Calcul științific

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

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) (50% din nota finală):
  - un referat la seminar în care să se sumarizeze noțiunile teoretice și să se exemplifice aplicarea lor (25% din nota finală),
  - activitatea la seminar (25% din nota finală);
- Examen final mixt (50% din nota finală)
  - rezolvarea a două exerciții/scriere de programe din care unul foarte asemănător cu cele din fișele de lucru pentru laboratoare sau din referatele prezentate (25% din nota finală),
  - explicarea noțiunilor teoretice prin extragerea a două bilete (25% din nota finală).

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 $v \in \mathbb{C}^n$ 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

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

Dar cum știm dacă vom întâlni pivoți nuli fă ră 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 macar 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 în secțiunea următoare. 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

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

Dacă ar fi posibil să avem

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

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:

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

unde

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

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 $p = 1, q = 1$ este 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

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

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.

Sisteme nedeterminate

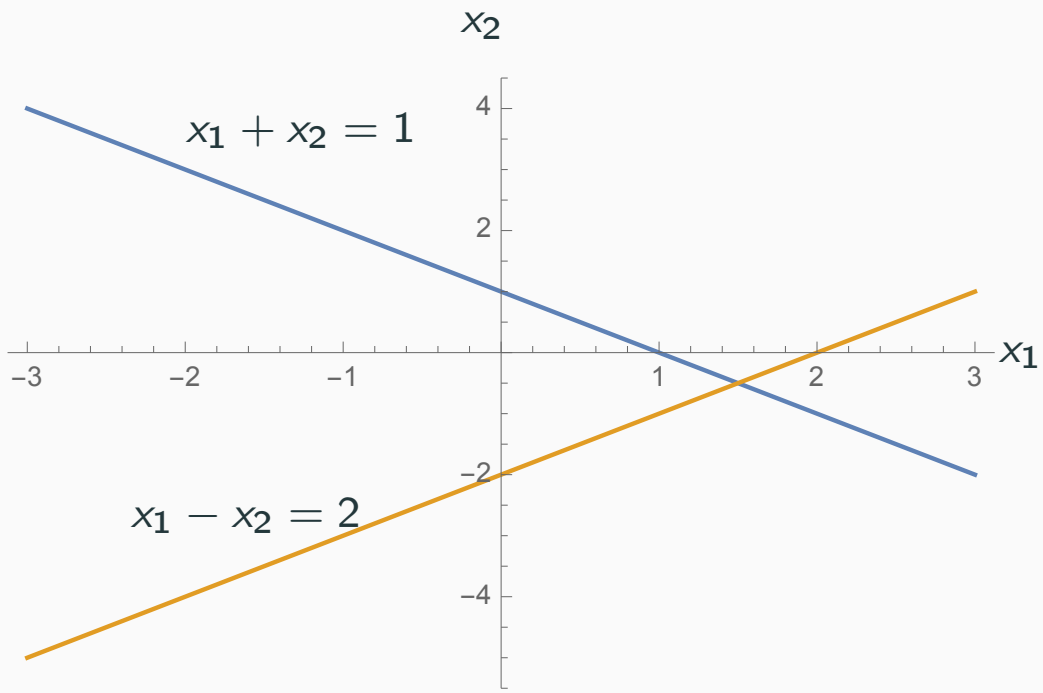
“Soluția” sistemelor supradeterminate

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 $x_1 = \frac{3}{2}$.

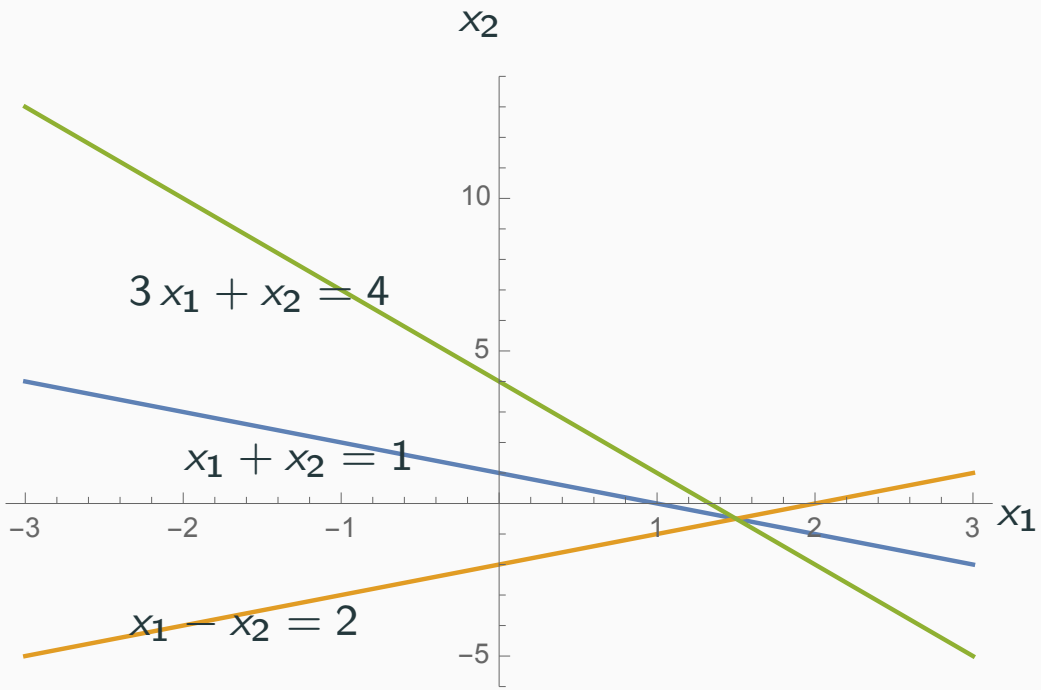
Prin urmare, punctul de intersecție al celor două drepte este punctul $(x_1, x_2) = (\frac{3}{2}, -\frac{1}{2})$.

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.

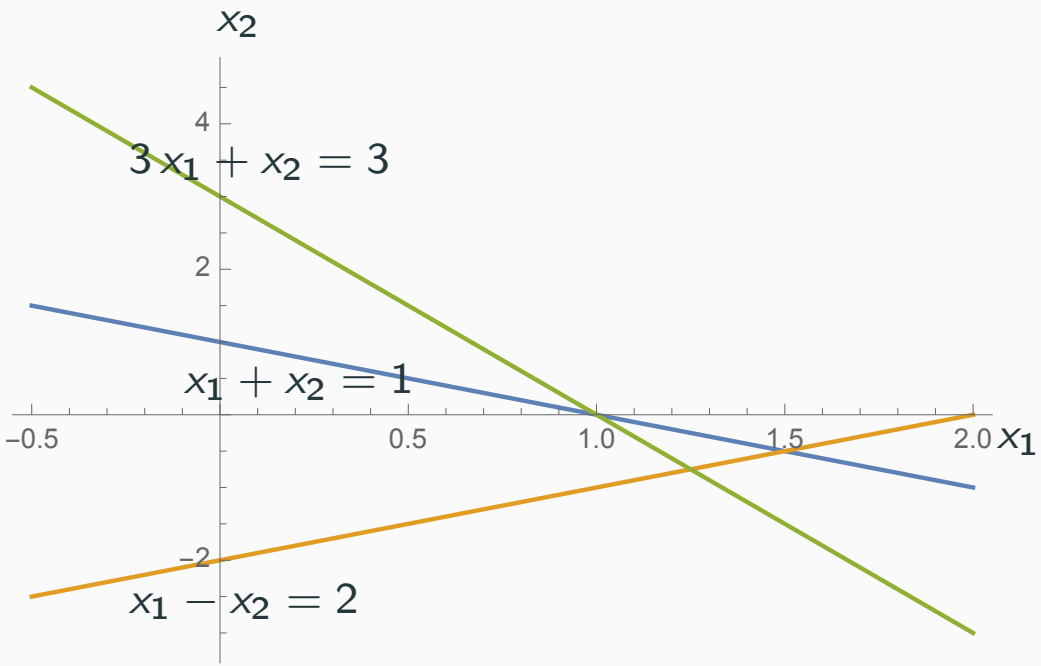


Sisteme supradeterminate

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

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

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

$$f(x^*) = \|Ax^* - b\|_2^2 \leq \min_{x \in \mathbb{R}^n} \underbrace{\|Ax - b\|_2^2}_{:= f(x)} = \min_{x \in \mathbb{R}^n} f(x). \tag{107}$$

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

$$f(x) = (Ax - b)^T (Ax - b) = x^T A^T Ax - 2x^T A^T b + b^T b, \qquad (108)$$

aflăm că

$$\nabla f(x^*) = 2A^T Ax^* - 2A^T b = 0, \qquad (109)$$

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

$$A^T Ax^* = A^T b \qquad (110)$$

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

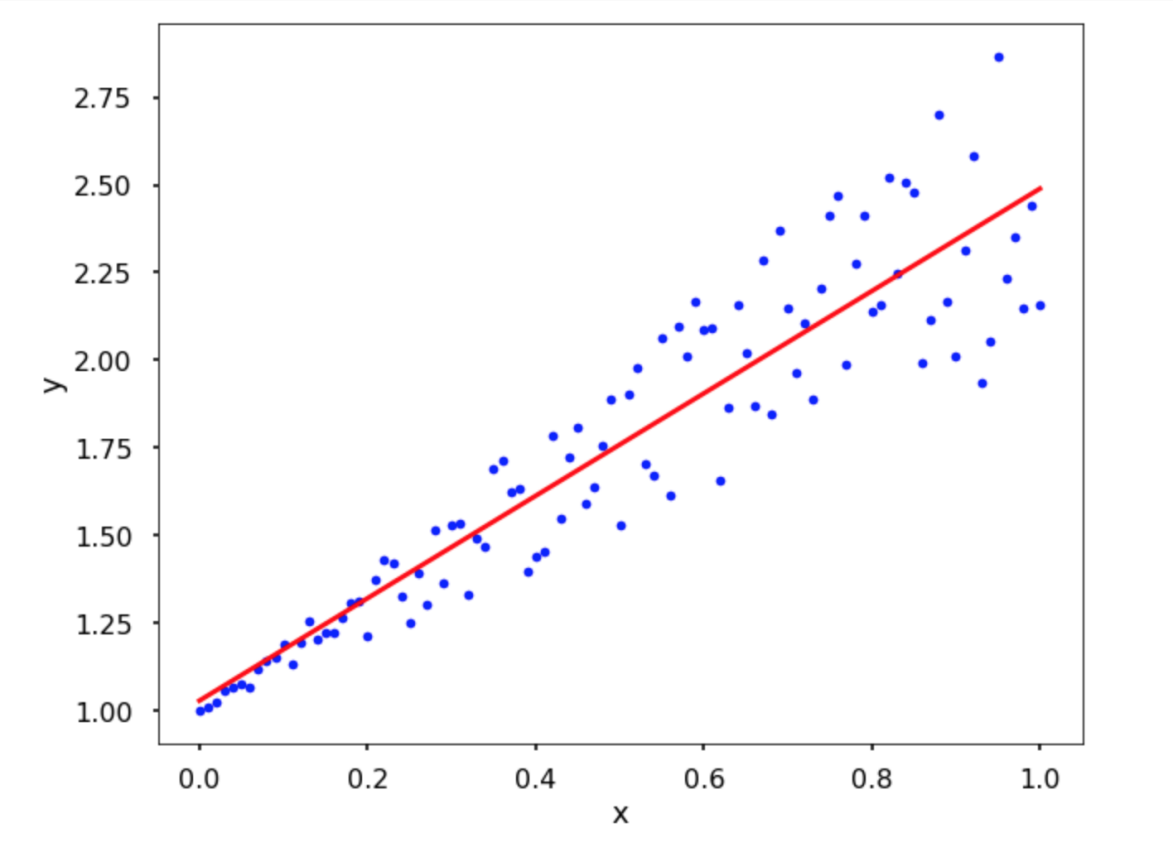
$$A = \begin{pmatrix} 1 & 1 \\ 2^{-27} & 0 \\ 0 & 2^{-27} \end{pmatrix}, \qquad fl(A^T A) = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}. \tag{111}$$

