analiza ratei de convergență pentru funcții pătratice

Fie x_k șirul generat de metoda gradientului cu pas constant pentru rezolvarea problemei de minimizare pătratică

$$\min_{x \in \mathbb{R}^n} \{f(x) := \langle A\,x, x\rangle\}, \quad \text{unde} \quad A \succ 0.$$

Atunci, pentru orice $k = 0, 1, \ldots$

$$f(x_{k+1}) \leq \left(\frac{M-m}{M+m}\right)^2 f(x_k),$$

unde $M = \lambda_{\max}(A)$, $m = \lambda_{\min}(A)$.

Aceasta implică faptul că pentru orice $k = 0, 1, ...$

$$f(x_{k+1}) \leq c^k \, f(x_0), \quad \text{where } c = \left(\frac{M-m}{M+m} \right)^2 = \left(\frac{\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} - 1}{\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} + 1} \right)^2 .$$

Numărul $\kappa = \dfrac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$ se numește *număr de condiționare* al lui A.

Matricele cu un număr mare de condițioare se numesc *rău condiționate*, iar matricele cu un număr mic de condiționare (aproape de 1) se numesc *bine condiționate*.

Rescalare

Rescalare

Problema matricelorr prost condiționate este una majoră și au fost dezvoltate multe metode pentru a o evita. Una dintre cele mai populare abordări constă în "condiționarea" problemei prin efectuarea unei transformări liniare corespunzătoare a variabilelor.

Mai precis, să considerăm problema de minimizare fără constrângeri

$$\min_{x \in \mathbb{R}^n} f(x).$$

Pentru o matrice nesingulară dată $S \in \mathbb{R}^{n \times n}$, efectuăm transformarea liniară $x = S\,y$ și obținem problema echivalentă

$$\min_{y \in \mathbb{R}^n} g(y) := f(S\,y).$$

Deoarece $\nabla g(y) = S^T \nabla f(S\,y) = S^T \nabla f(x)$, rezultă că metoda gradientului aplicată problemei transformate ia forma

$$y_{k+1} = y_k - t_k\, S^T \nabla f(S\,y_k).$$

Din moment ce $\nabla g(y) = S^T \nabla f(S\,y) = S^T \nabla f(x)$, rezultă că metoda gradientului aplicată problemeei transformate ia forma

$$y_{k+1} = y_k - t_k\, S^T \nabla f(S\,y_k).$$

Înmulțind această ultimă egalitate cu S la stânga și folosind notația $x_k = S\,y_k$, obținem formula recursivă

$$x_{k+1} = x_k - t_k\, S\,S^T \nabla f(x_k).$$

Definind $D = S\,S^T$, obținem următoarea versiune a metodei gradientului, pe care o numim metoda gradientului scalat cu matrice de scalare D (pozitiv definită):

$$x_{k+1} = x_k - t_k\, D \nabla f(x_k).$$

Direcția $-D\nabla f(x_k)$ este o direcție de descreștere a lui f în x_k atunci când $\nabla f(x_k) \neq 0$, deoarece

$$f'(x_k; -D\nabla f(x_k)) = -\left\langle \nabla f(x_k), D\nabla f(x_k) \right\rangle < 0.$$

Pentru a rezuma discuția de mai sus, am arătat că metoda gradientului scalat cu matricea de scalare D este echivalentă cu metoda gradientului utilizată pentru funcția $g(y) = f(D^{1/2}y)$.

Aici $S = D^{1/2}$ înseamnă că $S^T S = D$.

Analiza convergenței metodei gradientului (Opțional)

Vom prezenta o analiză a convergenței metodei gradientului utilizată pentru problema de minimizare fără restricții.

$$\min_{x \in \mathbb{R}^n} f(x).$$

Vom presupune că funcția obiectiv f este de clasă C^1 și că gradientul său ∇f este Lipschitz continuu pe $\mathbb{R}^n$, ceea ce înseamnă că

$$\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\| \quad \text{pentru orice} \quad x, y \in \mathbb{R}^n.$$

Rețineți că dacă ∇f este Lipschitz cu constanta L, atunci este de asemenea Lipschitz cu constanta $\widetilde{L} \geq L$. Prin urmare, există un număr infinit de constante Lipschitz pentru o funcție cu gradient Lipschitz.

Clasa funcțiilor cu gradient Lipschitz cu constanta L este notată cu $C_L^{1,1}(\mathbb{R}^n)$ sau pur și simplu $C_L^{1,1}$.

- Funcții liniare: $f(x) = \langle a, x \rangle$ cu $L = 0$.
- Funcții pătratice: $f(x) = \langle A\,x, x \rangle + 2\langle b, x \rangle + c$ cu $L = 2\,\|A\|$, deoarece

$$\|\nabla f(x) - \nabla f(y)\| \leq 2\,\|A\,x - A\,y\| \leq 2\,\|A\|\,\|x - y\|.$$

Propoziție

Fie f o funcție de clasă C^2 pe $\mathbb{R}^n$. Atunci următoarele două afirmații sunt echivalente

(a) $f \in C_L^{1,1}(\mathbb{R}^n)$.

(b) $\|\nabla^2 f(x)\| \leq L$ pentru orice $x \in \mathbb{R}^n$, unde $\|\cdot\|$ reprezintă norma spectrală.

(b)$\rightarrow$ (a). Să presupunem că $\|\nabla^2 f(x)\| \leq L$ pentru orice $x \in \mathbb{R}^n$. Atunci, prin teorema fundamentală a calculului integral, avem pentru orice $x, y \in \mathbb{R}^n$

$$\begin{aligned}\nabla f(y) &= \nabla f(x) + \int_0^1 \nabla^2 f(x + t(y-x))\,(y-x)dt \\ &= \nabla f(x) + \left(\int_0^1 \nabla^2 f(x + t(y-x))dt\right)\,(y-x),\end{aligned}$$

Deci

$$\begin{aligned}\|\nabla f(y) - \nabla f(x)\| =& \left\| \left(\int_0^1 \nabla^2 f(x + t(y-x)) dt \right) (y-x) \right\| \\ \leq& \left\| \int_0^1 \nabla^2 f(x + t(y-x)) dt \right\| \|y-x\| \\ \leq& \int_0^1 \|\nabla^2 f(x + t(y-x))\| dt \, \|y-x\| \\ \leq& L \|y-x\|,\end{aligned}$$

care demonstrează rezultatul dorit, adică $f \in C_L^{1,1}$.

(a)$\rightarrow$ (b). Să presupunem acum că $f \in C_L^{1,1}$. Atunci, prin teorema fundamentală a calculului integral, avem pentru toți $d \in \mathbb{R}^n$ și $\alpha > 0$

$$\nabla f(x + \alpha d) = \nabla f(x) + \int_0^\alpha \nabla^2 f(x + t\, d)\, d\, dt$$

Astfel,

$$\| \int_0^\alpha \nabla^2 f(x + t\, d)\, dt\, d \| = \| \nabla f(x + \alpha d) - f(x) \| \leq \alpha\, L\, \|d\|.$$

Împărțind cu α și luând $\alpha \rightarrow 0^+$, obținem

$$\|\nabla^2 f(x)\, d\| \leq L \|d\|,$$

ceea ce implică faptul că $\|\nabla^2 f(x)\,\| \leq L$.

Lema descreșterii

Un rezultat important pentru funcțiile $C_L^{1,1}$ este acela că acestea pot fi mărginite superior de o funcție pătratică pe întregul spațiu.

Propoziție [Lema descreșterii]

Fie $f \in C_L^{1,1}(\mathbb{R}^n)$. Atunci, pentru orice $x, y \in \mathbb{R}^n$

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2}\|x - y\|^2.$$

Din teorema fundamentală a calculului integral avem

$$f(y) - f(x) = \int_0^1 \langle \nabla f(x + t(y - x)), y - x \rangle \, dt.$$

Prin urmare,

$$f(y) - f(x) = \langle \nabla f(x), y - x \rangle + \int_0^1 \langle \nabla f(x + t(y - x)) - \nabla f(x), y - x \rangle \, dt.$$

Deci,

$$\|f(y) - f(x) - \langle \nabla f(x), y - x \rangle\| = \left| \int_0^1 \langle \nabla f(x + t(y-x)) - \nabla f(x), y - x \rangle \, dt \right|$$

$$\leq \int_0^1 |\langle \nabla f(x + t(y-x)) - \nabla f(x), y - x \rangle| \, dt$$

$$\leq \int_0^1 \|\nabla f(x + t(y-x)) - \nabla f(x)\| \, \|y - x\| \, dt$$

$$\leq \int_0^1 t \, L \, \|y - x\|^2 \, dt = \frac{L}{2} \|y - x\|.$$

Rețineți că demonstrația lemei descreșterii arată de fapt atât limitele superioare, cât și cele inferioare ale funcției:

$$f(x)+\langle \nabla f(x), y-x\rangle-\frac{L}{2}\|y-x\|^2 \leq f(y) \leq f(x)+\langle \nabla f(x), y-x\rangle+\frac{L}{2}\|y-x\|^2.$$

Lema scăderii suficiente

Lema scăderii suficiente

Să presupunem că $f \in C_L^{1,1}(\mathbb{R}^n)$. Atunci, pentru orice $x \in \mathbb{R}^n$ și $t > 0$

$$f(x) - f(x - t\,\nabla f(x)) \geq t\left(1 - \frac{L\,t}{2}\right)\|\nabla f(x)\|^2.$$

Proof.

Prin lema descreșterii avem

$$\begin{aligned} f(x - t\,\nabla f(x)) &\leq f(x) - t\,\|\nabla f(x)\|^2 + \frac{L\,t^2}{2}\|\nabla f(x)\|^2 \\ &= f(x) - t\left(1 - \frac{L\,t}{2}\right)\|\nabla f(x)\|^2 \end{aligned}$$

□

Scopul nostru este acum să arătăm că există pași viabili pentru fiecare dintre strategiile de selectare a mărimii pașilor:

- pas constant;
- căutarea exactă pe linie;
- backtracking.