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

Data Fitting

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

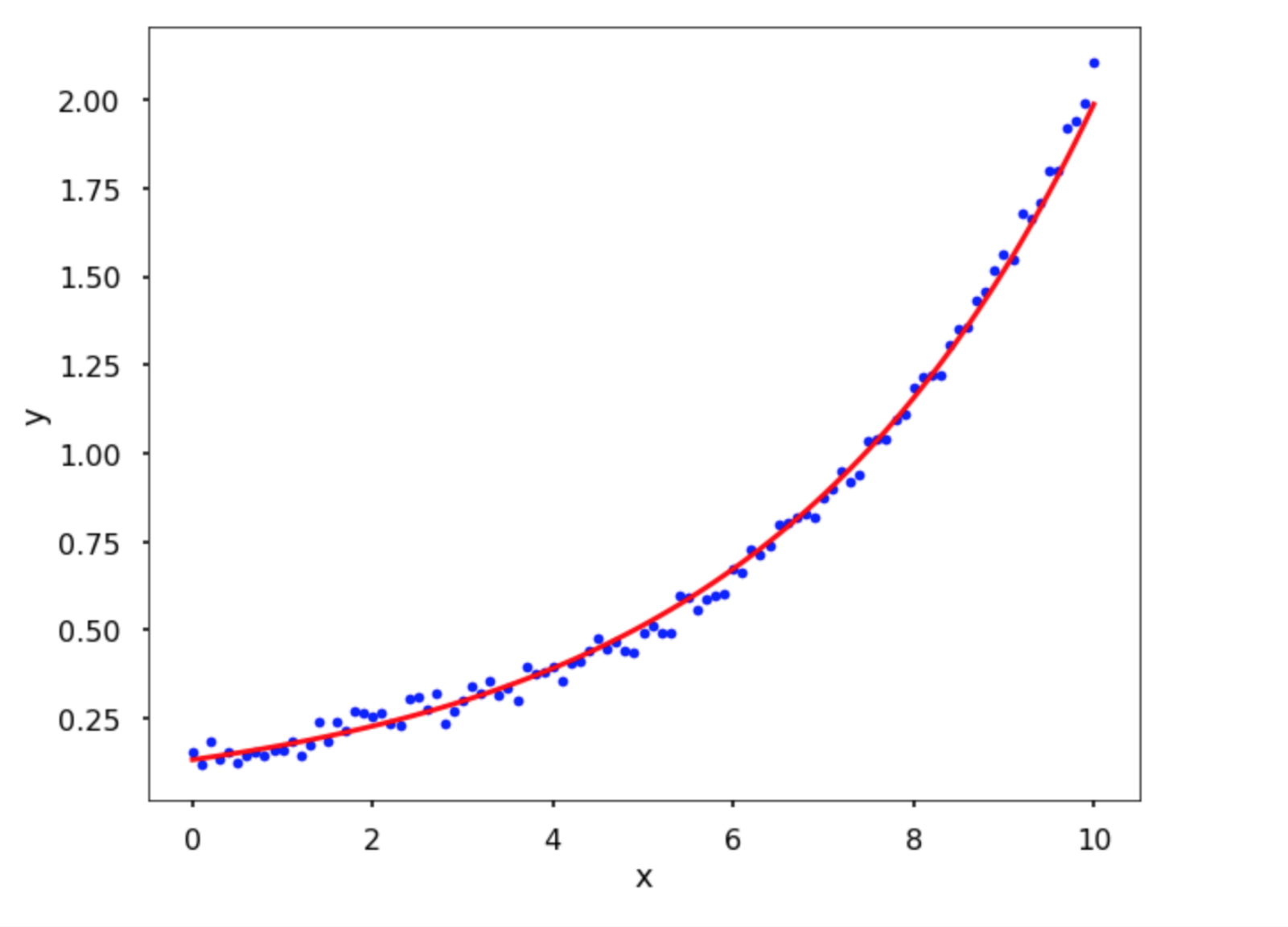
$$\min_{x \in \mathbb{R}^n} \sum_{i=1}^{m} (\langle s_i, x \rangle - t_i)^2.$$

Aceasta este o problemă (LS) scrisă ca

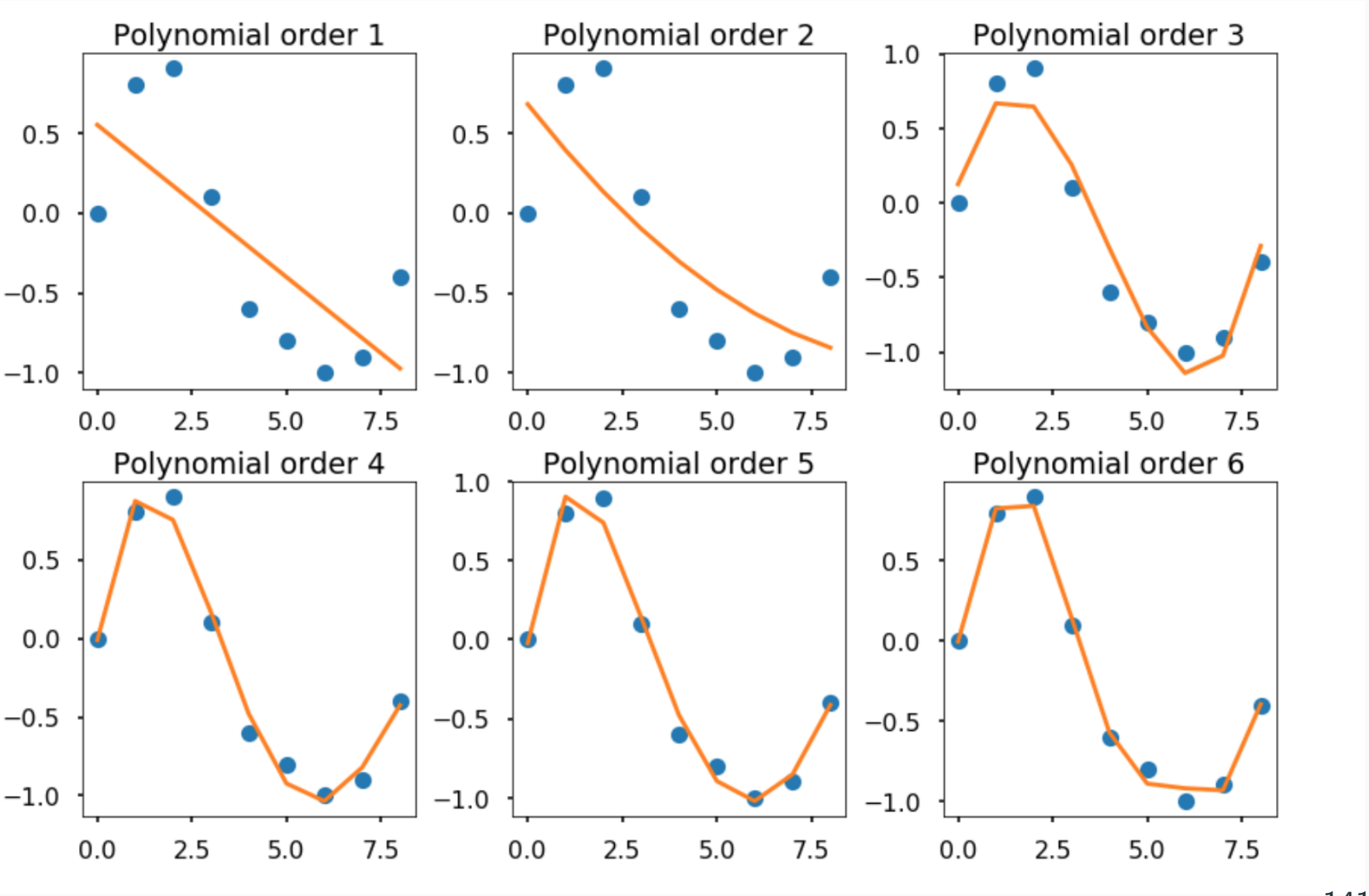
$$\min_{x \in \mathbb{R}^n} \|S\,x - t\|^2,$$

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

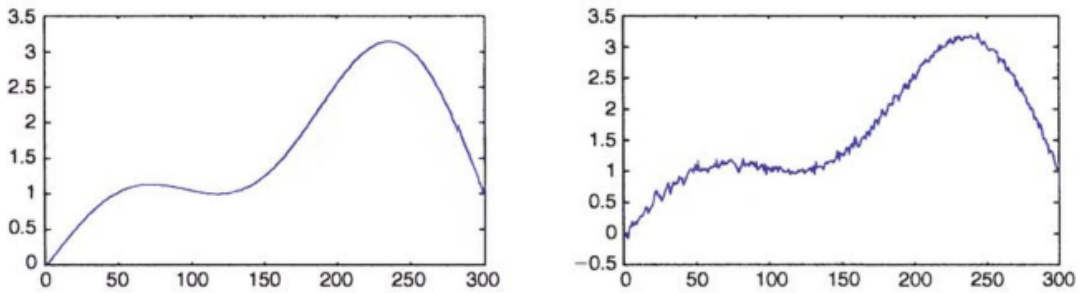


Figure 3.2. *A signal (left image) and its noisy version (right image).*

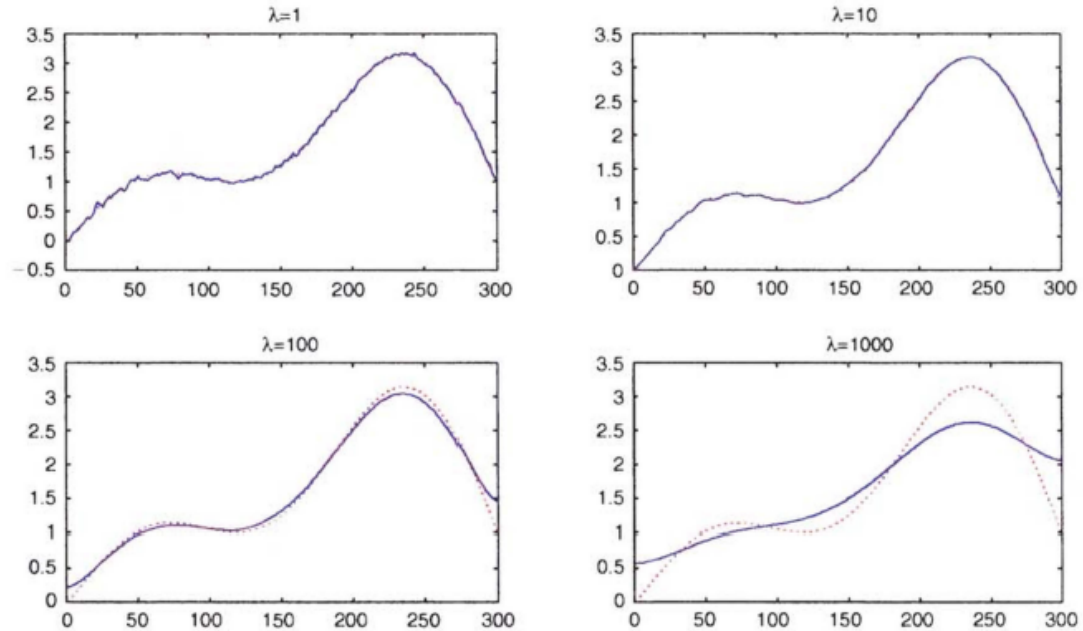


Figure 3.3. *Four reconstructions of a noisy signal by RLS solutions.*

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

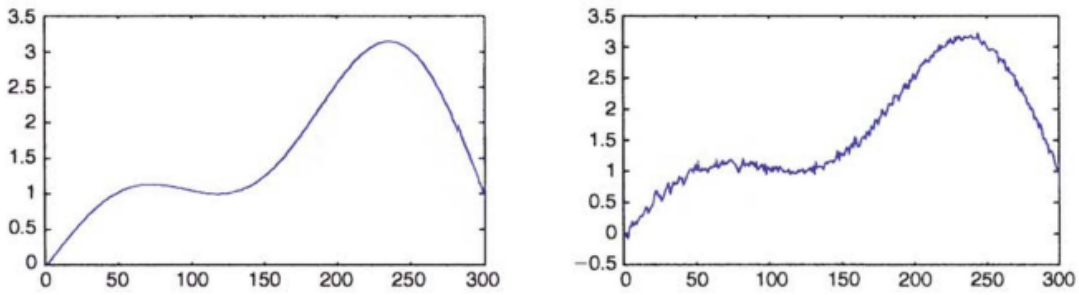


Figure 3.2. *A signal (left image) and its noisy version (right image).*

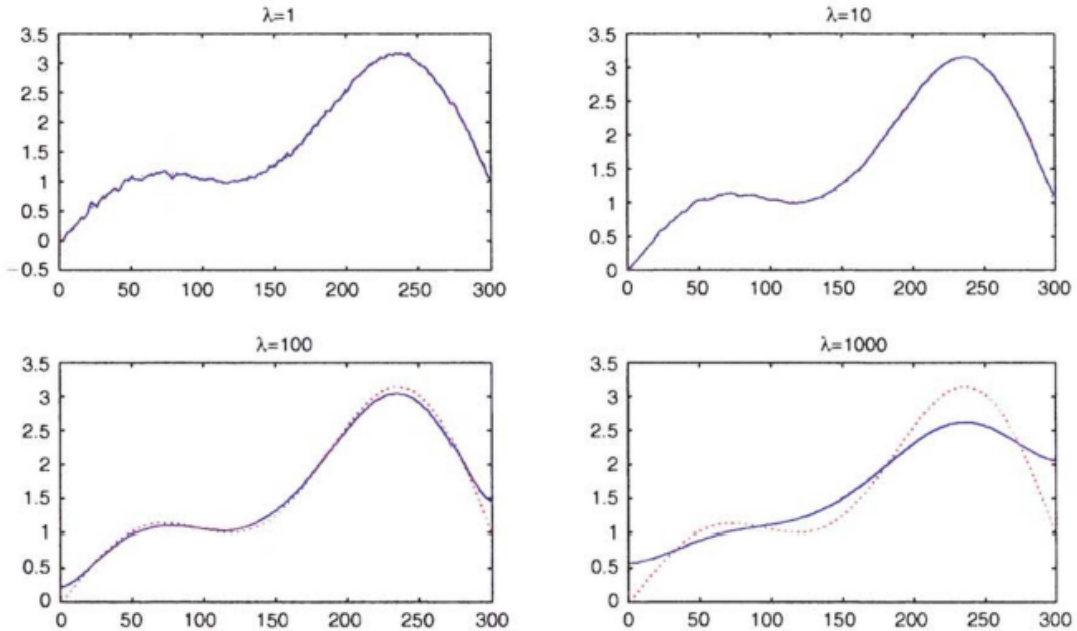


Figure 3.3. *Four reconstructions of a noisy signal by RLS solutions.*

Problema neliniară a celor mai mici pătrate: Circle Fitting

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:

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

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, $A^T A = \widetilde{R}^T \widetilde{R}$.