În cazul unui pas constant, presupunem că $t_k = \overline{t} \in (0, \frac{2}{L})$. Înlocuind $x = x_k$, $t = \overline{t}$ în lema de scădere suficientă rezultă inegalitatea

$$f(x_k) - f(x_{k+1}) \geq \overline{t} \left(1 - \frac{L\,\overline{t}}{2}\right) \|\nabla f(x_k)\|^2.$$

Rețineți că descreștere în metoda gradientului pe iterație este

$$\overline{t} \left(1 - \frac{L\,\overline{t}}{2}\right) \|\nabla f(x_k)\|^2$$

Dacă dorim să obținem cea mai mare limită garantată a scăderii, atunci căutăm maximul lui $\bar{t}\left(1-\frac{L\,\bar{t}}{2}\right)$ în $(0,\frac{2}{L})$. Acest maxim este atins pentru $\bar{t}=\frac{1}{L}$ și, prin urmare, o alegere potrivită pentru mărimea pasului este $\frac{1}{L}$.

În acest caz

$$f(x_k)-f(x_{k+1})\geq\frac{1}{L}\left(1-\frac{L\,\frac{1}{L}}{2}\right)\|\nabla f(x_k)\|^2\geq\frac{1}{2\,L}\|\nabla f(x_k)\|^2.$$

În cadrul pasului constrant, formula iterativă a algoritmului este

$$x_{k+1} = x_k - t_k \nabla f(x_k),$$

unde $t_k = \operatorname{argmin}_{t \geq 0} f(x_k - t \nabla f(x_k))$.

Prin definiția lui t_k știm că

$$f(x_k - t_k \nabla f(x_k)) \leq f(x_k - \frac{1}{L} \nabla f(x_k)),$$

și astfel avem

$$f(x_k) - f(x_{k+1}) \geq f(x_k) - f(x_k - t_k \nabla f(x_k)) \geq \frac{1}{2L} \|\nabla f(x_k)\|^2.$$

Aceeași estimare ca și în cazul mărimii constante a pașilor.

În cazul backtracking căutăm pasul t_k suficient de mic astfel încât

$$f(x_k) - f(x_k - \frac{t_k}{\beta}\nabla f(x_k)) < \alpha\frac{t_k}{\beta}\,\|\nabla f(x_k)\|^2.$$

Înlocuind $x = x_k$, $t = \frac{t_k}{\beta}$ în lema scăderii suficiente obținem că

$$f(x_k) - f(x_k - \frac{t_k}{\beta}\nabla f(x_k)) \geq \frac{t_k}{\beta}\,\left(1 - \frac{L\,t_k}{2\,\beta}\right)\,\|\nabla f(x_k)\|^2$$

care, combinat cu estimarea de mai sus, implică faptul că

$$\frac{t_k}{\beta}\,\left(1 - \frac{L\,t_k}{2\,\beta}\right)\,\|\nabla f(x_k)\|^2 < \alpha\frac{t_k}{\beta}\|\nabla f(x_k)\|^2$$

ceea ce este același lucru cu

$$t_k > \frac{2(1-\alpha)\beta}{L}.$$

În general, obținem că în cadrul backtracking-ului avem

$$t_k > \min\{s, \frac{2(1-\alpha)\beta}{L}\},$$

ceea ce implică faptul că

$$f(x_k) - f(x_k - t_k\nabla f(x_k)) \geq \alpha\,\min\{s, \frac{2(1-\alpha)\beta}{L}\}\|\nabla f(x_k)\|^2.$$

Lemma

Fie $f \in C_L^{1,1}(\mathbb{R}^n)$. Fie $\{x_k\}_{k\geq 0}$ șirul generat de metoda gradientului pentru rezolvarea $\min_{x\in\mathbb{R}^n} f(x)$ cu una dintre următoarele strategii găsire a pașilor:

- pas constant $\bar{t} \in (0, \frac{2}{L})$,
- pas exact,
- backtracking cu parametrii $s \in (0, \infty)$, $\alpha \in (0, 1)$ și $\beta \in (0, 1)$.

Atunci

$$f(x_k) - f(x_{k+1}) \geq M\|\nabla f(x_k)\|^2,$$

unde

$$M = \begin{cases} \bar{t}\left(1 - \frac{\bar{t}L}{2}\right) & \text{pas constant,} \\ \frac{1}{2L} & \text{pas exact,} \\ \alpha \min\{s, \frac{2(1-\alpha)\beta}{L}\} & \text{backtracking.} \end{cases}$$

Convergența metodei gradientului

Propoziție

Fie $f \in C_L^{1,1}(\mathbb{R}^n)$. Fie $\{x_k\}_{k\geq 0}$ șirul generat de metoda gradientului pentru rezolvarea $\min_{x\in\mathbb{R}^n} f(x)$ cu una dintre următoarele strategii de găsire a pașilor:

- pas constant $\overline{t} \in (0, \frac{2}{L})$,
- pas exact,
- backtracking cu parametrii $s \in (0,\infty)$, $\alpha \in (0,1)$ și $\beta \in (0,1)$.

Să presupunem că f este mărginit inferior pe $\mathbb{R}^n$, adică există $m \in \mathbb{R}$ astfel încât $f(x) > m$ pentru orice $x \in \mathbb{R}^n$. Atunci,

a) șirul $\{f(x_k)\}_{k\geq 0}$ este descrescător. În plus, pentru orice $k \geq 0$, $f(x_{k+1}) < f(x_k)$, cu excepția cazului în care $\nabla f(x_k) = 0$.

b) $\nabla f(x_k) \to 0$ pentru $k \to \infty$.

Proof.

a) Din lema anterioară avem că

$$f(x_k) - f(x_{k+1}) \geq M\|\nabla f(x_k)\|^2 \geq 0,$$

pentru o anumită constantă $M > 0$ și, prin urmare, egalitatea $f(x_k) = f(x_{k+1})$ poate avea loc numai atunci când $\nabla f(x_k) = 0$.

b) Deoarece șirul $\{f(x_k)\}_{k\geq 0}$ este descrescător și mărginit inferior, deci converge. Astfel, în particular $f(x_k) - f(x_{k+1}) \to 0$ când $k \to \infty$, ceea ce, combinat cu inegalitatea de mai sus, implică faptul că $\|\nabla f(x_k)\| \to 0$ pe măsură ce $k \to \infty$.

□

Rata de convergență a normelor de gradient (Opțional)

Propoziție

În condițiile propoziției anterioare, fie f^* limita șirului $\{f(x_k)\}_{k\geq 0}$. Atunci, pentru orice $n = 0, 1, 2, \ldots$.

$$\min_{k=0,1,\ldots,n} \|\nabla f(x_k)\| \leq \sqrt{\frac{f(x_0) - f^*}{M(n+1)}},$$

unde

$$M = \begin{cases} \bar{t}\left(1 - \frac{\bar{t}L}{2}\right) & \text{dimensiunea pasului constant,} \\ \frac{1}{2L} & \text{pas exact,} \\ \alpha \min\{s, \frac{2(1-\alpha)\beta}{L}\} & \text{backtracking.} \end{cases}$$

Proof.
Prin adunarea inegalităților

$$f(x_k) - f(x_{k+1}) \geq M\|\nabla f(x_k)\|^2,$$

pentru $k = 0, 1, ..., n$, obținem

$$f(x_0) - f(x_{n+1}) \geq M\sum_{k=0}^{n}\|\nabla f(x_k)\|^2,$$

Deoarece $f(x_{n+1}) \geq f^*$, putem astfel concluziona că

$$f(x_0) - f^* \geq M\sum_{k=0}^{n}\|\nabla f(x_k)\|^2.$$

În final, folosind această ultimă inegalitate împreună cu faptul că pentru fiecare $k = 0, 1, ..., n$ avem inegalitatea evidentă
$\|\nabla f(x_k)\|^2 \geq \min_{k=0,1,...,n}\|\nabla f(x_k)\|^2$, rezultă că

$$f(x_0) - f^* \geq M(n+1)\min_{k=0,1,...,n}\|\nabla f(x_k)\|^2.$$

□

The Gauss–Newton Method

The Gauss—Newton Method: Nonlinear Least Squares

There are situations in which we are given a system of nonlinear equations

$$f_i(x) = c_i, \quad i = 1, 2, ..., m,$$

where $f_i : \mathbb{R}^n \to \mathbb{R}$, $c_i \in \mathbb{R}$ are given and x is to be funded.

In this case, the approximation problem is as in the following

Nonlinear least squares (NLS) problem

NLS is formulated as

$$\min_{x \in \mathbb{R}^n} g(x) := \sum_{i=1}^{m} (f_i(x) - c_i)^2.$$

There is no easy way to solve NLS problems. Gauss-Newton method is an way.