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. Pornind de la un set de vectori liniar independenți, $x_1, \cdots, x_n$, acest algoritm generează un nou set de vectori mutual ortogonali, $q_1, \cdots, q_n$, dați de

$$q_1 = x_1,$$

$$q_{k+1} = x_{k+1} - \sum_{i=1}^{k} \frac{\langle q_i, x_{k+1} \rangle}{\langle q_i, q_i \rangle} q_i, \quad k = 1, \cdots, n-1. \tag{112}$$

Notând cu $a_1, \cdots, a_n$ vectorii coloană ai lui A, setăm

$$\widetilde{q}_1 = \frac{a_1}{\|a_1\|} \tag{113}$$

și, pentru $k = 1, \cdots, n-1$, calculăm vectorii coloană ai lui $\widetilde{Q}$ ca

$$\widetilde{q}_{k+1} = \frac{q_{k+1}}{\|q_{k+1}\|}, \text{unde } q_{k+1} = a_{k+1} - \sum_{i=1}^{k} \frac{\langle \widetilde{q}_i, a_{k+1} \rangle}{\langle \widetilde{q}_i, \widetilde{q}_i \rangle} \widetilde{q}_i. \tag{114}$$

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

$$H^TH = A^TA = \widetilde{R}^T\widetilde{Q}^T\widetilde{Q}\widetilde{R} = \widetilde{R}^T\widetilde{R}, \tag{115}$$

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 are o utilitate practică redusă, deoarece vectorii generați își pierd independența liniară din cauza erorilor de rotunjire. Într-adevăr, în aritmetica cu virgulă mobilă, algoritmul produce valori foarte mici pentru $\|q_{k+1}\|_2$ și $\widetilde{r}_{kk}$, ceea ce duce la instabilitate numerică și pierderea ortogonalității pentru matricea $\widetilde{Q}$.

Aceste neajunsuri sugerează utilizarea unei versiuni mai stabile, cunoscută sub numele de metoda Gram-Schmidt modificată.

La începutul pasului $k+1$, proiecțiile vectorului a_{k+1} pe direcția vectorilor $\widetilde{q}_1, \cdots, \widetilde{q}_{k+1}$ sunt progresiv scăzute din a_{k+1}. Pe vectorul rezultat, se realizează apoi pasul de ortogonalizare.

În practică, după ce se calculează $\langle \widetilde{q}_1, a_{k+1} \rangle \widetilde{q}_1$ la pasul $k+1$, acest vector este imediat scăzut din a_{k+1}. De exemplu, se definește

$$a_{k+1}^{(1)} = a_{k+1} - \langle \widetilde{q}_1, a_{k+1} \rangle \widetilde{q}_1. \tag{116}$$

Acest nou vector $a_{k+1}^{(1)}$ este proiectat pe direcția lui $\widetilde{q}_2$, iar proiecția obținută este scăzută din $a_{k+1}^{(1)}$, obținându-se

$$a_{k+1}^{(2)} = a_{k+1}^{(1)} - \langle \widetilde{q}_2, a_{k+1}^{(1)} \rangle \widetilde{q}_2 \tag{117}$$

și așa mai departe, până când $a_{k+1}^{(k)}$ este calculat.

Se poate verifica faptul că $a_{k+1}^{(k)}$ coincide cu vectorul corespunzător q_{k+1} din procesul Gram-Schmidt standard, deoarece, datorită ortogonalității lui $\widetilde{q}_1, \cdots, \widetilde{q}_k$

$$\begin{aligned} a_{k+1}^{(k)} &= a_{k+1} - \langle \widetilde{q}_1, a_{k+1} \rangle \widetilde{q}_1 - \langle \widetilde{q}_2, a_{k+1} - \langle \widetilde{q}_1, a_{k+1} \rangle \widetilde{q}_1 \rangle \widetilde{q}_2 + \cdots \\ &= a_{k+1} - \sum_{i=1}^{k} \langle \widetilde{q}_i, a_{k+1} \rangle \widetilde{q}_i. \end{aligned} \tag{118}$$

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

$$x^* = \widetilde{R}^{-1}\widetilde{Q}^T b, \tag{119}$$

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

$$\Phi(x^*) = \sum_{i=n+1}^{m} [(Q^T b)_i]^2. \tag{120}$$

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

$$\|Ax - b\|_2^2 = \|Rx - Q^T b\|_2^2. \tag{121}$$

Aducându-ne aminte că R este trapezoidală superior, avem

$$\|Rx - Q^T b\|_2^2 = \|\widetilde{R}x - \widetilde{Q}^T b\|_2^2 + \sum_{i=n+1}^{m} [(Q^T b)_i]^2, \tag{122}$$

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

Dacă A nu are rang complet, tehnicile de soluționare 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:

$$\begin{aligned} &\text{găsiți } \; x^* \in \mathbb{R}^n \; \text{ cu norma euclidiană minimă astfel încât} \\ &\|A\,x^* - b\|_2^2 \leq \min_{x \in \mathbb{R}^n} \underbrace{\|Ax - b\|_2^2}_{:=\Phi(x)}. \end{aligned} \tag{123}$$

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

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

see 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

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

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

$$a_2 = \begin{pmatrix} 0 \\ A^{(1)}(2:m, 2) \end{pmatrix}, \tag{124}$$

apoi definim

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

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

$$a_k = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ A^{(k-1)}(k:m,k) \end{pmatrix}, \tag{125}$$

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 $\operatorname{rang} A = n$.

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 $A \in \mathbb{R}^{n \times m}$. Atunci există matrice ortogonale $U \in \mathbb{R}^{n \times n}$ și $V \in \mathbb{R}^{m \times m}$ și o matrice diagonală $\Sigma = diag(\sigma_1, \ldots, \sigma_n) \in \mathbb{R}^{n \times m}$ 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

Norma 2 a lui A este definită de $\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2$. Astfel, există un vector x cu $\|x\|_2 = 1$ astfel încât

$$z = Ax, \quad \|z\|_2 = \|A\|_2 =: \sigma.$$

Notăm $y := \frac{z}{\|z\|_2}$. Acest lucru duce la $Ax = \sigma y$ cu $\|x\|_2 = \|y\|_2 = 1$.

Apoi, extindem x în o bază ortonormată a lui $\mathbb{R}^n$. Dacă $V \in \mathbb{R}^{n \times n}$ este matricea care conține vectorii bazei drept coloane, atunci V este o matrice ortogonală care poate fi scrisă ca $V = [x, V_1]$, unde $V_1^\top x = 0$. Similar, putem construi o matrice ortogonală $U \in \mathbb{R}^{n \times m}$ care să satisfacă $U = [y, U_1]$, $U_1^\top y = 0$.

Demonstrație

$$A_1 = U^\top AV = \begin{bmatrix} y^\top \\ U_1^\top \end{bmatrix} A[x, V_1] = \begin{bmatrix} y^\top Ax & y^\top AV_1 \\ U_1^\top Ax & U_1^\top AV_1 \end{bmatrix} = \begin{bmatrix} \sigma & \omega^\top \\ 0 & B \end{bmatrix},$$

pentru că $y^\top Ax = y^\top \sigma y = \sigma y^\top y = \sigma$ și $U_1^\top Ax = \sigma U_1^\top y = 0$ deoarece $U_1 \perp y$.

Afirmația noastră este că $\omega^\top := y^\top AV_1 = 0$. Pentru a demonstra acest lucru, calculăm

$$A_1 \begin{pmatrix} \sigma \\ \omega \end{pmatrix} = \begin{pmatrix} \sigma^2 + \|\omega\|_1^2 \\ B\omega \end{pmatrix}$$

și concluzionăm din această ecuație că

$$\left\| A_1 \begin{pmatrix} \sigma \\ \omega \end{pmatrix} \right\|_2^2 = (\sigma^2 + \|\omega\|_2^2)^2 + \|B\omega\|_2^2 \geq (\sigma^2 + \|\omega\|_2^2)^2.$$

Demonstrație

Acum, deoarece V și U sunt ortogonale, $\|A_1\|_2 = \|U^\top AV\|_2 = \|A\|_2 = \sigma$ deducem

$$\sigma^2 = \|A_1\|_2^2 = \max_{\|x\|_2 \neq 0} \frac{\|A_1 x\|_2^2}{\|x\|_2^2} \geq \frac{\left\| A_1 \begin{pmatrix} \sigma \\ \omega \end{pmatrix} \right\|_2^2}{\left\| \begin{pmatrix} \sigma \\ \omega \end{pmatrix} \right\|_2^2} \geq \frac{(\sigma^2 + \|\omega\|_2^2)^2}{\sigma^2 + \|\omega\|_2^2}.$$

Ultima ecuație se scrie

$$\sigma^2 \geq \sigma^2 + \|\omega\|_2^2,$$

și concluzionăm că $\omega = 0$. Astfel, am obținut

$$A_1 = U^\top AV = \begin{bmatrix} \sigma & 0 \\ 0 & B \end{bmatrix}.$$

Demonstrație

Putem acum aplica aceeași construcție la sub-matricea B și astfel, in final, să ajungem la o matrice diagonală.
Dacă scriem ecuația $A = U\Sigma V^T$ în formă partitionată, în care Σ_r conține doar valorile singulare nenule, obținem

$$A = [U_1, U_2] \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} [V_1, V_2]^T = U_1 \Sigma_r V_1^T = \sum_{i=1}^{r} \sigma_i u_i v_i^T . \qquad (126)$$

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".

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 $A\,A^T$.

În ceea ce privește rangul, dacă

$$\sigma_1 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_p = 0. \tag{127}$$

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

$$\Sigma^\dagger = \operatorname{diag}\left(\frac{1}{\sigma_1}, \cdots, \frac{1}{\sigma_r}, 0, \cdots, 0\right). \tag{128}$$

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

$$\begin{aligned} &\textit{găsește } \; x^* \in \mathbb{R}^n \; \textit{ cu norma Euclidiană minimă astfel că} \\ &\|A\,x^* - b\|_2^2 \leq \min_{x \in \mathbb{R}^n} \underbrace{\|Ax - b\|_2^2}_{:=\Phi(x)} \end{aligned} \tag{129}$$

este

$$x^* = A^\dagger b, \tag{130}$$

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

$$\|\Sigma w - U^T b\|_2^2 \leq \|\Sigma y - U^T b\|_2^2 \qquad \forall\ y \in \mathbb{R}^n. \tag{131}$$

Dacă r este numărul de valori singulare nenule σ_i ale lui A, atunci

$$\|\Sigma w - U^T b\|_2^2 = \sum_{i=1}^{r} \left(\sigma_i w_i - (U^T b)_i\right)^2 + \sum_{i=r+1}^{m} \left((U^T b)_i\right)^2, \tag{132}$$

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.

Metoda gradientului folosită pentru rezolvarea sistemelor

Scop

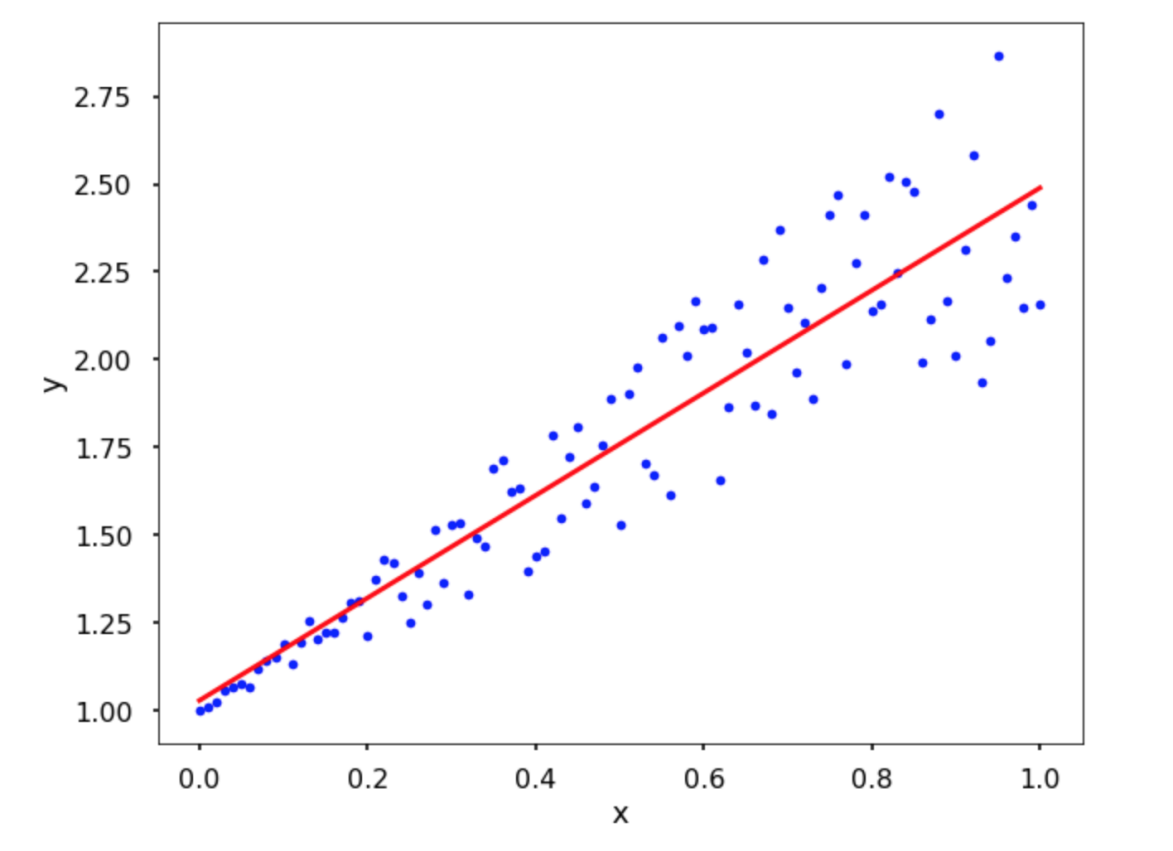
Dorim să construim o soluție aproximativă pentru problema de minim

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

De ce?

Pentru c ă dup ă cum am v ăzut, rezolvarea aproximativ ă a unui sistem de ecuații se reduce la o problem ă de optimizare, adic ă g ăsirea unei valori minime a unei funcții p ă tratice.