We will assume that f_i, $i = 1, 2, ..., m$ are continuously differentiable over $\mathbb{R}$ and $c_i \in \mathbb{R}$. The problem is sometimes also written in the terms of the function

$$F(x) = \begin{pmatrix} f_1(x) - c_1 \\ f_2(x) - c_2 \\ \vdots \\ f_m(x) - c_m \end{pmatrix},$$

and then it takes the form

$$\min_{x \in \mathbb{R}^n} \|F(x)\|^2.$$

THE general step of the Gauss-Newton method goes as follows: given the kth iterate x_k, the next iterate is chosen to minimize the sum of squares of the linear approximations of f_i at x_k, that is,

$$x_{k+1} = \operatorname{argmin}_{x \in \mathbb{R}^n} \left\{ \sum_{i=1}^{m} [f_i(x_k) + \langle \nabla f_i(x_k), x - x_k \rangle - c_i]^2 \right\}.$$

The minimization problem above is essential a linear least squares problem

$$\min_{x \in \mathbb{R}^n} \|A_k\, x - b_k\|^2,$$

where

$$A_k = \begin{pmatrix} \nabla f_1(x_k)^T \\ \nabla f_2(x_k)^T \\ \vdots \\ \nabla f_m(x_k)^T \end{pmatrix} = J(x_k)$$

is the so-called Jacobian matrix and

$$b_k = \begin{pmatrix} \langle \nabla f_1(x_k), x_k \rangle - f_1(x_k) + c_1 \\ \langle \nabla f_2(x_k), x_k \rangle - f_2(x_k) + c_2 \\ \vdots \\ \langle \nabla f_m(x_k), x_k \rangle - f_m(x_k) + c_m \end{pmatrix} = J(x_k)x_k - F(x_k).$$

The underlying assumption is of course that $J(x_k)$ is of a full columns rank. In that case, we can also write explicit expression for the Gauss-Newton iterates

$$x_{k+1} = (J(x_k)^T J(x_k))^{-1} J(x_k)^T b_k.$$

Note that the method can also be written as

$$\begin{aligned} x_{k+1} &= (J(x_k)^T J(x_k))^{-1} J(x_k)^T (J(x_k)x_k - F(x_k)) \\ &= x_k - (J(x_k)^T J(x_k))^{-1} J(x_k)^T F(x_k). \end{aligned}$$

The Gauss-Newton direction is therefore $d_k = (J(x_k)^T J(x_k))^{-1} J(x_k)^T F(x_k)$. Noting that $\nabla g(x) = 2\, J(x)^T F(x)$, we can conclude that

$$d_k = \frac{1}{2}(J(x_k)^T J(x_k))^{-1} \nabla g(x_k),$$

meaning that the Gauss-Newton method is essentially a scaled gradient method with the following positive definite scaling matrix

$$D_k = \frac{1}{2}(J(x_k)^T J(x_k))^{-1}.$$

Damped Gauss-Newton Method

This fact also explains why Gauss-Newton method is a descent direction method. The method described so far is also called the pure Gauss-Newton method since no stepsize is involved. To transform this method into a practical algorithm, a stepsize is introduced, leading to the damped Gauss-Newton method.

Damped Gauss-Newton Method

- **Input**: $\varepsilon > 0$ as the tolerance parameter.
- **Initialization**: Pick $x_0 \in \mathbb{R}^n$ arbitrarily.
- **General step**: For any $k = 0, 1, 2, \ldots$ execute the following steps:
  - Set $d_k = (J(x_k)^T J(x_k))^{-1} J(x_k)^T F(x_k)$.
  - Pick a stepsize t_k by a line search procedure on the function

$$h(t) = g(x_k - t\, d_k).$$

  - Set $x_{k+1} = x_k - t_k\, d_k$.
  - If $\|\nabla g(x_{k+1})\| \leq \varepsilon$, the STOP and x_{k+1} is the output.

Sisteme nedeterminate cu descompunerea în valori singulare (SVD) și pseudoinversă

Descompunerea în valori singulare (SVD)

Descompunerea în valori singulare (SVD) a unei matrice A este un instrument foarte util în contextul problemei celor mai mici pătrate. Este de asemenea foarte utilă pentru analiza proprietăților unei matrice. Cu SVD-ul, poți „radiografia" o matrice!

Teoremă
Fie $A \in \mathbb{R}^{m \times n}$. Atunci există matrice ortogonale $U \in \mathbb{R}^{m \times m}$ și $V \in \mathbb{R}^{n \times n}$ și o matrice diagonală $\Sigma = diag(\sigma_1, \ldots, \sigma_n) \in \mathbb{R}^{m \times n}$ cu $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$, astfel încât:

$$A = U \Sigma V^\top$$

are loc.

$$A = U\Sigma V^\top$$

Definiție
Vectorii coloană ai lui $U = [u_1, \ldots, u_m]$ sunt numiți vectori singulari stângi și, similar, $V = [v_1, \ldots, v_n]$ sunt vectorii singulari drepți. Valorile $\sigma_i = \sqrt{\lambda_i(A^\top A)}$ sunt numite valorile singulare ale lui A (unde $\lambda_i(A^\top A)$ sunt valorile proprii ale lui $A^\top A$).

Exemplu

Fie matricea $A = \begin{pmatrix} 1 & 3 \\ 5 & 7 \\ 9 & 1 \end{pmatrix} \in \mathbb{R}^{3\times 2}$. Descompunerea sa este dată de

$$\underbrace{\begin{pmatrix} 1 & 3 \\ 5 & 7 \\ 9 & 1 \end{pmatrix}}_{=A} = \underbrace{\begin{pmatrix} 0.207621 & 0.370412 & 0.905366 \\ 0.679634 & 0.611049 & -0.405854 \\ 0.703556 & -0.699581 & 0.124878 \end{pmatrix}}_{:=U\in\mathbb{R}^{3\times 3}} \underbrace{\begin{pmatrix} 11.6522 & 0. \\ 0. & 5.4979 \\ 0. & 0. \end{pmatrix}}_{:=\Sigma\in\mathbb{R}^{3\times 2}} \underbrace{\begin{pmatrix} 0.852871 & 0.522122 \\ -0.522122 & 0.852871 \end{pmatrix}}_{=V^T\in\mathbb{R}^{2\times 2}},$$

valorile singulare fiind $\sigma_1 = 11.6522$ și $\sigma_2 = 5.4979$.

Din numărul valorilor singulare nenule ne putem da seama că rangul matricei este 2.

Exemplu

Fie matricea $A = \begin{pmatrix} 1 & 5 & 9 \\ 3 & 7 & 1 \end{pmatrix} \in \mathbb{R}^{3\times 2}$. Descompunerea sa este dată de

$$\underbrace{\begin{pmatrix} 1 & 5 & 9 \\ 3 & 7 & 1 \end{pmatrix}}_{=A} = \underbrace{\begin{pmatrix} 0.852871 & -0.522122 \\ 0.522122 & 0.852871 \end{pmatrix}}_{:=U\in\mathbb{R}^{2\times 2}} \underbrace{\begin{pmatrix} 11.6522 & 0. & 0. \\ 0. & 5.4979 & 0. \end{pmatrix}}_{:=\Sigma\in\mathbb{R}^{3\times 2}} \underbrace{\begin{pmatrix} 0.207621 & 0.370412 & 0.905366 \\ 0.679634 & 0.611049 & -0.405854 \\ 0.703556 & -0.699581 & 0.124878 \end{pmatrix}}_{=V^T\in\mathbb{R}^{3\times 3}},$$

valorile singulare fiind $\sigma_1 = 11.6522$ și $\sigma_2 = 5.4979$.

Din numărul valorilor singulare nenule ne putem da seama că rangul matricei este 2.

Demonstrație

Considerăm matricea

$$B := A^T A \in \mathbb{R}^{n \times n}.$$

- B este simetrică:

$$B^T = (A^T A)^T = A^T (A^T)^T = A^T A = B.$$

- B este pozitiv semidefinită:

$$x^T B x = x^T A^T A x = (Ax)^T (Ax) = \|Ax\|^2 \geq 0, \quad \forall x \in \mathbb{R}^n.$$

Deci toate valorile proprii λ_i ale lui B sunt nenegative: $\lambda_i \geq 0$.

Demonstrație

Prin teorema spectrală pentru matrici simetrice, există o matrice ortogonală $V \in \mathbb{R}^{n \times n}$ și o matrice diagonală Λ, astfel încît

$$B = A^T A = V \Lambda V^T, \quad \Lambda = \operatorname{diag}(\lambda_1, \ldots, \lambda_n), \ \lambda_i \geq 0.$$

Notăm coloanele lui V cu

$$V = (v_1 \mid v_2 \mid \cdots \mid v_n),$$

unde vectorii v_i sunt ortonormali și satisfac

$$B v_i = \lambda_i v_i.$$

Demonstrație

Punem

$$\sigma_i := \sqrt{\lambda_i} \geq 0.$$

Acestea vor fi valorile singulare ale lui A.

Presupunem că exact r dintre valorile proprii sunt strict pozitive:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_r > 0, \quad \lambda_{r+1} = \cdots = \lambda_n = 0.$$

Pentru $i = 1, \ldots, r$ definim

$$u_i := \frac{1}{\sigma_i} A v_i.$$

Norma lui u_i:

$$\|u_i\|^2 = u_i^T u_i = \left(\frac{1}{\sigma_i} A v_i\right)^T \left(\frac{1}{\sigma_i} A v_i\right) = \frac{1}{\sigma_i^2} v_i^T A^T A v_i = \frac{1}{\sigma_i^2} v_i^T B v_i.$$

Deoarece $B v_i = \lambda_i v_i$, rezultă

$$\|u_i\|^2 = \frac{1}{\sigma_i^2} \lambda_i v_i^T v_i = \frac{\lambda_i}{\sigma_i^2} \cdot 1 = \frac{\lambda_i}{\lambda_i} = 1.$$

Deci $\|u_i\| = 1$.

Demonstrație

Ortogonalitatea: pentru $i \neq j$,

$$u_i^T u_j = \left(\frac{1}{\sigma_i} A v_i\right)^T \left(\frac{1}{\sigma_j} A v_j\right) = \frac{1}{\sigma_i \sigma_j} v_i^T A^T A v_j = \frac{1}{\sigma_i \sigma_j} v_i^T B v_j.$$

Folosing $Bv_j = \lambda_j v_j$,

$$u_i^T u_j = \frac{1}{\sigma_i \sigma_j} \lambda_j v_i^T v_j = \frac{\lambda_j}{\sigma_i \sigma_j} \cdot 0 = 0,$$

deoarece $v_i^T v_j = 0$ pentru $i \neq j$.

Astfel, $u_1, \ldots, u_r$ sunt ortonormali în $\mathbb{R}^m$.

Demonstrație

Dacă $r < m$, completăm mulțimea $\{u_1, \ldots, u_r\}$ la o bază ortonormală a lui $\mathbb{R}^m$ adăugând vectori $u_{r+1}, \ldots, u_m$ (de exemplu, prin Gram–Schmidt).

Definim

$$U := (u_1 \mid u_2 \mid \cdots \mid u_m) \in \mathbb{R}^{m \times m},$$

care este ortogonală: $U^T U = I_m$.