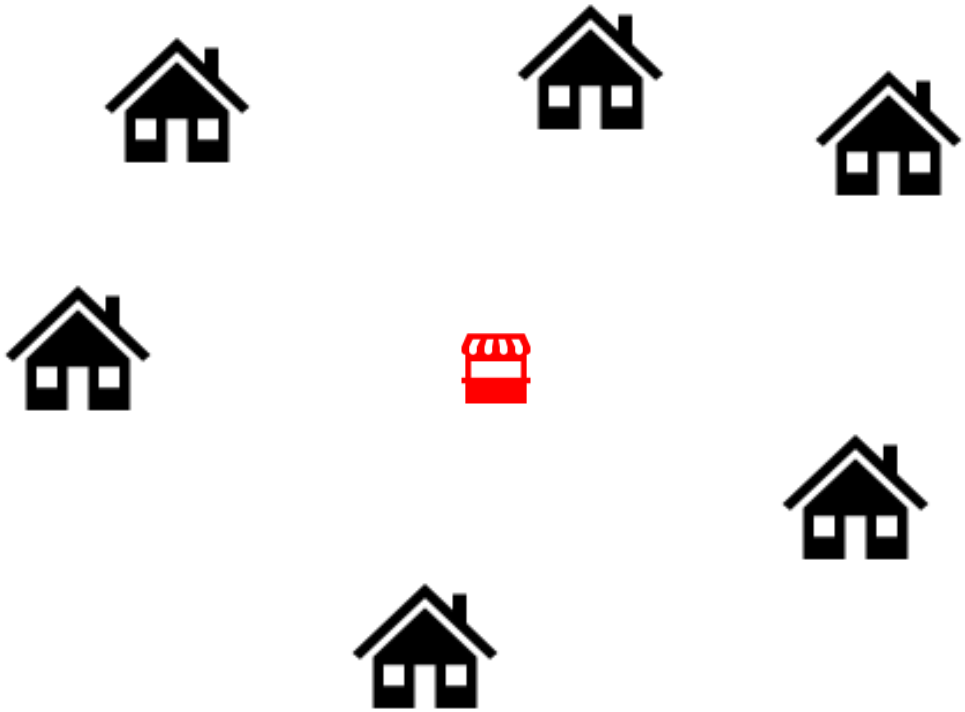
Corelarea datelor



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

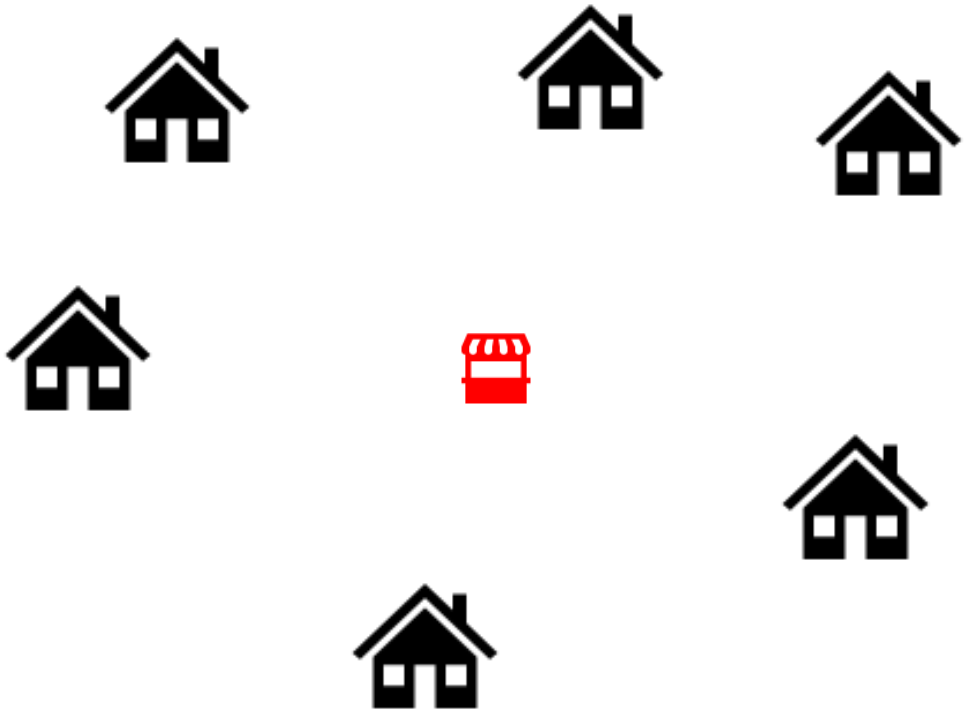
Alte probleme

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.

Metoda direcțiilor de descreștere

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

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

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

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

este satisfăcută.

Validitatea condiției de scădere suficientă

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

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 $k = 0, 1, 2, \ldots$ 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.

Numărul de condiționare pentru funcții pătratice. Rescalarea problemei.

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 $k = 0, 1, \ldots$.

$$f(x_{k+1}) \leq c^k \, f(x_0), \quad \text{where } c = \left(\frac{M-m}{M+m} \right)^2 = \left(\frac{\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} - 1}{\frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} + 1} \right)^2 .$$

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

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: $f(x) = \langle Ax, x \rangle + 2\langle b, x \rangle + c$ cu $L = 2\|A\|$, deoarece

$$\|\nabla f(x) - \nabla f(y)\| \leq 2\|Ax - Ay\| \leq 2\|A\|\,\|x - y\|.$$

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 $t_k = \bar{t} \in (0, \frac{2}{L})$. Înlocuind $x = x_k$, $t = \bar{t}$ în lema de scădere suficientă rezultă inegalitatea

$$f(x_k) - f(x_{k+1}) \geq \bar{t} \left(1 - \frac{L\bar{t}}{2}\right) \|\nabla f(x_k)\|^2.$$

Rețineți că descreștere în metoda gradientului pe iterație este

$$\bar{t} \left(1 - \frac{L\bar{t}}{2}\right) \|\nabla f(x_k)\|^2$$

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

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.

Pure Newton's Method, Damped Newton's Method, Hybrid Gradient-Newton Method

Pure Newton's Method

We note that the gradient method uses only first order information when the following unconstrained minimization problem is solved

$$\min\{f(x) : x \in \mathbb{R}^n\},$$

where we assumed that f is continuously differentiable.

Now, we assume that f is twice continuously differentiable and we present a second order method, i.e. we use information on both the gradient and the Hessian matrix.

We assume that $\nabla^2 f(x)$, $\forall x \in \mathbb{R}^n$ is positive definite, which implies that there exists a unique optimal solution x^*.

The main idea of Newton method is the following:

$$x_{k+1} = \operatorname{argmin}_{x\in\mathbb{R}^n} \Big\{ f_i(x_k) + \big\langle \nabla f_i(x_k), x - x_k \big\rangle + \frac{1}{2}\big\langle \nabla^2 f(x_k)(x - x_k), (x - x_k)\big\rangle \Big\}.$$

Note that since $\nabla^2 f(x_k)$ is positive definite, the above formula is well-defined. Why?

The unique minimizer of the minimization problem for finding x_{k+1} is the the unique stationary point

$$\nabla f(x_k) + \nabla^2 f(x_k)(x_{k+1} - x_k) = 0 \quad \Leftrightarrow \quad x_{k+1} = x_k \underbrace{-(\nabla^2 f(x_k))^{-1}\nabla f(x_k)}_{\text{Newton direction}}.$$

We remark that when $\nabla^2 f(x_k)$ is positive definite for any k, pure Newton's method is a scaled gradient method, and Newton's directions are descent directions.

The algorithm

Pure Newton's 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:
  - Compute the Newton direction d_k, which is the solution to the linear system $\nabla^2 f(x_k)\, d_k = -\nabla f(x_k)$.
  - Set $x_{k+1} = x_k + d_k$.
  - If $\|\nabla f(x_{k+1})\| \leq \varepsilon$, the STOP and x_{k+1} is the output.

Remark about its convergence

Even if the assumption $\nabla^2 f(x) \succ 0$, $\forall x \mathbb{R}^n$ implies that there exists a unique optimal solution x^*, this is not enough to guarantee convergence, see the following example.

Consider the function $f(x) = \sqrt{1 + x^2}$ defined over the real line. The minimizer of f over $\mathbb{R}$ is of course $x = 0$. The first and second derivatives of f are $f'(x) = \frac{x}{\sqrt{1+x^2}}$, $f''(x) = \frac{1}{\sqrt{(1+x^2)^3}}$. Therefore, (pure) Newton's method has the form

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)} = -x_k^3.$$

Therefore, if $|x_0| \geq 1$ the method diverges and that for $|x_0| < 1$ the method converges very rapidly to the correct solution $x^* = 0$.

Quadratic Local convergence of Newton's method

Theorem

Let f be a twice continuously differentiablle function defined over $\mathbb{R}^n$. Assume that

i) there exists $m > 0$ for which $\nabla^2 f(x) \succeq m\, I$ for any $x \in \mathbb{R}^n$,

ii) there exists $L > 0$ for which $\|\nabla^2 f(x) - \nabla^2 f(y)\| \leq L\|x - y\|$ for any $x, y \in \mathbb{R}^n$, where the considered matrix norm is the spectral norm.

Let $\{x_k\}_{k\geq 0}$ be the sequence generated by Newton's method, and let x^* be the unique minimizer of f over $\mathbb{R}^n$. Then for any $k = 0, 1, 2, ...$ the inequality

$$\|x_{k+1} - x^*\| \leq \frac{L}{2\,m}\,\|x_k - x^*\|^2$$

holds. In addition, if $\|x_0 - x^*\| \leq \frac{m}{L}$, then

$$\|x_k - x^*\| \leq \frac{2\,m}{L}\left(\frac{1}{2}\right)^{2^k}, \quad k = 0, 1, 2.....$$

Proof

Let k be a nonnegative integer. Then

$$\begin{aligned}
x_{k+1} - x^* =& x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k) - x^* \\
\overset{\nabla f(x^*)=0}{=}& x_k - x^* + (\nabla^2 f(x_k))^{-1} [\nabla f(x^*) - \nabla f(x_k)] \\
=& x_k - x^* + (\nabla^2 f(x_k))^{-1} \int_0^1 \nabla^2 f(x_k + t\,(x^* - x_k))\,(x^* - x_k)\,dt \\
=& (\nabla^2 f(x_k))^{-1} \int_0^1 [\nabla^2 f(x_k + t\,(x^* - x_k)) - \nabla^2 f(x_k)]\,(x^* - x_k)\,dt.
\end{aligned}$$

Combining the latter equality with the fact that $\nabla^2 f(x_k) \succeq m\,I$ implies that $\|(\nabla^2 f(x_k)^{-1}\| \leq \frac{1}{m}$, we deduce

Proof

$$
\begin{aligned}
\|x_{k+1} - x^*\| \leq & \|(\nabla^2 f(x_k))^{-1}\| \left\| \int_0^1 [\nabla^2 f(x_k + t\,(x^* - x_k)) - \nabla^2 f(x_k)]\,(x^* - x_k)\,dt \right\| \\
\leq & \|(\nabla^2 f(x_k))^{-1}\| \int_0^1 \left\| [\nabla^2 f(x_k + t\,(x^* - x_k)) - \nabla^2 f(x_k)]\,(x^* - x_k) \right\| dt \\
\leq & \|(\nabla^2 f(x_k))^{-1}\| \int_0^1 \left\| [\nabla^2 f(x_k + t\,(x^* - x_k)) - \nabla^2 f(x_k)] \right\| \|(x^* - x_k)\|\,dt \\
\leq & \frac{L}{m} \int_0^1 t\,\|(x^* - x_k)\|^2\,dt = \frac{L}{2\,m}\,\|(x^* - x_k)\|^2 .
\end{aligned}
$$

We will show our desired estimate by induction on k.

Proof

Since we have assumed that

$$\|x_0 - x^*\| \leq \frac{m}{L},$$

so in particular

$$\|x_0 - x^*\| \leq \frac{2\,m}{L}\left(\frac{1}{2}\right)^{2^{\mathbf{0}}},$$

establishing the first step of the induction. Assume that the estimate holds for a given k, that is

$$\|x_k - x^*\| \leq \frac{2\,m}{L}\left(\frac{1}{2}\right)^{2^k},$$

we will show it holds for $k+1$. Indeed, we have

$$\|x_{k+1} - x^*\| \leq \frac{L}{2\,m}\|x_k - x^*\|^2 \leq \frac{L}{2\,m}\left(\frac{2\,m}{L}\left(\frac{1}{2}\right)^{2^k}\right)^2 = \frac{2\,m}{L}\left(\frac{1}{2}\right)^{2^{k+\mathbf{1}}}.\square$$

However, in general, convergence is unfortunately not guaranteed in the absence of these very restrictive assumptions and the implementation must includes a divergence is criteria, e.g.the number of iterations is smaller than 10000.

Pure Newton's method does not guarantee descent of the generated sequence of function values even when the Hessian is positive definite, e.g. for

$$\min_{x1,x_2 \in \mathbb{R}} \sqrt{x_1^2 + 1} + \sqrt{x_2^2 + 1},$$

when the basic assumption $\nabla^2 f(x) \succeq m\, I$ is violated.

Damped Newton's Method

What is missing is a stepsize chosen, leading to the so-called damped Newton's Method.

Damped Newton's Method

- **Input**: $\varepsilon > 0$ as the tolerance parameter. $\alpha, \beta \in (0,1)$-parameters for the backtracking procedure.
- **Initialization**: Pick $x_0 \in \mathbb{R}^n$ arbitrarily.
- **General step**: For any $k = 0, 1, 2, \ldots$ execute the following steps:
  - Compute the Newton direction d_k, which is the solution to the linear system $\nabla^2 f(x_k)\, d_k = -\nabla f(x_k)$.
  - Set $t_k = 1$. While $f(x_k) - f(x_k + t_k d_k) < -\alpha\, t_k\, \nabla f(x_k)^T d_k$ set $t_k = \beta\, t_k$,
  - Set $x_{k+1} = x_k + t_k\, d_k$.
  - If $\|\nabla f(x_{k+1})\| \leq \varepsilon$, the STOP and x_{k+1} is the output.

The Cholesky Factorization

There are some important issue that naturally arises when employing Newton's method:

- one of validating whether the Hessian matrix is positive definite;
- if it is, then how to solve the linear system $\nabla^2 f(x_k)\, d_k = -\nabla f(x_k)$.

These two issues are resolved by using the Cholesky factorization, which means that given a matrix $A \in \mathbb{R}^{n\times n}$, find a lower triangular matrix $L \in \mathbb{R}^{n\times n}$ whose diagonal is positive, such that

$$A = L\, L^T.$$

Given a Cholesky factorisation, the task of solving a linear system of equations of the form $A\, x = b$ can be easily done by the following two steps:

- Find the solution of the lower triangular algebraic system $L\, u = b$.
- Find the solution of the upper triangular algebraic system $L^T\, x = u$.

The process of computing the Cholesky factorization is well-defined as long as all the diagonal ellements l_{ii} that are computed during the process are positive, so that computing their square roots is possible.

The positiveness of these elements is equivalent to the property that the matrix to be factored is positive definite.

Therefore, the Cholesky factorization process can be viewed as a criteria for positive definiteness, and it is actually the test that is used in many algorithms.

The Cholesky factorization is not the main aim of this course, so we will use the Matlab command `chol(A,'lower')`.

Hybrid Gradient-Newton Method

Newton's method (pure or not) assumes that the Hessian matrix is positive definite and we are thus left with the question of how to employ Newton's method when the Hessian is not always positive definite.

There are several ways to deal with this situation, but perhaps the simplest one is to construct a hybrid method that employs either a Newton step at iterations in which the Hessian is positive definite or a gradient step when the Hessian is not positive definite.

Hybrid Gradient-Newton Method

Hybrid Gradient– Newton Method

- **Input**: $\varepsilon > 0$ as the tolerance parameter. $\alpha, \beta \in (0,1)$-parameters for the backtracking procedure.
- **Initialization**: Pick $x_0 \in \mathbb{R}^n$ arbitrarily.
- **General step**: For any $k = 0, 1, 2, \ldots$ execute the following steps:
  - Compute the Newton direction d_k as in the following
    - If $\nabla^2 f(x_k) \succ 0$, then take d_k as the Newton direction, which is the solution to the linear system $\nabla^2 f(x_k)\, d_k = -\nabla f(x_k)$.
    - Otherwise, set $d_k = -\nabla f(x_k)$.
  - Set $t_k = 1$. While $f(x_k) - f(x_k + t_k d_k) < -\alpha\, t_k\, \nabla f(x_k)^T d_k$ set $t_k = \beta\, t_k$,
  - Set $x_{k+1} = x_k + t_k\, d_k$.
  - If $\|\nabla f(x_{k+1})\| \leq \varepsilon$, the STOP and x_{k+1} is the output.

Fermat-Weber problem

Fermat problem

Pierre de Fermat posed the following problem:

> *Given three distinct points in the plane, find the point having the minimal sum of distances to these three points.*

The italian physicist Torricelli solved this problem and defined a construction by ruler and compass for finding it. This point is called *the Torricelli point* or *the Torricelli-Fermat point*.

Fermat-Weber problem

Later on, the Fermat problem was generalized by the German economist Weber to a problem in the space $\mathbb{R}^n$ and with an arbitrary number of points.

The problem known today as "the Fermat-Weber problem" is the following:

> *Given m distinct points $a_1, a_2, ..., a_m$ in $\mathbb{R}^n$, also called "anchor points"- and m weights $\omega_1, \omega_2, ..., \omega_m > 0$, find a point $x \in \mathbb{R}^n$ that minimizes the weighted distance of x to each of the points $a_1, a_2, ..., a_m$.*

Mathematically, the Fermat-Weber Problem can be cast as the minimization

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

Note that the objective function is not differentiable at the anchor points $a_1, a_2, ..., a_m$.

Weiszfeld's solution (1937)

The starting point is the first order optimality condition

$$\nabla f(x) = 0 \,\forall\, x \notin \{a_1, a_2, ..., a_m\} \Leftrightarrow \sum_{i=1}^{m} \omega_i \frac{x - a_i}{\|x - a_i\|} = 0 \,\forall\, x \notin \{a_1, a_2, ..., a_m\}.$$

After some algebraic manipulation, the latter relation can be written as

$$\left(\sum_{i=1}^{m} \frac{\omega_i}{\|x - a_i\|} \right) x = \sum_{i=1}^{m} \frac{\omega_i \, a_i}{\|x - a_i\|} \quad \forall\, x \notin \{a_1, a_2, ..., a_m\},$$

which gives us

$$x = \frac{1}{\left(\sum_{i=1}^{m} \frac{\omega_i}{\|x - a_i\|} \right)} \sum_{i=1}^{m} \frac{\omega_i \, a_i}{\|x - a_i\|} \quad \forall\, x \notin \{a_1, a_2, ..., a_m\}.$$

We can reformulate the optimality condition as a fixed point problem

$$x = T(x),$$

where T is the operator

$$T(x) = \frac{1}{\left(\sum_{i=1}^{m} \frac{\omega_i}{\|x - a_i\|}\right)} \sum_{i=1}^{m} \frac{\omega_i \, a_i}{\|x - a_i\|} \quad \forall \, x \notin \{a_1, a_2, ..., a_m\}.$$

For a fixed point problem, a natural approach for solving the problem is via the iterations

$$x_{k+1} = T(x_k).$$

The algorithm

Weiszfeld's Method

- Initialization: Pick $x_0 \in \mathbb{R}^n$ such that $x_0 \notin \{a_1, a_2, ..., a_m\}$.
- General step: For any $k = 0, 1, 2, ...$ compute $x_{k+1} = T(x_k)$.

!? Note that the algorithm is defined only when the iterates x_k are all different from $a_1, a_2, ..., a_m$.

Although the algorithm was initially presented as a fixed point method, the surprising fact is that it is basically a gradient method by writting

$$\begin{aligned} x_{k+1} &= \frac{1}{\left(\sum_{i=1}^{m} \frac{\omega_i}{\|x_k - a_i\|}\right)} \sum_{i=1}^{m} \frac{\omega_i\, a_i}{\|x_k - a_i\|} \\ &= x_k - \frac{1}{\left(\sum_{i=1}^{m} \frac{\omega_i}{\|x_k - a_i\|}\right)} \sum_{i=1}^{m} \omega_i \frac{x_k - a_i}{\|x_k - a_i\|} \\ &= x_k - \frac{1}{\left(\sum_{i=1}^{m} \frac{\omega_i}{\|x_k - a_i\|}\right)} \nabla f(x_k). \end{aligned}$$

Therefore, Weiszfeld's method is essentially the gradient method with a special choice of stepsize:

$$t_k = \frac{1}{\left(\sum_{i=1}^{m} \frac{\omega_i}{\|x_k - a_i\|}\right)}.$$

Questions!

- Is the method well-defined?
- That is, can we guarantee that none of the iterates x_k is equal to any of the points $a_1, a_2, ..., a_m$?
- The stepsize is not too large? Is the sequence of objective function values decreases?
- Does the sequence $\{x_k\}_{k \geq 0}$ converge to a global optimal solution?

The first aim is to show that the generated sequence of function values is nonincreasing.

for that, we define the auxiliary function

$$h(y,x) := \sum_{i=1}^{m} \omega_i \frac{\|y - a_i\|^2}{\|x - a_i\|}, \quad y \in \mathbb{R}^n, \quad x \in \mathbb{R}^n \setminus \mathcal{A},$$

where $\mathcal{A} = \{a_1, a_2, ..., a_m\}$.

Lemma

For any $x \in \mathbb{R}^n \setminus \mathcal{A}$, one has

$$T(x) = \operatorname{argmin}_{y \in \mathbb{R}^n} \{h(y,x) \,:\, y \in \mathbb{R}^n\}.$$

Lemma

For any $x \in \mathbb{R}^n \setminus \mathcal{A}$, one has

$$T(x) = \operatorname{argmin}_{y \in \mathbb{R}^n} \{h(y, x) \,:\, y \in \mathbb{R}^n\}.$$

Proof. The function $y \mapsto h(y, x)$ is a quadratic function whose associated matrix is $\left(\sum_{i=1}^m \frac{\omega_i}{\|x - a_i\|}\right) I_n$, which is clear positive definite.

Therefore, the unique global minimimum of the mapping $y \mapsto h(y, x)$, which we denote by y^*, is the unique stationary point of $y \mapsto h(y, x)$, that is, the point for which the gradient vanishes:

$$\nabla_y h(y^*, x) = 0.$$

Thus,

$$2 \sum_{i=1}^m \omega_i \frac{y^* - a_i}{\|x - a_i\|} = 0.$$

Extracting y^* from above yields $y^* = T(x)$, and the result is established.

Therefore, the above lemma tells us that the iterations of Weiszfeld's method read

$$x_{k+1} = \operatorname{argmin}\{h(y, x_k) \,:\, y \in \mathbb{R}^n\}.$$

In the next lemma we prove that the sequence $\{f(x_k)\}_{k\geq 0}$ is nonincreasing.

Lemma

If $x \in \mathbb{R}^n \setminus \mathcal{A}$, then

1. $h(x, x) = f(x)$;
2. $h(y, x) \geq 2\, f(y) - f(x)$ for any $y \in \mathbb{R}^n$;
3. $f(T(x)) \leq f(x)$ and $f(T(x)) = f(x)$ if and only if $x = T(x)$;
4. $x = T(x)$ if and only if $\nabla f(x) = 0$.

1. $h(x,x) = \sum_{i=1}^{m} \omega_i \frac{\|x-a_i\|^2}{\|x-a_i\|} = \sum_{i=1}^{m} \omega_i \|x-a_i\| = f(x)$;
2. For any nonnegative number a and positive number b, the inequality

$$\frac{a^2}{b} \geq 2\,a - b$$

holds. Substituting $a = \|y-a_i\|$ and $b = \|x-a_i\|$, it follows that any $i = 1, 2, ..., m$

$$\frac{\|y-a_i\|^2}{\|x-a_i\|} \geq 2\,\|y-a_i\| - \|x-a_i\|.$$

Hence

$$\sum_{i=1}^{m} \omega_i \frac{\|y-a_i\|^2}{\|x-a_i\|} \geq 2 \sum_{i=1}^{m} \omega_i \|y-a_i\| - \sum_{i=1}^{m} \omega_i \|x-a_i\|.$$

Therefore, $h(y,x) \geq 2\,f(y) - f(x)$ for any $y \in \mathbb{R}^n$;

3. Since $T(x) = \operatorname{argmin}_{y \in \mathbb{R}^n} h(y, x)$, it follows from 1. that

$$h(T(x), x) \leq h(x, x) = f(x).$$

Using 2. we have

$$h(T(x), x) \geq 2 f(T(x)) - f(x),$$

which implies

$$f(x) \geq h(T(x), x) \geq 2 f(T(x)) - f(x)$$

establishing the fact that

$$f(x) \geq f(T(x)).$$

To complete the proof of this item, we need to show that $f(T(x)) = f(x)$ if and only if $T(x) = x$. Of course, if $T(x) = x$, then $f(T(x)) = f(x)$. To show the reverse implication, let us assume that $f(T(x)) = f(x)$.
By the chain of

$$f(x) \geq h(T(x), x) \geq 2\, f(T(x)) - f(x)$$

it follows that $f(T(x)) = f(x)$ implies
$h(T(x), x) = f(x) = f(T(x))$. Since the unique minimizer of $y \mapsto h(y, x)$ is $T(x)$, it follows that $x = T(x)$.

4. Clear.

Lemma

Let $\{x_k\}_{k\geq 0}$ be the sequence generated by Weiszfeld's method, where we assume that $x_k \notin \mathcal{A}$ for all $k \geq 0$. Then we have the following:

1. The sequence $\{f(x_k)\}_{k\geq 0}$ is noincreasing: for any $k \geq 0$ the inequality $f(x_{k+1}) \leq f(x_k)$ holds.
2. For any k, $f(x_{k+1}) = f(x_k)$ if and only if $\nabla f(x_k) = 0$.

Proof.

1. Since $x_k \notin \mathcal{A}$ for all $k \geq 0$, this result follows by substituting $x = x_k$ in the previous lemma.
2. From the previous lemma it follows that for any k, $f(T(x_k)) = f(x_{k+1}) = f(x_k)$ if and only if $x_k = x_{k+1} = T(x_k)$, which is equivalent to $\nabla f(x_k) = 0$.

We have shown that $\{f(x_k)\}_{k\geq 0}$ is noincreasing as long as we are not stuck at a stationary point. The underlying assumption that $x_k \notin \mathcal{A}$ is problematic in the sense that it cannot be controlled easily.

One approach to make sure that the sequence generated by the method does not contain anchor points is to choose the strating point x_0 so that its values is strictly smaller than the values of the anchor points:

$$f(x_0) < \min\{f(a_1), f(a_2), ..., f(a_m)\}.$$

This assumption, combined with the monotonicity of the function values of the sequence generated by the method, implies that the iterates do not include anchor points.

Theorem

Let $\{f(x_k)\}_{k\geq 0}$ be the sequence generated by Weiszfeld's method and assume that $f(x_0) < \min\{f(a_1), f(a_2), ..., f(a_m)\}$. Then all the limit points of $\{x_k\}_{k\geq 0}$ are stationary points of f.