Deja avem V ortogonală:

$$V = (v_1 \mid \cdots \mid v_n), \quad V^T V = I_n.$$

Demonstrație

Pentru $i \leq r$, prin definiție:

$$Av_i = \sigma_i u_i.$$

Pentru $i > r$, $\lambda_i = 0$, deci

$$Bv_i = A^T A v_i = 0.$$

Atunci

$$\|Av_i\|^2 = (Av_i)^T (Av_i) = v_i^T A^T A v_i = v_i^T B v_i = 0,$$

deci $Av_i = 0$.

Prin urmare:

$$Av_i = \begin{cases} \sigma_i u_i, & i \leq r, \\ 0, & i > r. \end{cases}$$

Demonstrație

Considerăm produsul

$$AV = A(v_1 \mid \cdots \mid v_n) = (Av_1 \mid \cdots \mid Av_n).$$

Din relațiile de mai sus, aceasta este

$$AV = (\sigma_1 u_1 \mid \cdots \mid \sigma_r u_r \mid 0 \mid \cdots \mid 0).$$

Definim $\Sigma \in \mathbb{R}^{m \times n}$ ca fiind matricea diagonală dreptunghiulară

$$\Sigma = \begin{pmatrix} \sigma_1 & & & & & & \\ & \sigma_2 & & & & & \\ & & \ddots & & & & \\ & & & \sigma_r & & & \\ & & & & 0 & & \\ & & & & & \ddots & \\ & & & & & & 0 \end{pmatrix},$$

unde diagonala are $\sigma_1, \ldots, \sigma_r$, iar restul elementelor sunt 0.

Demonstrație

Atunci produsul

$$U\Sigma = (u_1 \mid \cdots \mid u_m)\Sigma$$

are coloana i egală cu:

- $\sigma_i u_i$, dacă $i \lambda_e r$;
- 0, dacă $i > r$.

Deci coloanele lui AV coincid cu coloanele lui $U\Sigma$, de unde

$$AV = U\Sigma.$$

Înmulțind la dreapta cu V^T (folosind că $V^T = V^{-1}$) obținem

$$A = U\Sigma V^T,$$

ceea ce doream să demonstrăm.

Valori și vectori singulari

Proprietăți:

- $A\,v_i = \sigma_i u_i$ și $A^T\,u_i = \sigma_i v_i$ pentru $i = 1 : n$, unde u_i și v_i sunt coloanele matricelor U și V din descompunerea spectrală.
- $\sigma_{\min}\|x\|_2 \leq \|A\,x\|_2 \leq \sigma_{\max}\|x\|_2$.
- $A = \sum_{i=1}^{r} \sigma_i u_i\, v_i^T$.
- $A^T A\,v_i = \sigma_i^2 v_i$ și $AA^T\,u_i = \sigma_i^2 u_i$ pentru $i = 1 : n$, ceea ce înseamnă că v_i e vector propriu pentru $A^T A$ iar u_i e vector propriu pentru $A\,A^T$.

Exemplul 1: matrice cu mai multe linii decât coloane ($m > n$)

Luăm

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 0 & 0 \end{pmatrix} \in \mathbb{R}^{3\times 2}.$$

Calculăm

$$A^T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \end{pmatrix}, \quad A^T A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 4 \end{pmatrix}.$$

Valorile proprii ale lui $A^T A$ sunt $\lambda_1 = 1$ și $\lambda_2 = 4$. Valorile singulare sunt

$$\sigma_1 = \sqrt{1} = 1, \quad \sigma_2 = \sqrt{4} = 2.$$

Calculăm vectorii proprii și matricea V

Pentru $\lambda_1 = 1$, un vector propriu este $v_1 = (1, 0)^T$. Pentru $\lambda_2 = 4$, un vector propriu este $v_2 = (0, 1)^T$.

Deci putem lua

$$V = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I_2.$$

Calculăm vectorii u_i și matricea U

Definim

$$u_1 = \frac{1}{\sigma_1} A v_1 = A v_1 = A \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix},$$

$$u_2 = \frac{1}{\sigma_2} A v_2 = \frac{1}{2} A \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

Observăm că u_1 și u_2 sunt deja ortonormali. Completăm la o bază ortonormală adăugând

$$u_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Astfel

$$U = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = I_3.$$

Scriem matricea Σ

Σ are dimensiunea 3×2 și diagonala (σ_1, σ_2):

$$\Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 0 & 0 \end{pmatrix}.$$

Verificarea lui $A = U\Sigma V^T$

Aici $U = I_3$, Σ este chiar A, iar $V = I_2$, deci

$$U\Sigma V^T = I_3 \cdot \Sigma \cdot I_2 = \Sigma = A.$$

Am obținut descompunerea SVD:

$$A = U\Sigma V^T = I_3 \begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 0 & 0 \end{pmatrix} I_2.$$

Exemplul 2: matrice cu mai puține linii decât coloane ($m < n$)

Luăm

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \end{pmatrix} \in \mathbb{R}^{2\times 3}.$$

Calculul lui $A^T A$

$$A^T = \begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 0 & 0 \end{pmatrix}, \quad A^T A = \begin{pmatrix} 1 & 0 \\ 0 & 2 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

Valorile proprii:

$$\lambda_1 = 1, \quad \lambda_2 = 4, \quad \lambda_3 = 0.$$

Valorile singulare:

$$\sigma_1 = 1, \quad \sigma_2 = 2, \quad \sigma_3 = 0.$$

Vectorii proprii și matricea V

Pentru $\lambda_1 = 1$, putem lua $v_1 = (1, 0, 0)^T$. Pentru $\lambda_2 = 4$, luăm $v_2 = (0, 1, 0)^T$. Pentru $\lambda_3 = 0$, luăm $v_3 = (0, 0, 1)^T$.

Deci

$$V = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = I_3.$$

Vectorii u_i și matricea U

Pentru $i = 1, 2$,

$$u_1 = \frac{1}{\sigma_1} A v_1 = A \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix},$$

$$u_2 = \frac{1}{\sigma_2} A v_2 = \frac{1}{2} A \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 0 \\ 2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

Aceștia sunt deja ortonormali în $\mathbb{R}^2$. Nu mai avem nevoie de alți vectori, deci

$$U = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = I_2.$$

Matricea Σ

Σ este de dimensiune 2×3, cu diagonala $(\sigma_1, \sigma_2) = (1, 2)$:

$$\Sigma = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \end{pmatrix}.$$

Verificarea lui $A = U\Sigma V^T$

Aici

$$U = I_2, \quad \Sigma = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \end{pmatrix}, \quad V = I_3.$$

Deci

$$U\Sigma V^T = I_2 \cdot \Sigma \cdot I_3 = \Sigma = A.$$

Am obținut din nou descompunerea SVD:

$$A = U\Sigma V^T.$$

Aceste două exemple arată modul în care se aplică teorema descompunerii spectrale pentru matrici dreptunghiulare atât în cazul $m > n$, cât și în cazul $m < n$.

Pentru Matlab În Matlab există două variante pentru calculul SVD:

`[U S V ]=svd(A)` – dă o descompunere completă

`[U S V ]=svd(A,0)` – dă o matrice $m \times n$ pentru U

Apelul $svd(A, 0)$ calculează o versiune între una completă și una economică cu o matrice nepătratică $U \in \mathbb{R}^{n \times m}$. Această formă este uneori numită „SVD subțire".

În ceea ce privește rangul, dacă

$$\sigma_1 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_p = 0. \tag{156}$$

atunci rangul lui A este r, nucleul lui A este generat de vectorii coloană $V(:, r+1:n)$ ai lui V, iar range(A) este general de vectorii coloană $U(:, 1:r)$ ai lui U.

Definiție
Să presupunem că $A \in \mathbb{R}^{m \times n}$ are rang egal cu r și că admite SVD de tipul $U^T A V = \Sigma$. Matricea $A^\dagger = V \Sigma^\dagger U^T$ este numită matricea pseudo-inversă Moore-Penrose, unde

$$\Sigma^\dagger = \operatorname{diag}\left(\frac{1}{\sigma_1}, \cdots, \frac{1}{\sigma_r}, 0, \cdots, 0\right). \tag{157}$$

Matricea $A^\dagger$ este, de asemenea, numită inversa generalizată a lui A. Într-adevăr, dacă $\operatorname{rank}(A) = n < m$, atunci $A^\dagger = (A^T A)^{-1} A^T$, iar dacă $n = m = \operatorname{rank}(A)$, $A^\dagger = A^{-1}$.

Revenind la sisteme

Dacă A nu are rang complet, tehnicile de soluționare prin descompunerea QR de mai sus nu funcționează și avem nevoie de o altă tehnică

Teoremă
Să considerăm $A \in \mathbb{R}^{m \times n}$ cu SVD dat de $A = U\Sigma V^T$. Atunci soluția unică a problemei de minimizare

$$\text{găsește } x^* \in \mathbb{R}^n \text{ cu norma Euclidiană minimă astfel că}$$
$$\|Ax^* - b\|_2^2 \leq \min_{x \in \mathbb{R}^n} \underbrace{\|Ax - b\|_2^2}_{:=\Phi(x)} \tag{158}$$

este

$$x^* = A^\dagger b, \tag{159}$$

unde $A^\dagger$ este pseudo-inversa lui A.

Demonstrație

Folosind SVD-ul lui A, sarcina este de a găsi $w = V^T$ astfel ca w are norma Euclidiană minimă și

$$\|\Sigma w - U^T b\|_2^2 \leq \|\Sigma y - U^T b\|_2^2 \qquad \forall\ y \in \mathbb{R}^n. \tag{160}$$

Dacă r este numărul de valori singulare nenule σ_i ale lui A, atunci

$$\|\Sigma w - U^T b\|_2^2 = \sum_{i=1}^{r} \left(\sigma_i w_i - (U^T b)_i\right)^2 + \sum_{i=r+1}^{m} \left((U^T b)_i\right)^2, \tag{161}$$

ceea ce este minim daca $w_i = (U^T b)_i / \sigma_i$ pentru $i = 1, \ldots, r$.