Proof. Let $\{x_{k_n}\}_{n\geq 0}$ be a subsequence of $\{x_k\}_{k\geq 0}$ that converge to a point x^*. By the monotonicity of the method and the continuity of the objective function we have $f(x^*) \leq f(x_0) < \min\{f(a_1), f(a_2), ..., f(a_m)\}$. Therefore, $x^* \notin \mathcal{A}$, and hence $\nabla f(x^*)$ is defined. We will show that $\nabla f(x^*) = 0$. By the continuity of the operator T at x^*, it follows that the sequence $x_{k_n+1} = T(x_{k_n}) \to T(x^*)$ as $n \to \infty$. The sequence of function values $\{f(x_k)\}_{k\geq 0}$ is noincreasing and bounded below by 0 and thus converges to a value which we denote by f^*. Obviously, both $\{f(x_{k_n})\}_{n\geq 0}$ and $\{f(x_{k_n+1})\}_{n\geq 0}$ converge to f^*. By the continuity of f, we thus obtain that $f(T(x^*)) = f(x^*) = f^*$, which leads to $T(x^*) = x^*$ and $\nabla f(x^*) = 0$.

In fact, the stationary points are global optimal solutions, but this analysis is beyond our lectures.

Linear Programming

Formulating the Problem and a Graphical Solution, Discussion on possible approaches

A simple problem, for motivation [1] i

A family run a business that produced and sells dairy products from the milk of the family cows, Algebra, Analysis and Geometry. Together, the three cows produces 22 gallons of milk each week, and the family turn the milk into ice cream and butter that they then sell at the Farmer's Market each Saturday morning.

The butter-making process requires 2 gallons of milk to produce on kilogram of butter, and 3 gallons of milk is required to make one gallon of ice cream. The family has a huge refrigerator that can store practically unlimited amounts of butter, but his freezer can hold at most 6 gallons of ice cream.

A simple problem, for motivation [1] ii

The family has at most 6 hours per week in total to spend on manufacturing their delicious products. One hours of work is needed to produce either 4 gallons of ice cream or one kilogram of butter. Any fraction of one hour is needed to produce the corresponding fraction of product.

The family's products have a great reputation, and he always sells everything he produces. He sets the prices to ensure a profit of \$5 per gallon of ice cream and \$4 per kilogram of butter.

He would like to figure out how much ice cream and butter he should produce to maximize his profit.

Exercise

Solve the above problem by using the already known methods.

The Professor's dairy: Constraints and objective

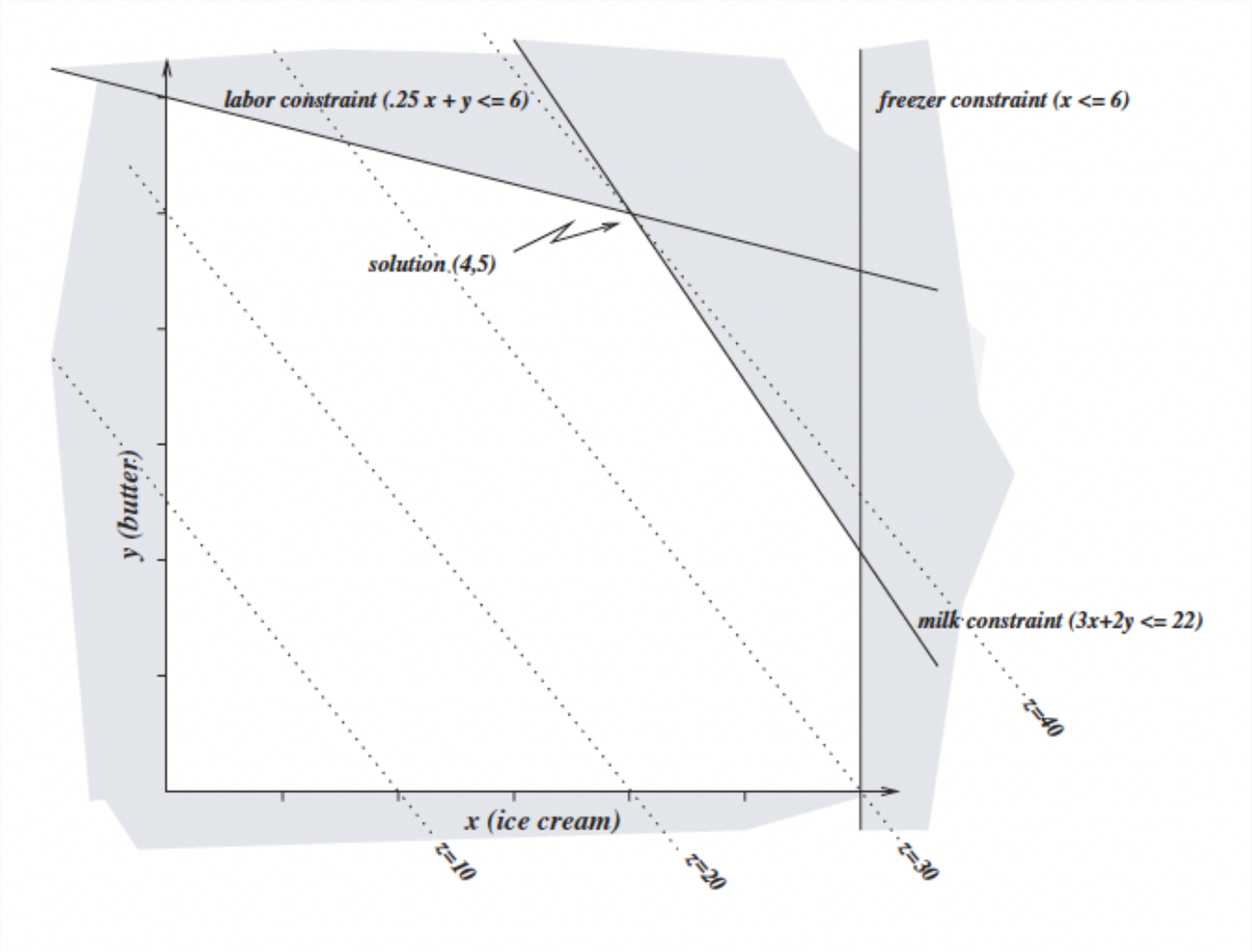
The algebraic form of the professor's dairy problem is:

$$\begin{aligned} \max_{x_1, x_2 \in \mathbb{R}} \quad & z = 5\,x_1 + 4\,x_2 \\ \text{subject to} \quad & x_1 \leq 6, \\ & 0.25\,x_1 + x_2 \leq 6, \\ & 3\,x_1 + 2\,x_2 \leq 22, \\ & x_1, x_2 \geq 0. \end{aligned}$$

We call z, which in this case is the profit, the *objective*.

The set of points satisfying all five of the constraints is known as the *feasible region*. In this problem the feasible region is the five-sided polygonal region, see the next figure.

A geometrical view of the problem



The linear programming problem is to find a point in the feasible region that maximizes the objective $z = 5\,x_1 + 4\,x_2$. As a step towards this goal, we plot in Figure a dotted line representing the set of points at which $z = 20$. This line indicates feasible points such as $(x_1, x_2) = (0, 5)$ and $(x_1, x_2) = (2, 2.5)$ that yield a profit of 20 dolars.

The Setup

Linear programming or linear programs

The optimization problems considered in this Section have some particularities:

- Their variables can take any real values, subject to satisfying the bounds and constraints.
- All constraints an bounds involve linear functions of the variables.
- The objective function (*called profit*) is also a linear function of the variable.

Formulation in the standard form

Linear programs in standard form

$$
\begin{aligned}
\min_{x1,x_2,\ldots,x_n \in \mathbb{R}} \quad & z = p_1 x_1 + p_2 x_2 + \ldots + p_n x_n \\
\text{subject to} \quad & A_{11} x_1 + \ldots + A_{1n} x_n \geq b_1, \\
& \vdots \qquad\quad \ddots \qquad \vdots \qquad\quad \vdots \\
& A_{m1} x_1 + \ldots + A_{mn} x_n \geq b_m, \\
& x_1, x_2, \ldots, x_n \geq 0.
\end{aligned}
$$

Formulation in the standard form

By grouping the variables into vectors and matrices, i.e.

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad A = \begin{pmatrix} A_{11} & \dots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{m1} & \dots & A_{mn} \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}, \quad p = \begin{pmatrix} p_1 \\ \vdots \\ p_n, \end{pmatrix}$$

the linear programs in standard form becomes.

In matrix formulation

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & z = \langle p, x \rangle \\ \text{subject to} \quad & A x \geq b \quad x \geq 0, \end{aligned}$$

where " $\geq$ " is considered on each component.

The canonical form

The canonical form of a linear program is

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & z = \langle p, x \rangle \\ \text{subject to} \quad & Ax = b \quad x \geq 0. \end{aligned}$$

A linear programming written in standard form can be converted in a canonical form by introducing the so called slack variables, suggested by the difference $Ax - b$.

Tableau representation of the problem, Vertices

The settings

All the linear programs can be reduce to the following

Standard form

$$\min_{x \in \mathbb{R}^n} \quad z = \langle p, x \rangle$$
$$\text{subject to} \quad Ax \geq b \quad x \geq 0,$$

where $p \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and $A \in \mathbb{R}^{m \times n}$.

To create the initial tableau for the simplex method, we rewrite the problem in the following

Canonical form

$$\begin{aligned} \min_{x \in \mathbb{R}^{n+m}} \quad & z = \langle p, x_N \rangle + \langle 0, x_B \rangle \\ \text{subject to} \quad & x_B = A x_N - b \quad x_B, x_N \geq 0, \end{aligned}$$

where the index sets N and B are defined initially as $N = \{1, 2, ..., n\}$ and $B = \{n+1, n+2, ..., n+m\}$.

The variables $x_{n+1}, ..., x_{n+m}$ are introduced to represent the slack in the inequalities $Ax \geq b$ and they are called *slack variables*.

Tableau representation of the problem

We shall represent this canonical linear program by the following tableau

$$
\begin{array}{ccc|ccc|c}
 & & & x_1 & \cdots & x_n & 1 \\
\hline
x_{n+1} & = & & A_{11} & \cdots & A_{1n} & -b_1 \\
\vdots & & & \vdots & \ddots & \vdots & \vdots \\
x_{n+m} & = & & A_{m1} & \cdots & A_{mn} & -b_m \\
\hline
z & = & & p_1 & \cdots & p_n & 0
\end{array}
$$

In this tableau

- $x_{n+1}, \ldots, x_{n+m}$ are the dependent variables, called *basic*;
- $x_1, \ldots, x_n$ are the independent variables, called *nonbasic*.

A more succinct form of the initial tableau is know as the condensed tableau, which is written as follows

$$\begin{array}{rcc} & x_N & 1 \\ x_B = & A & -b \\ z = & p^T & 0 \end{array}$$

We "read" a tableau by setting the nonbasic variables x_N to zero, thus assigning the basic variables x_B and the objective variable z the values in the last columns of the tableau.

Thus, the tableau above represents the point $x_N = 0$ and $x_B = -b$ with the objective of $z = 0$.

The tableau is said to be *feasible* if the values assigned to the basic variables by this procedure are nonnegative. In the above, the tableau will be feasible if $b \leq 0$.

A Simple Example

Linear programs in standard form

$$
\begin{aligned}
\min_{x1, x_2 \in \mathbb{R}} \quad & z = 3\,x_1 - 6\,x_2 \\
\text{subject to} \quad & x_1 + 2x_2 \geq -1, \\
& 2\,x_1 + x_2 \geq 0, \\
& x_1 - x_2 \geq -1, \\
& x_1 - 4\,x_2 \geq -13, \\
& -4\,x_1 + x_2 \geq -23, \\
& x_1, x_2 \geq 0.
\end{aligned}
$$

Linear programs in canonical form

The first step is to add slack variables to convert the constraints into a set of general equalities combined with nonnegativity requirements on all the variables. The slacks are defined as follows:

$$
\begin{aligned}
x_3 &= x_1 + 2x_2 - 1, \\
x_4 &= 2\,x_1 + x_2, \\
x_5 &= x_1 - x_2 + 1, \\
x_6 &= x_1 - 4\,x_2 + 13, \\
x_7 &= -\,4\,x_1 + x_2 + 23,
\end{aligned}
$$