În plus, este clar că printre vectorii w ai lui $\mathbb{R}^n$ care au primele r componente fixe, vectorul cu norma Euclidiană minimă are celelalte $n - r$ componente egale cu zero.

Așadar, vectorul soluție este $w^* = \Sigma^\dagger U^T b$, adică $x^* = V\Sigma^\dagger U^T b = A^\dagger b$, unde $\Sigma^\dagger$ este matricea diagonală definită în definiția pseudo-inversei.

Aplicații ale descompunerii în valori singulare (SVD) în compresia imaginilor

Descompunerea în valori singulare (SVD)

Descompunerea în valori singulare (SVD) a unei matrice A nu este un instrument foarte util doar în contextul problemei celor mai mici pătrate ci este folosit deseori în practicua.

În continuare vom indica câteva metode simple și directe de aplicare în comprimarea imaginilor.

Ne aducem aminte.

Teoremă
Fie $A \in \mathbb{R}^{m \times n}$. Atunci există matrice ortogonale $U \in \mathbb{R}^{m \times m}$ și $V \in \mathbb{R}^{n \times n}$ și o matrice diagonală $\Sigma = diag(\sigma_1, \ldots, \sigma_n) \in \mathbb{R}^{m \times n}$ cu $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$, astfel încât:

$$A = U \Sigma V^\top$$

are loc.

$$A = U\Sigma V^\top$$

Definiție

Vectorii coloană ai lui $U = [u_1, \ldots, u_m]$ sunt numiți vectori singulari stângi și, similar, $V = [v_1, \ldots, v_n]$ sunt vectorii singulari drepți. Valorile $\sigma_i = \sqrt{\lambda_i(A^\top A)}$ sunt numite valorile singulare ale lui A (unde $\lambda_i(A^\top A)$ sunt valorile proprii ale lui $A^\top A$).

În ceea ce privește rangul, dacă

$$\sigma_1 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_p = 0. \tag{162}$$

atunci rangul lui A este r, nucleul lui A este generat de vectorii coloană $V(:, r+1:n)$ ai lui V, iar range(A) este generat de vectorii coloană $U(:, 1:r)$ ai lui U.

Matricea A se poate scrie ca o sumă de matrice

$$A = \sum_{i=1}^{r} \sigma_i u_i v_i^T = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ldots + \sigma_r u_r v_r^T + 0 + 0 + \ldots + 0.$$

Tensor de ordin 2

Vom folosi în continuare noțiunea de tensor de ordin 2.

Prin produsul tensorial a doi vectori $\overline{\xi} \in \mathbb{R}^n$, $\overline{\eta} \in \mathbb{R}^m$, notat $\overline{\xi} \otimes \overline{\eta} \in \mathbb{R}^{n \times m}$, înțelegem un operator liniar

$$\overline{\xi} \otimes \overline{\eta} : \mathbb{R}^m \to \mathbb{R}^n, \quad (\overline{\xi} \otimes \overline{\eta}) \cdot \overline{v} = \langle \overline{\eta}, \overline{v} \rangle \, \overline{\xi} \, . \tag{163}$$

Fiind un operator liniar, un tensor este un obiect matematic care poate fi caracterizat prin elementele unei baze din domeniu și a unei baze din codomeniu.

Prin urmare în baze diferite, vom avea caracterizări diferite.

Fie o bază ortonormată $\overline{e}_i$, $i = 1, 2, ..., n$ în $\mathbb{R}^n$ și o bază ortonormată $\overline{f}_j$, $j = 1, 2, ..., n$ în $\mathbb{R}^m$.

Din

$$\overline{\xi} \otimes \overline{\eta} = \sum_{i=1}^{n} \sum_{j=1}^{m} \xi_i \eta_j \overline{e}_i \otimes \overline{f}_j$$

rezultă că

$$\{\overline{e}_i \otimes \overline{f}_j \,|\, i = 1, 2, ..., n, j = 1, 2, ..., m\}$$

este un sistem de generatori pentru mulțimea tuturor produselor tensoriale

$$\{\overline{\xi} \otimes \overline{\eta} \,|\, \xi \in \mathbb{R}^n, \quad \eta \in \mathbb{R}^m\}.$$

De fapt, este ușor de arătat că

$$\{\overline{e}_i \otimes \overline{f}_j \,|\, i = 1, 2, ..., n, j = 1, 2, ..., m\}$$

este o bază pe spațiul vectorial al tuturor produselor tensoriale

$$\{\overline{\xi} \otimes \overline{\eta} \,|\, \xi \in \mathbb{R}^n, \quad \eta \in \mathbb{R}^m\}.$$

Tensor de ordin 2

Fie o bază ortonormată $\overline{e}_i$, $i = 1, 2, ..., n$ în $\mathbb{R}^n$ și o bază ortonormată $\overline{f}_j$, $j = 1, 2, ..., n$ în $\mathbb{R}^m$.

Pentru cele necesare nouă, prin tensor de ordin 2 vom înțelege un obiect matematic I definit prin

$$I = \sum_{i=1}^{n} \sum_{j=1}^{m} I_{ij} (\overline{e}_i \otimes \overline{f}_j),$$

iar I_{ij} se vor numi componentele tensorului I relativ la bazele $\overline{e}_i$ și $\overline{f}_j$.

Tensorul de ordin 2 se poate identifica cu matricea sa într-o pereche de baze. ATENȚIE! Dacă schimbăm una dintre baze, atunci matricea cu care se ideentifică tensorul se SCHIMBĂ după regulile pentru operatori liniari!

Considerând doi vectori $\overline{\xi} = \sum_{i=1}^{n} \xi_i \overline{e}_i \in \mathbb{R}^n$, $\overline{\eta} = \sum_{j=1}^{m} \eta_j \overline{f}_j \in \mathbb{R}^m$ *produsul tensorial* al lor este complet definit de matricea operatorului liniar.

Deoarece

$$\overline{\xi} \otimes \overline{\eta} = \sum_{i=1}^{n} \sum_{j=1}^{m} \xi_i \eta_j \overline{e}_i \otimes \overline{f}_j$$

și

$$(\overline{\xi} \otimes \overline{\eta}).\overline{f}_k = \sum_{i=1}^{n} \sum_{j=1}^{m} \xi_i \eta_j (\overline{e}_i \otimes \overline{f}_j).\overline{f}_k = \sum_{i=1}^{n} \sum_{j=1}^{m} \xi_i \eta_j \langle \overline{f}_j, \overline{f}_k \rangle \overline{e}_i = \sum_{i=1}^{n} \xi_i \eta_k \overline{e}_i$$

matricea operatorului în bazele considerate are componentele

$$(\overline{\xi} \otimes \overline{\eta})_{ij} = \xi_i \, \eta_j .$$

Dar $\xi_i \, \eta_j$ sunt chiar componentele matricei produs $\overline{\xi} \cdot \overline{\eta}^T$.

Avem

$$(\overline{\xi} \otimes \overline{\eta})_{ij} = \xi_i \, \eta_j.$$

Dar $\xi_i \, \eta_j$ sunt chiar componentele matricei produs $\overline{\xi} \cdot \overline{\eta}^T$.

Să ne întoarcem acum la descompunerea în valori singulare și la scrierea matricei A ca

$$A = \sum_{i=1}^{r} \sigma_i u_i \, v_i^T = \sigma_1 u_1 \, v_1^T + \sigma_2 u_2 \, v_2^T + \ldots + \sigma_r u_r \, v_r^T + 0 + 0 + \ldots + 0.$$

Folosind terminologia de la tensori, putem scrie

Scrierea matricei ca sumă de tensori

Prin urmare, identificând produsul tensorial cu matricea sa în baza canonică, avem

$$A = \sum_{i=1}^{r} \sigma_i u_i \otimes v_i = \sigma_1 u_1 \otimes v_1 + \sigma_2 u_2 \otimes v_2 + \ldots + \sigma_r u_r \otimes v_r.$$

Scrierea matricei ca sumă de tensori

Prin urmare, identificând produsul tensorial cu matricea sa în baza canonică, avem

$$A = \sum_{i=1}^{r} \sigma_i u_i \otimes v_i = \sigma_1 u_1 \otimes v_1 + \sigma_2 u_2 \otimes v_2 + \ldots + \sigma_r u_r \otimes v_r.$$

Mai mult, deoarece u_i și v_i sunt vectori ortonormați în $\mathbb{R}^n$, respectiv, $\mathbb{R}^m$, putem spune că scriere

$$A = \sum_{i=1}^{r} \sigma_i u_i \otimes v_i$$

ne oferă o descompunere a matricei A în spațiul tensorilor de ordin 2.

Să mai remarcăm că rangul matricelor asociate oricărui produs tensorial $\overline{\xi} \otimes \overline{\eta}$ este 1, scriem $\mathrm{rank}(\overline{\xi} \otimes \overline{\eta}) = 1$.

Deci, în scrierea

$$A = \sum_{i=1}^{r} \sigma_i u_i \otimes v_i = \sigma_1 u_1 \otimes v_1 + \sigma_2 u_2 \otimes v_2 + \ldots + \sigma_r u_r \otimes v_r$$

avem descompunerea lui A în funție de o bază formată din tensori de rang 1.

Mai mult, se poate arata că pentru orice $1 < k \geq r$ avem

$$\operatorname{rank}(\sigma_1 u_1 \otimes v_1 + \sigma_2 u_2 \otimes v_2 + \ldots + \sigma_k u_k \otimes v_k) = k.$$

Ne întrebăm dacă nu putem **COMPRIMA** matricea A folosind doar o parte din termenii sumei. POATE DOAR PÂNĂ CÂND σ_i începe să nu mai conteze. Ce înseamnă oare asta?

Deoarece am scris valorile singulare ordonate descrescător, aceasta ar însemna că se aproximează matricea A cu o matrice de rang k.

Dar care o fi cea mai bună aproximare a sa?