Tableau representation of the linear program

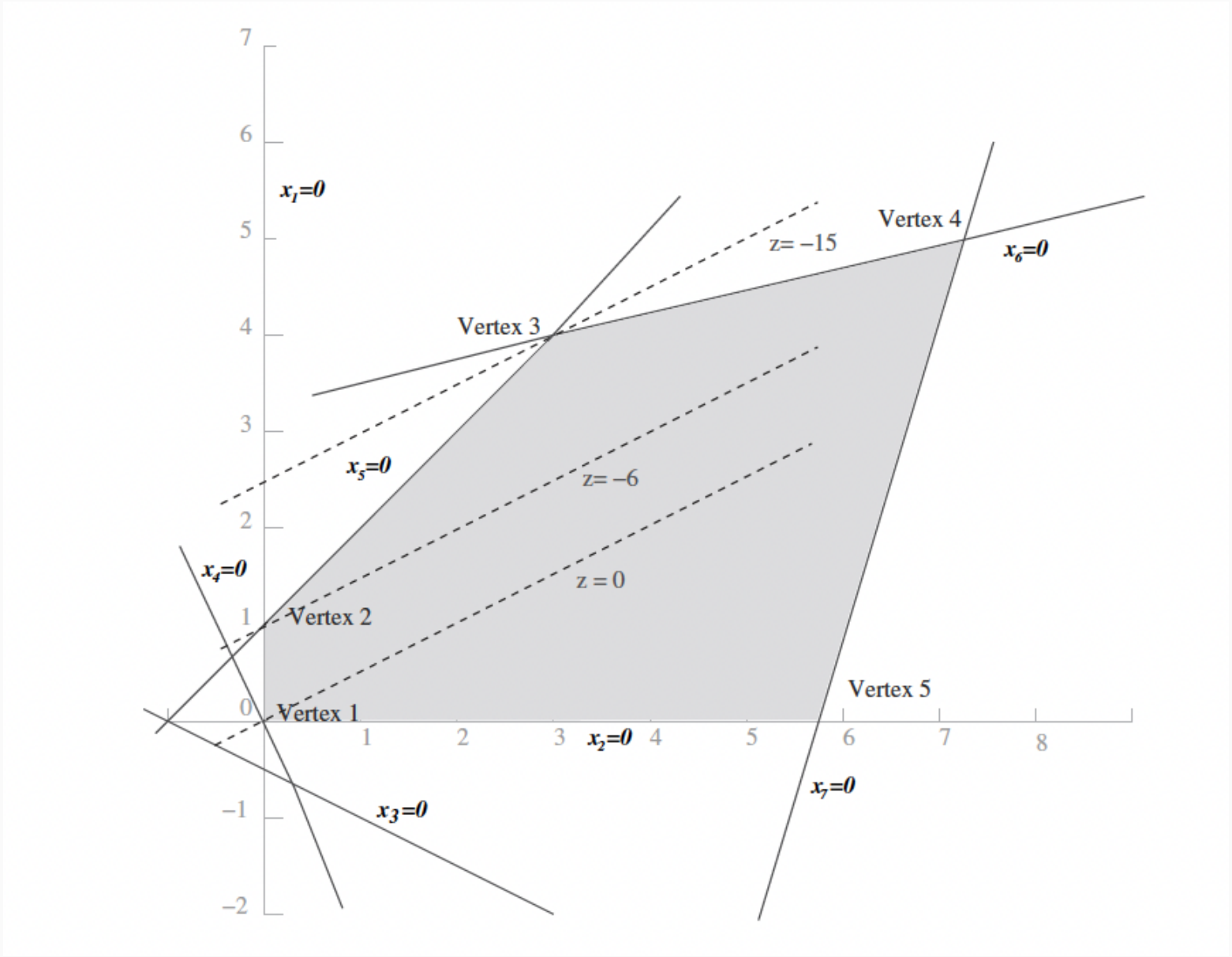
		x_1	x_2	1
x_3	$=$	1	2	1
x_4	$=$	2	1	0
x_5	$=$	1	-1	1
x_6	$=$	1	-4	13
x_7	$=$	-4	1	23
z	$=$	3	-6	0

In MATLAB we store this tableau in a structure.

We may set a starting searching point by setting (the nonbasic variables) $x_1 = 0$ and $x_2 = 0$, since the table is feasible, i.e. the corresponding slack variables (basic) remain positive.

The values of the objective in this tableau, $z = 0$, is obtained from the bottom right element.

Let us move from this starting position! Why and How?


7
6
5
4
3
2
1
0
−1
−2
1
2
3
4
5
6
7
8
$x_1=0$
$x_2=0$
$x_3=0$
$x_4=0$
$x_5=0$
$x_6=0$
$x_7=0$
Vertex 1
Vertex 2
Vertex 3
Vertex 4
Vertex 5
z= −15
z= −6
z = 0

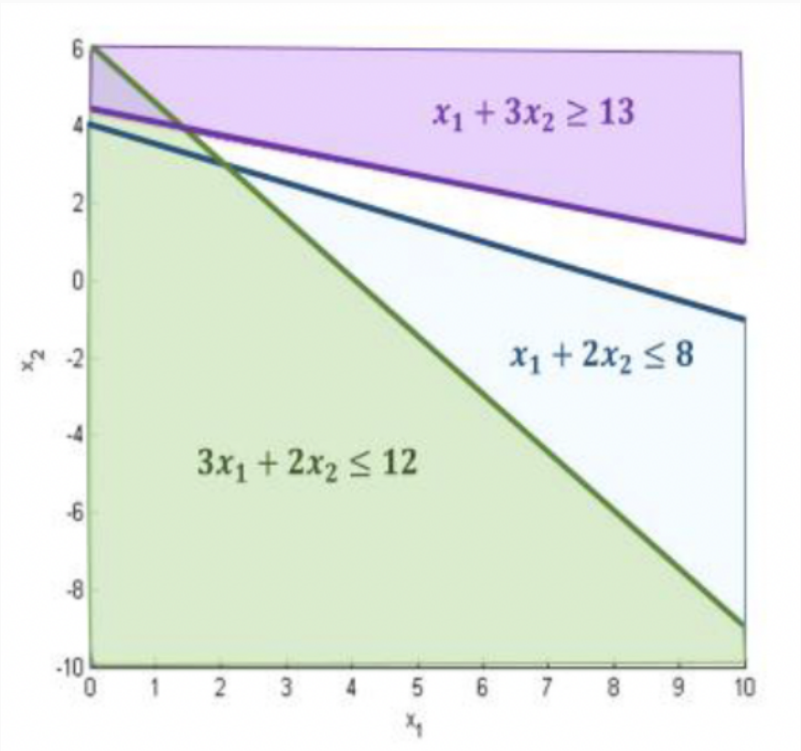
Why to move from vertex to vertex and not through all feasible domain?

All possibilities for a linear programming

Infeasible case

$$\min_{x1, x_2 \in \mathbb{R}} \quad z = x_1 + x_2$$

$$\begin{aligned} \text{subject to} \quad & x_1 + 2\,x_2 \leq 8, \\ & 3\,x_1 + 2\,x_2 \leq 12, \\ & x_1 + 3\,x_2 \geq 13, \\ & x_1, x_2 \geq 0. \end{aligned}$$

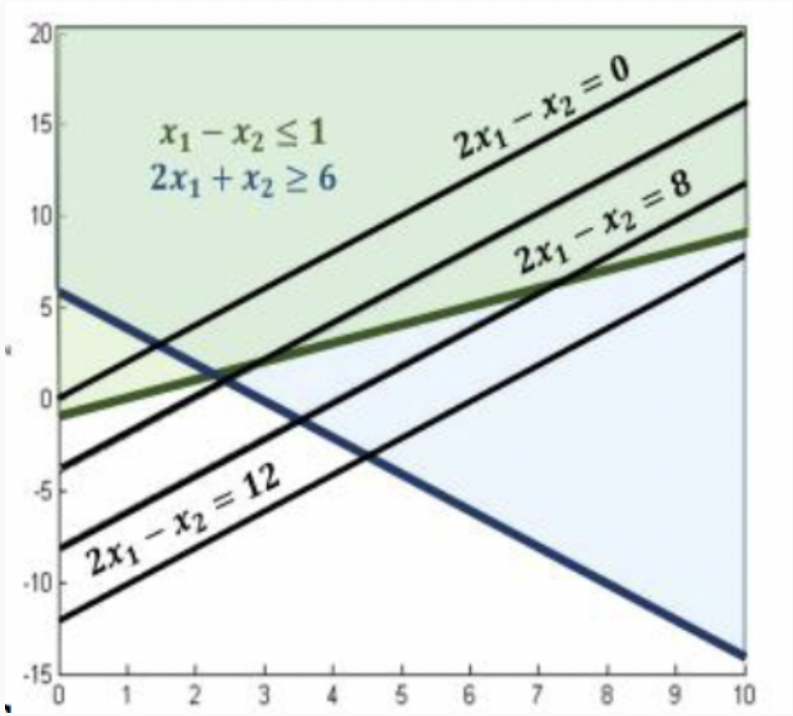


All possibilities for a linear programming

Unbounded case

$$\min_{x1, x_2 \in \mathbb{R}} \quad z = 2\,x_1 - x_2$$

$$\begin{aligned} \text{subject to} \quad x_1 \;-\; x_2 &\leq 1, \\ 2\,x_1 \;+\; x_2 &\geq 6, \\ x_1, x_2 &\geq 0. \end{aligned}$$

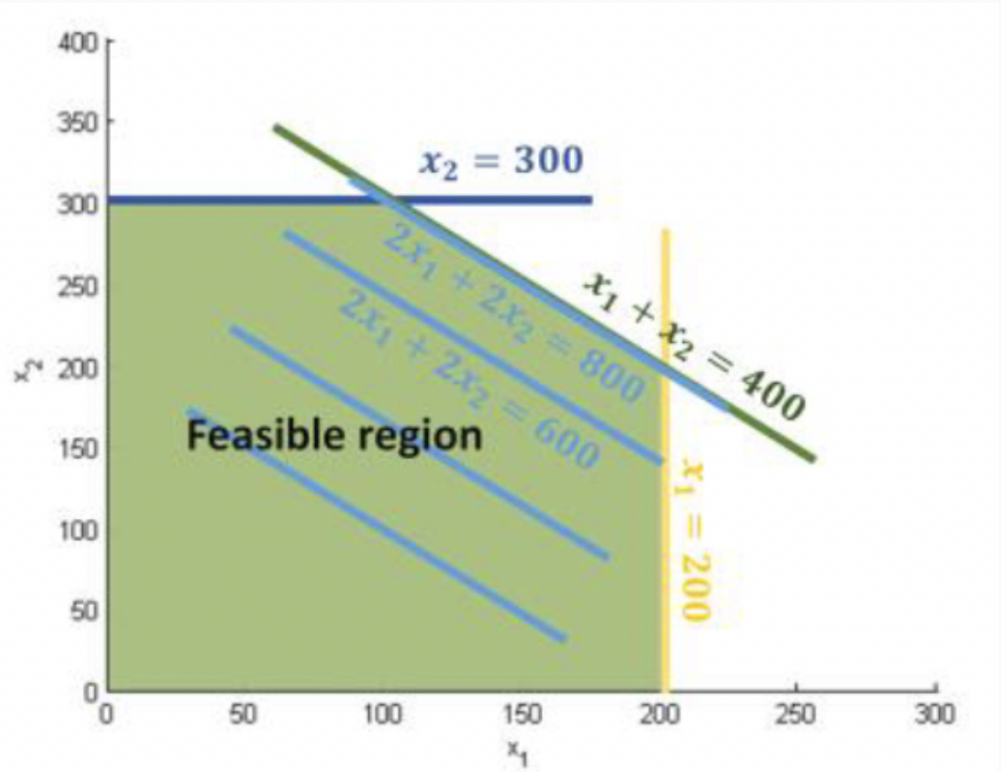


All possibilities for a linear programming

Infinite number of optimal solutions

$$\min_{x1, x_2 \in \mathbb{R}} \quad z = 2\,x_1 + 2\,x_2$$

$$\begin{aligned} \text{subject to} \quad x_1 &\leq 200, \\ x_2 &\leq 300, \\ x_1 + x_2 &\leq 400, \\ x_1, x_2 &\geq 0. \end{aligned}$$

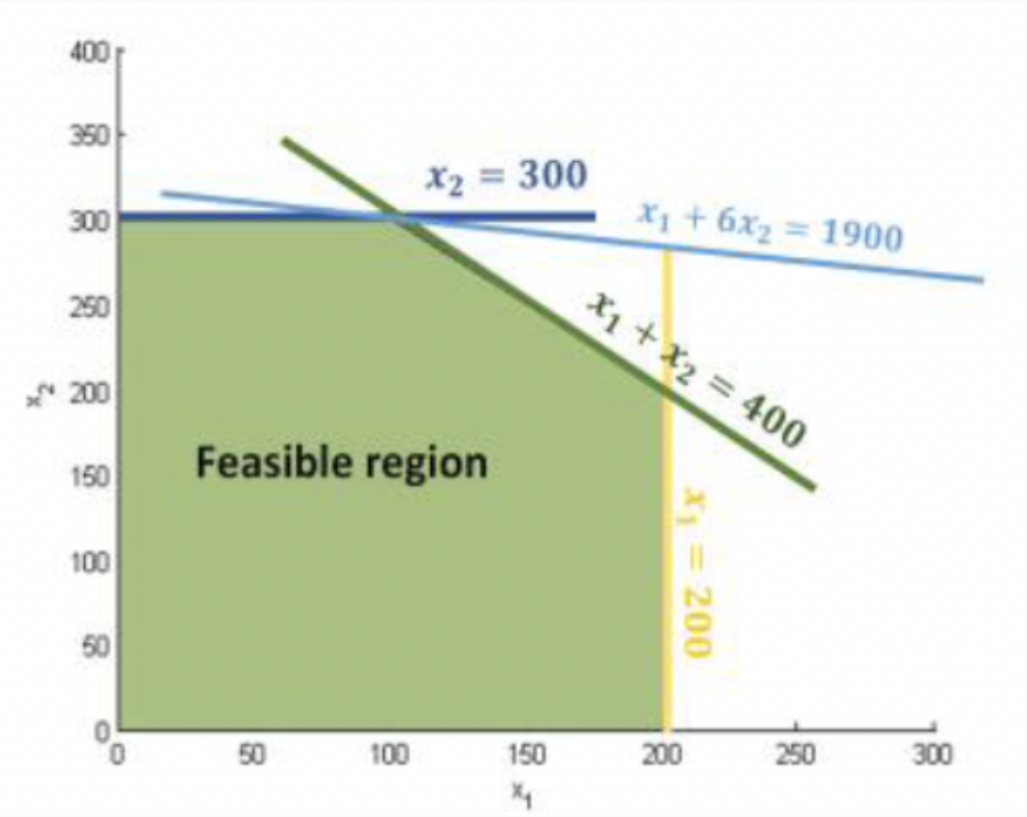


All possibilities for a linear programming

Unique optimal solution

$$\min_{x1, x_2 \in \mathbb{R}} \quad z = x_1 + 6\,x_2$$

$$\begin{aligned} \text{subject to} \quad x_1 &\leq 200, \\ x_2 &\leq 300, \\ x_1 + x_2 &\leq 400, \\ x_1, x_2 &\geq 0. \end{aligned}$$