Aproximarea cu o matrice de rang k

Fie A o matrice $m \times n$ de rang r și fie

$$A = U\Sigma V^\top$$

o descompunere în valori singulare (SVD) a lui A. Notăm cu u_i coloanele lui U, cu v_i coloanele lui V, și cu

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_p, \quad p = \min(m, n),$$

valorile singulare ale lui A.

Atunci, matricea de rang $k < r$ cea mai apropiată de A (în norma spectrală $\|\cdot\|_2$) este dată de

$$A_k = \sum_{i=1}^{k} \sigma_i u_i \otimes v_i^\top = U \operatorname{diag}(\sigma_1, \ldots, \sigma_k, 0, \ldots, 0)\, V^\top,$$

și

$$\|A - A_k\|_2 = \sigma_{k+1}.$$

Proof.
Prin construcție, A_k are rang k, și avem

$$\|A-A_k\|_2 = \left\| \sum_{i=k+1}^{p} \sigma_i u_i v_i^\top \right\|_2 = \left\| U \operatorname{diag}(0,\ldots,0,\sigma_{k+1},\ldots,\sigma_p)\, V^\top \right\|_2 = \sigma_{k+1}.$$

Rămâne de arătat că

$$\|A - B\|_2 \geq \sigma_{k+1}$$

pentru orice matrice B de rang k. Fie B o astfel de matrice. Atunci nucleul său are dimensiunea $n-k$. Subspațiul V_{k+1} generat de $(v_1,\ldots,v_{k+1})$ are dimensiunea $k+1$, iar deoarece suma dimensiunilor nucleului lui B și a lui V_{k+1} este

$$(n-k)+(k+1) = n+1,$$

cele două subspații trebuie să se intersecteze într-un subspațiu de dimensiune cel puțin 1. □

Proof.
Alegem un vector unitar $h \in \ker(B) \cap V_{k+1}$. Atunci $Bh = 0$, iar deoarece V și U sunt izometrii, obținem

$$\begin{aligned}\|A - B\|_2^2 &\geq \|(A - B)h\|_2^2 = \|Ah\|_2^2 \\ &= \|U\Sigma V^\top h\|_2^2 = \|\Sigma V^\top h\|_2^2 \geq \sigma_{k+1}^2 \|V^\top h\|_2^2 = \sigma_{k+1}^2,\end{aligned}$$

ceea ce demonstrează afirmația.

□

Observăm că A_k poate fi stocată folosind $(m + n)k$ elemente, în loc de mn elemente. Când $k \ll m, n$, acest lucru reprezintă un câștig substanțial.

Figure 1: Considerăm această fotografie și dorim să o comprimăm.

Salvarea acestei fotografii în Matlab se face sub forma unei matrice.

A comprima imaginea revine la a folosi o matrice de dimensiuni mai mici.

```
A = im2double(imread('cameraman.tif'));   % exemplu grayscale
[U,S,V] = svd(A,'econ');
size(A)
figure(1)
imshow(A), title(sprintf('Initial'));
k = 20;
Ak = U(:,1:k) * S(1:k,1:k) * V(:,1:k)';
figure(2)
imshow(Ak), title(sprintf('SVD rank-%d', k));
```

```
k = 40;
Ak = U(:,1:k) * S(1:k,1:k) * V(:,1:k)';
figure(3)
imshow(Ak), title(sprintf('SVD rank-%d', k));
k = 100;
Ak = U(:,1:k) * S(1:k,1:k) * V(:,1:k)';
figure(4)
imshow(Ak), title(sprintf('SVD rank-%d', k));
k = 200;
Ak = U(:,1:k) * S(1:k,1:k) * V(:,1:k)';
figure(5)
imshow(Ak), title(sprintf('SVD rank-%d', k));
s = svd(A);
cs = cumsum(s) / sum(s);
figure
plot(cs, "LineWidth", 2)
xlabel("k")
ylabel("Suma cumulativa normalizata")
```

Figure 2: Fotografie inițială.

Figure 3: Fotografie reconstruită

Figure 4: Fotografie inițială.

Figure 5: Fotografie reconstruită

Figure 6: Fotografie inițială.

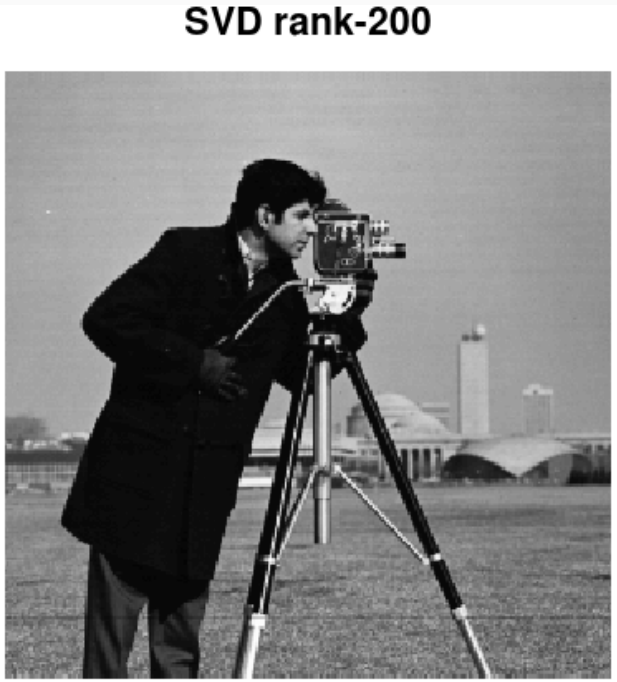


Figure 7: Fotografie reconstruită

Figure 8: Fotografie inițială.

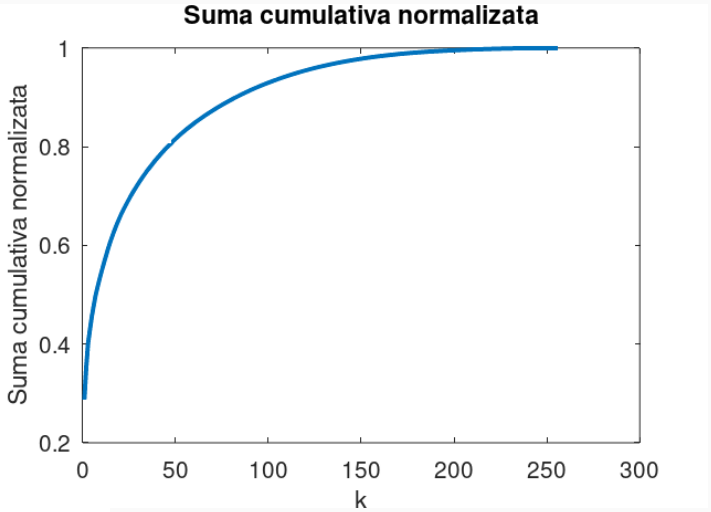


Figure 9: Se observă că în jur de valoarea 200 valoarea stagnează, ceea ce însemnă că nu mai am informații la care să nu se poată renunța.

Fotografii color

O imagine color nu este o matrice, ci un tensor de ordin 3:

$\mathcal{I} \in \mathbb{R}^{m \times n \times 3}$

- dimensiunea 1=m: înălțime
- dimensiunea 2=n: lățime
- dimensiunea 3=3: canal de culoare (R, G, B) (roșu, verde, albastru)

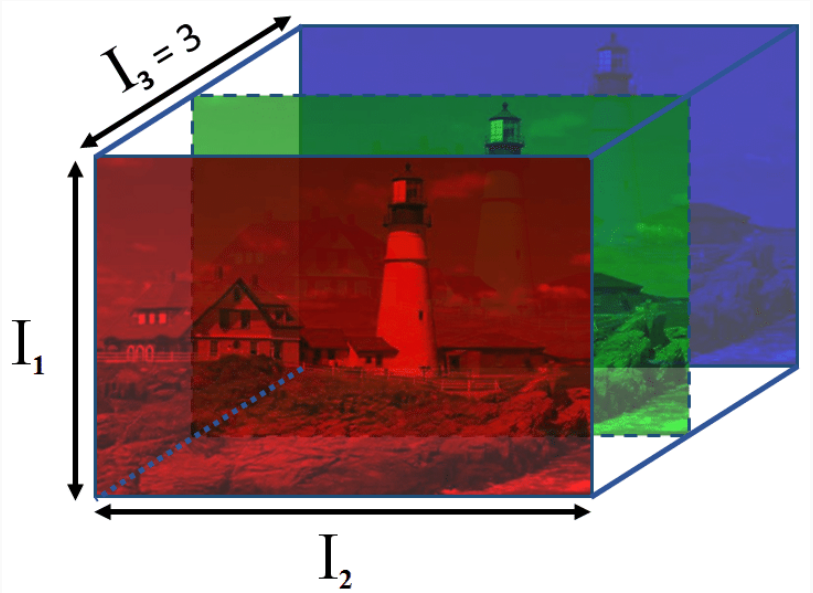


Figure 10: Fiecare "felie" este o imagine grayscale.

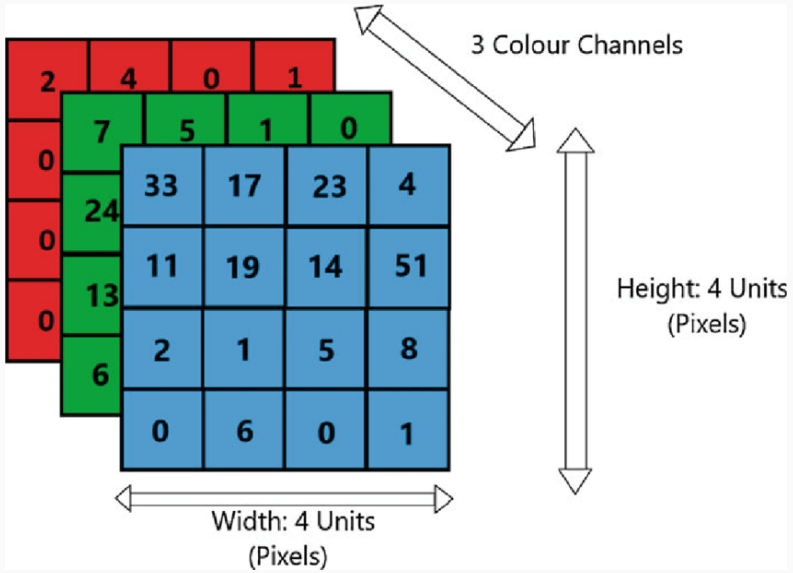


Figure 11: Avem "un vector de matrice", adică un tensor de ordin 3.

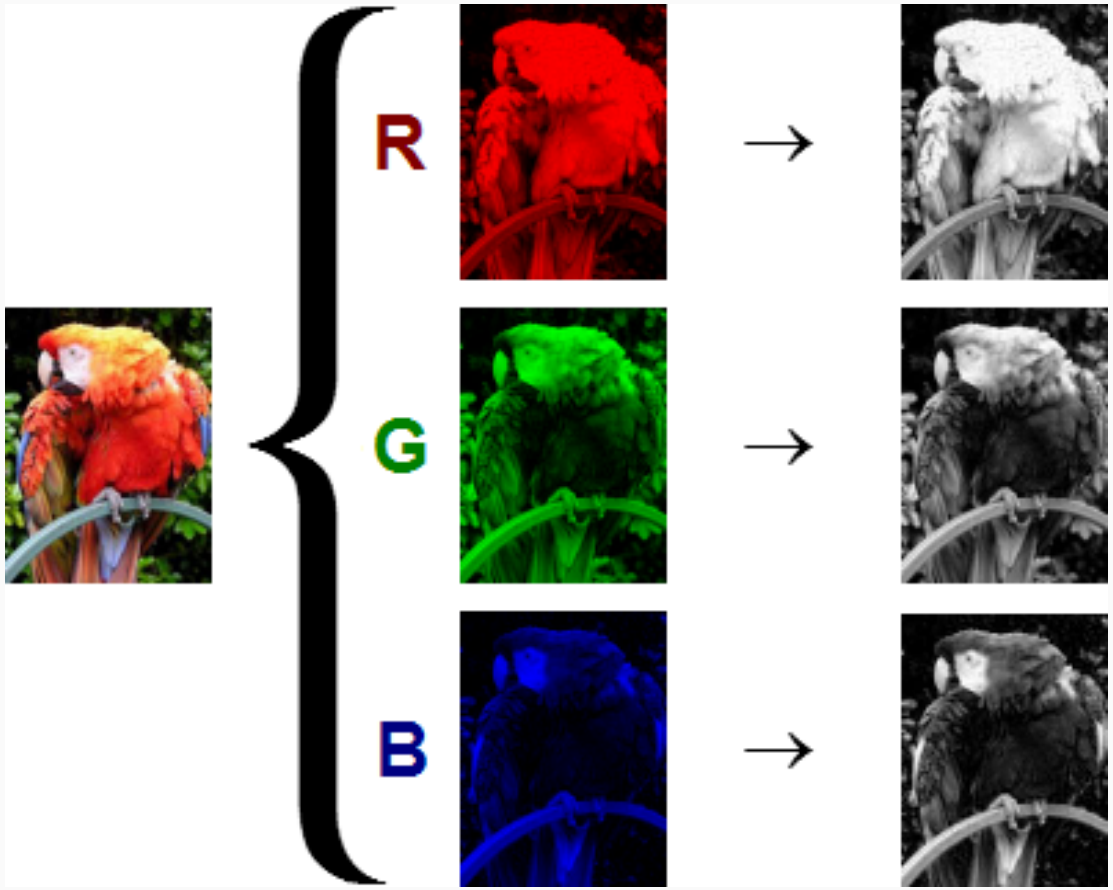


Figure 12: Fiecare “felie" este o imagine grayscale.

Putem să aplicăm SVD pentru fiecare canal

```
I = im2double(imread("peppers.png"));   % imagine RGB
size(I)   % m x n x 3
figure(1)
imshow(I)
k = 100;   % rangul de aproximare
Ic = zeros(size(I));
for c = 1:3
A = I(:,:,c);
[U,S,V] = svd(A, "econ");
Ic(:,:,c) = U(:,1:k) * S(1:k,1:k) * V(:,1:k)';
end
figure(2)
imshow(Ic); title(sprintf("SVD RGB -- rank %d", k))
for c = 1:3
s = svd(I(:,:,c)); cs = cumsum(s.^2) / sum(s.^2);
figure(c+2); plot(cs, "LineWidth", 2); xlabel("k");
ylabel("Energie cumulativa"); title(sprintf("Canal %d (R=1, G=2, B=3)
end
```

Figure 13: Fotografie inițială.

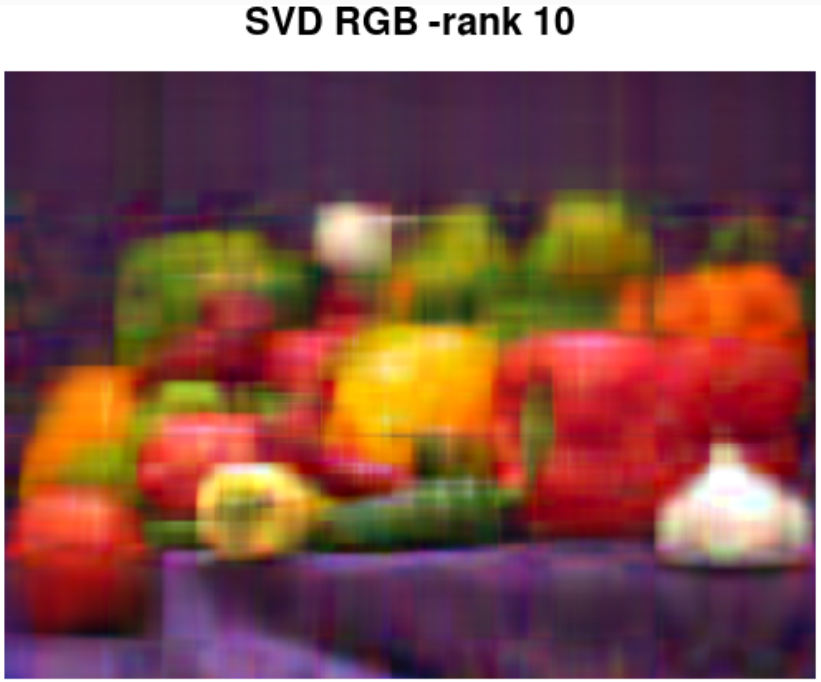


Figure 14: Fotografie reconstruită

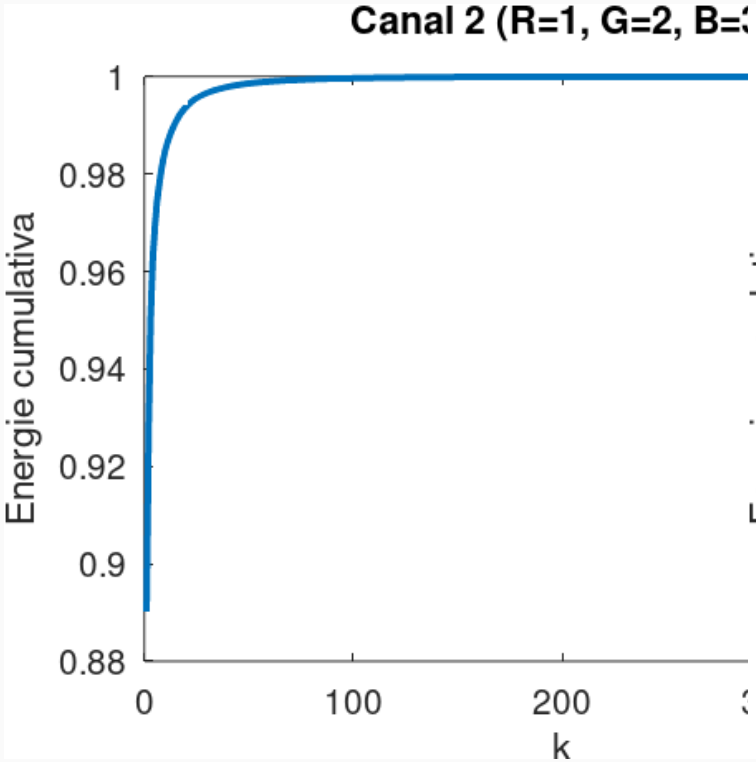


Figure 15: Suma cumulativă canal 1.

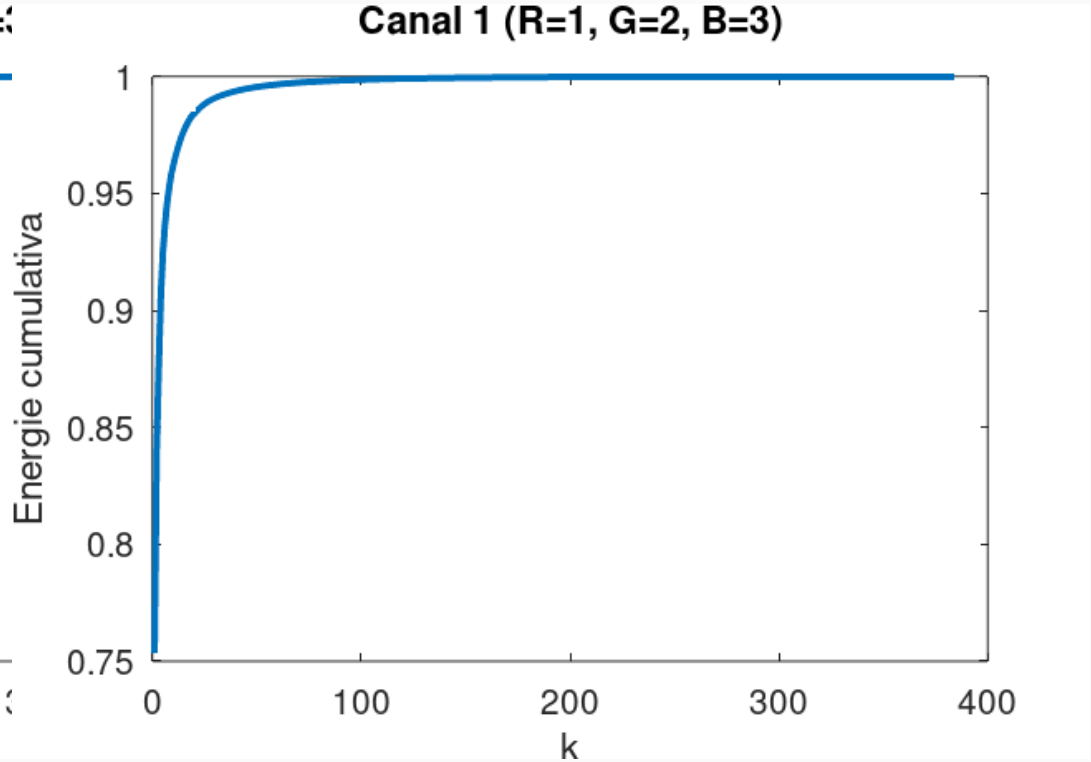


Figure 16: Suma cumulativă canal 2.

Figure 17: Fotografie inițială.

Figure 18: Fotografie reconstruită

O abordare care să nu separe canalele

O altă strategie este să se lipească matricele tensorului (se mai numesc canale) si să se facă SVD și compresie pentru matricea mare, încât reducerea să se facă corelat pe cele 3 canale, o aproximare care să țină cont de întregul tensor.

```
I = im2double(imread("peppers.png"));
[m,n,~] = size(I);
A = [I(:,:,1), I(:,:,2), I(:,:,3)];      % m x (3n)
[U,S,V] = svd(A,"econ");
            figure(1);  imshow(I); title(sprintf("Initial"))
k = 60;
Ak = U(:,1:k)*S(1:k,1:k)*V(:,1:k)';
R = Ak(:, 1:n);
G = Ak(:, n+1:2*n);
B = Ak(:, 2*n+1:3*n);
Ic = cat(3, R, G, B);
imshow(Ic)
   figure(2);  imshow(Ic); title(sprintf("SVD RGB -rank %d", k
 s = svd(A);
 cs = cumsum(s.^2) / sum(s.^2);
 figure(3); clf;  plot(cs, "LineWidth", 2); xlabel("k")
 ylabel("Energie cumulativa");
title(sprintf("Suma cumulativ-canale reunite"))
```

YCbCr în loc de RGB

YCbCr este folosit în loc de RGB pentru că permite o comprimare mult mai eficientă perceptual, adică se pierde mai puțin din informația percepută de ochiul uman, chiar dacă, din punct de vedere numeric, pierderile sunt mai mari.

Limitările spațiului de culoare RGB

În spațiul de culoare RGB:

- canalele R, G, B sunt puternic corelate;
- fiecare canal amestecă informația de:
  - luminanță (luminozitate);
  - culoare.

Dacă se aplică metode de comprimare direct în RGB:

- se pierd simultan luminanța și culoarea;
- degradarea imaginii devine rapid vizibilă.

Componentele spațiului de culoare YCbCr

Y, Cb și Cr sunt cele trei componente ale spațiului de culoare YCbCr, fiecare având un rol distinct în reprezentarea și comprimarea imaginilor.

Componenta Y – Luminanță (brightness). Componenta Y reprezintă luminozitatea imaginii:

- indică cât de deschis sau închis este un pixel;
- conține structura, contururile și detaliile imaginii;
- este componenta la care ochiul uman este cel mai sensibil.

Dacă se puastrează doar componenta Y, se obține o imagine grayscale foarte apropiată de original.

Componentele spațiului de culoare YCbCr

Componenta Cb – Crominanță albastră (blue-difference).

Componenta Cb muasoară diferența dintre componenta albastră și luminanțua:

$$Cb = B - Y \quad \text{(scalat)}$$

Aceasta:

- indică cât de albastru este pixelul față de luminozitatea sa;
- conține informație de culoare, nu de structură;
- este mai puțin importantă din punct de vedere perceptual.

Componentele spațiului de culoare YCbCr

Componenta Cr – Crominanță roșie (red-difference).

Componenta Cr muasoară diferența dintre componenta roșie și luminanță:

$$Cr = R - Y \quad \text{(scalat)}$$

Aceasta:

- indică cât de roșu este pixelul față de luminozitatea sa;
- conține informație de culoare;
- este, la fel ca Cb, mai puțin importantă perceptual.

Relația dintre RGB și YCbCr

Componenta Y poate fi aproximată ca o combinație ponderată a canalelor RGB, iar Cb și Cr reprezintă deviații de culoare față de Y. Conversia standard (simplificată) este:

$$Y = 0.299R + 0.587G + 0.114B,$$
$$Cb = B - Y,$$
$$Cr = R - Y.$$

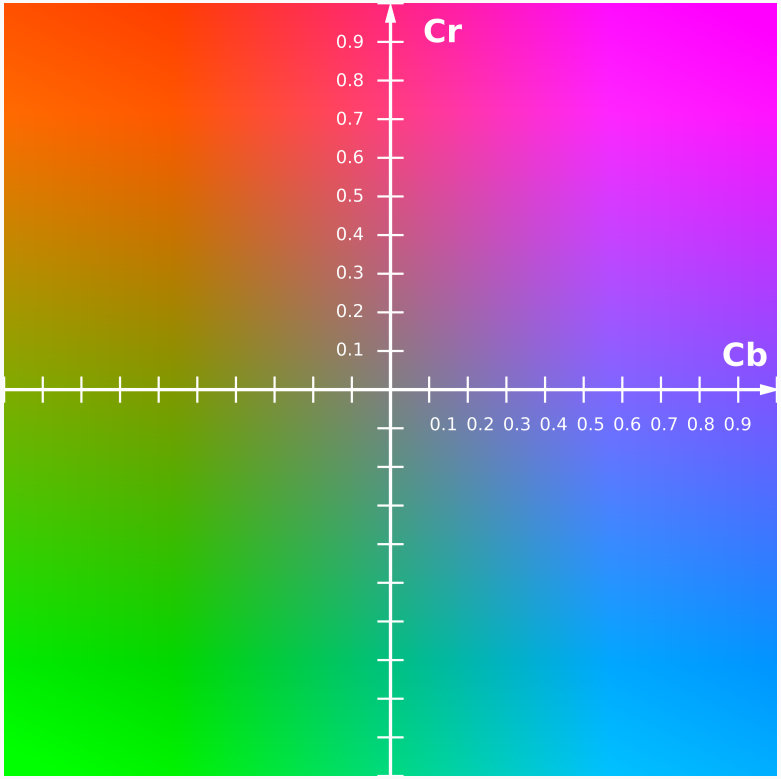


Figure 19: Valori pentru Cb și Cr.

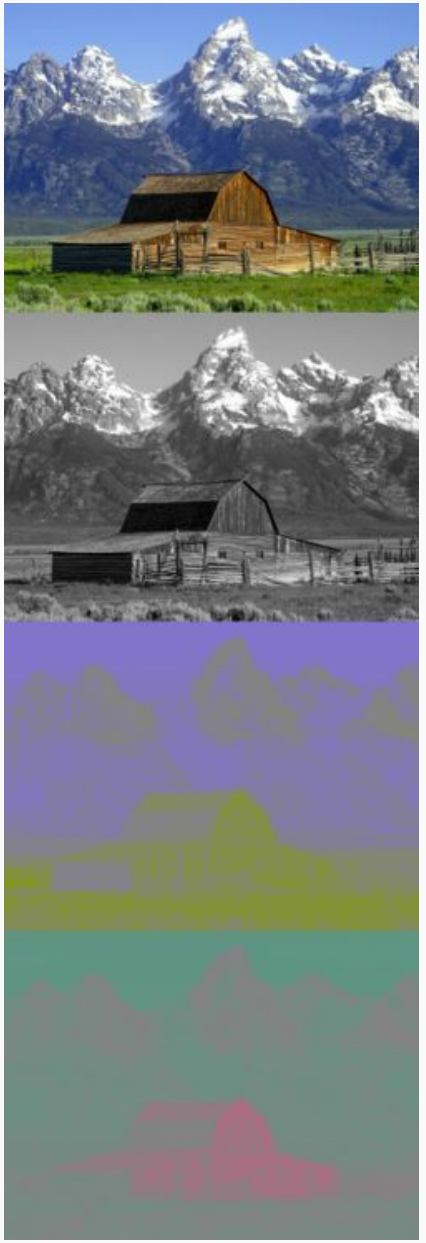


Figure 20: Ce extrage Y, Cb, Cr?

?Implementare cu YCbCr

```
I = im2double(imread("peppers.png"));          % m x n x 3
YCC = rgb2ycbcr(I);
kY = 80;   kC = 25;
J = zeros(size(YCC));
for c = 1:3
A = YCC(:,:,c);
[U,S,V] = svd(A,"econ");
k = (c==1)*kY + (c~=1)*kC;
J(:,:,c) = U(:,1:k)*S(1:k,1:k)*V(:,1:k)';
end
Ic = ycbcr2rgb(J);
figure(1)
imshow(Ic)
```

Alte abordări

- În loc să se aplice SVD pentru toată imaginea, se aplică pe blocuri, încât ce se pierde să fie local.
- Se generalizează descompunerea SVD pentru tensori de ordin superior.
- Se poate lucra cu optimizare pe spații de tensori.

Morala cursului?

Morala 1: E bine să ai în buzunar cunoștințe de algebră numerică și calcul științific

Figure 21:

Morala 2: Slide-ul conține o imagine generată de AI dar are și unele greșeli. PUTEȚI SĂ LE IDENTIFICAȚI?

Figure 22:

Soluții pot fi date de AI, dar noi trebuie să avem capacitatea de a le analiza critic!

Succes în călătoria voastră!