Geometry of linear programming

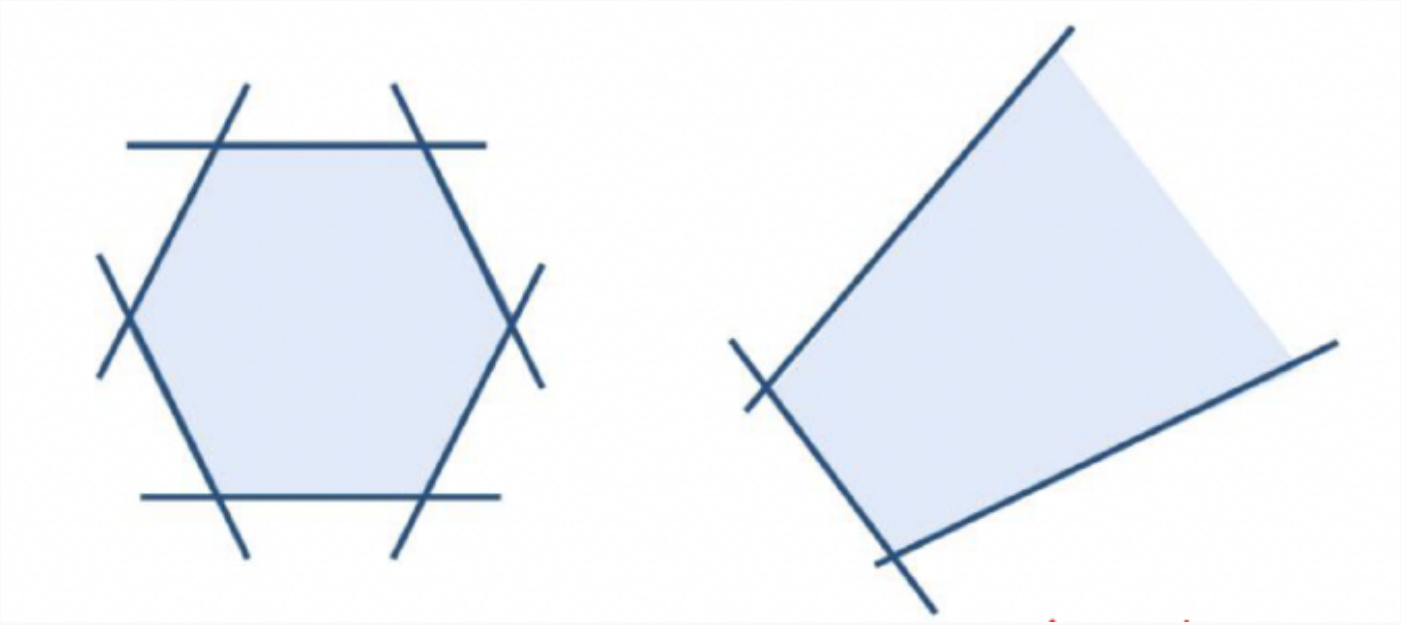
The geometry of linear programming is very beautiful. The simplex algorithm exploits this geometry in a very fundamental way. We will prove some basic geometric results here that are essential to this algorithm.

Definition

- The set $\{x \in \mathbb{R}^n \mid \langle a, x \rangle = b\}$, where $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$, is called *hyperplane*.
- The set $\{x \in \mathbb{R}^n \mid \langle a, x \rangle \geq b\}$, where $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$, is called *hyperspace*.
- The intersection of finitely many spaces is called a *polyhedron*.
    - This is always a convex set: Halfspaces are convex (why?) and intersections of convex sets are convex (why?).

Definition

- A bounded polyhedron is called a *polytope*.

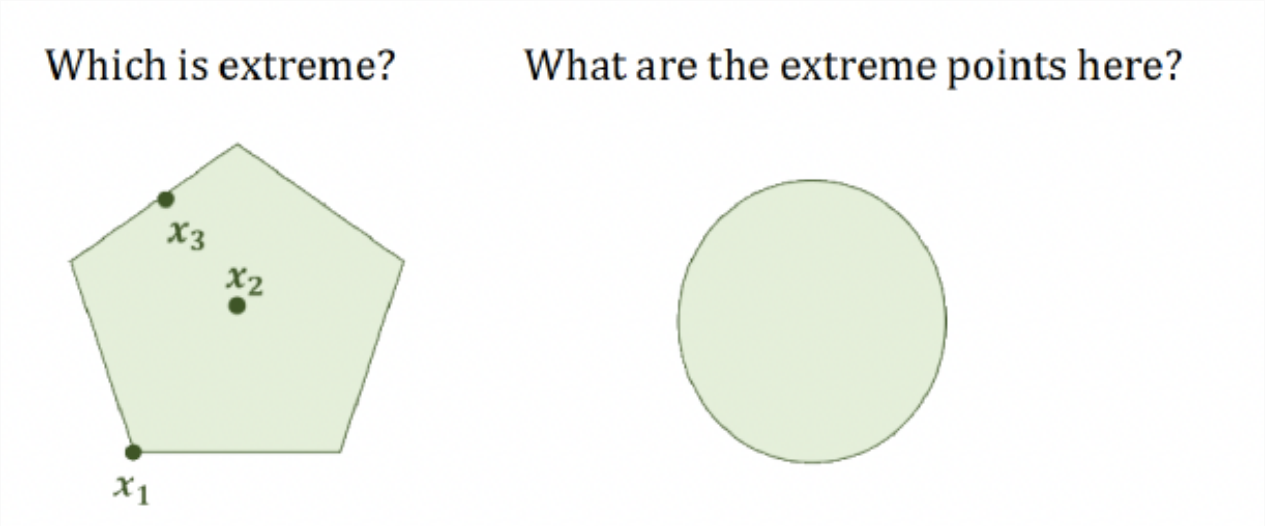


Which is polytope and which is polyhedron?

Extreme points

A point x is an extreme point of a convex set P if it cannot be written as a convex combination of two other points in P. In other words, there does not exist $y, z \in P$, $y, z \neq x$ and $\lambda \in [0, 1]$ such that $x = \lambda y + (1 - \lambda) x$.

Alternatively, $x \in P$ is an extreme point if $x = \lambda y + (1 - \lambda) z$, $y, z \in P$, $\lambda \in [0, 1]$ implies $x = y$ or $x = z$.



Extreme points are always on the boundary, but not every points on the boundary is extreme.

Linear Independence

A key idea in linear algebra is that of *linear dependence*, which is a generalization of the idea of parallel lines.

We define linear dependence of the rows of a matrix $A \in \mathbb{R}^{m \times n}$ formally as follows:

$$z^T A = 0 \qquad \text{for some nonzero} \quad z \in \mathbb{R}^m.$$

The negation of linear dependence is *linear independence* of the rows of A, which is defined by the implication

$$z^T A = 0 \qquad \Rightarrow \quad z = 0.$$

The idea of linear independence extends also to linear functions. The functions $y(x) = Ax$ are linearly independence if and only if the rows of the matrix A are linear independent.

Let us remark that $y(x)$ are linear independent if and only if

$$z^T Ax = 0 \;\; \forall x \in \mathbb{R}^n \qquad \Rightarrow \quad z = 0.$$

Definition

Consider a set of constrains

$$\begin{aligned} \langle a_i^T, x \rangle &\geq b_i, \qquad i \in M_1, \\ \langle a_i^T, x \rangle &\leq b_i, \qquad i \in M_2, \\ \langle a_i^T, x \rangle &= b_i, \qquad i \in M_3. \end{aligned}$$

Given a point $\overline{x}$, we say that a constraint i is *tight* (or active or binding) at $\overline{x}$ if $\langle a_i^T, x \rangle = b_i$.

Equality constraints are tight by definition.

Definition

Two constraints are linearly independent if the corresponding a_i's are independent.

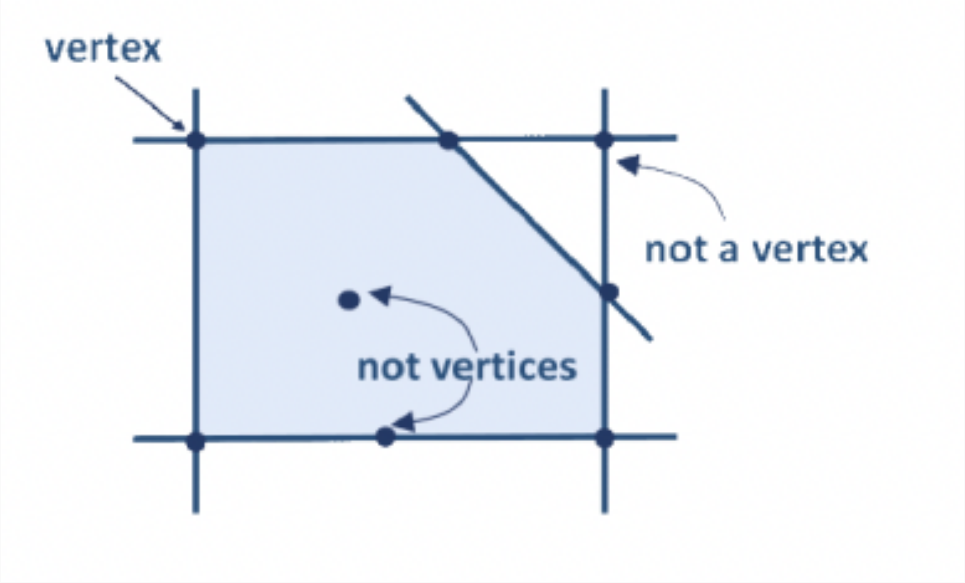
With these two definitions, we can now define the notion of a vertex of a polyhedron.

Vertex

A point $x \in \mathbb{R}^n$ is a vertex of a polyhedron P, if

i) it is feasible ($x \in P$),

ii) $\exists\, n$ linearly independent constraints that are tight at x.



Vertex vs. extreme point

- You may be wondering if extreme points and vertices are the same thing.
- Note that the notion of an extreme point is defined geometrically while the notion of a vertex is defined algebraically.
- The algebraic definition is more useful for algorithmic purpose and is crucial to the simplex algorithm. Yet, the geometric definition is used to prove the fundamental fact that an optimal solution to an linear program can always be found at a vertex. This is crucial to correctness of the simplex algorithm.

Equivalence of extreme point and vertex

Theorem

Let $P = \{x \in \mathbb{R}^n \mid Ax \geq b\}$ be a non-empty polyhedron with $A \in \mathbb{R}^{m \times n}$. Let $\overline{x} \in P$. Then, $\overline{x}$ is an extreme point if and only if $\overline{x}$ is a vertex.

Proof.

" $\Leftarrow$" Let $\overline{x} \in P$ be a vertex. This implies that n linearly independent constraints are tight at $\overline{x}$. Denote by $\widetilde{A}$ an $n \times n$ matrix whose rows are that of A associated with the tight constraints. Similarly let $\widetilde{b}$ be a vector of size n collecting entries of b corresponding to the tight constraints. So $\widetilde{A}\overline{x} = \widetilde{b}$.

Suppose, we could write $\overline{x} = \lambda y + (1 - \lambda)z$ for some $y, z \in P$ and $\lambda \in [0, 1]$, $y \neq \overline{x}$, $z \neq \overline{x}$. Then $Ay \geq b$, $Az \geq b$ implies $\widetilde{A}y \geq \widetilde{b}$ and $\widetilde{A}z \geq \widetilde{b}$.

We have

$$\widetilde{A}\overline{x} = \widetilde{b} = \lambda \widetilde{A}y + (1 - \lambda)\widetilde{A}z \geq \widetilde{b}.$$

□

Proof.

If $\lambda = 0$, then $\overline{x} = z$ as $\widetilde{A}$ is invertible. If $\lambda = 1$, then $\overline{x} = y$. If $\lambda \in (0, 1)$, then the previous equality forces $\widetilde{A}\, y = \widetilde{A}\, z = \widetilde{A}\, \overline{x}$ which means that $\overline{x} = y = z$ as $\widetilde{A}$ is invertible.

With this the "$\Leftarrow$" part of the proof is complete. $\square$

Proof.

" $\Rightarrow$" Suppose that $\overline{x} \in P$ is not a vertex. Let $I = \{i = 1, 2, ..., m \,|\, \langle a_i^T, \overline{x} \rangle = b_i\}$. Since $\overline{x}$ is not a vertex, there does not exist n linearly independent vectors a_i^T (rows of the matrix A viewed as vectors, i.e., as columns), $i \in I$.

We claim that there exists a vector $d \neq 0$ such that $\langle d, a_i^T \rangle = 0$, $\forall\, i \in I$. Indeed, take at most $k \leq (n-1)$ linearly independent a_i, $i \in I$. Let assume that their indices are $i = 1, 2, ..., k$. We want to argue that the linear system

$$\begin{aligned} \langle a_1^T, d \rangle &= 0, \\ \langle a_2^T, d \rangle &= 0, \\ &\vdots \\ \langle a_k^T, d \rangle &= 0 \end{aligned}$$

has nontrivial solution. But recall that each a_i^T is of length n. So this is an under-constrained linear system. $\square$

Proof.

Hence, it has infinitely many solutions, among which there is at least one nonzero solution, which we take to be d.

Let $y = \overline{x} + \varepsilon\, d$ and $z = \overline{x} - \varepsilon\, d$, where ε is some positive scalar. We claim that for small enough ε we have $y, z \in P$:

- for $i \in I$: $\langle a_i^T, y\rangle = \langle a_i^T, z\rangle = b_i$, because $\langle a_i^T, d\rangle = 0$.
- for $i \notin I$: the claim follows from continuity of the function $\delta \to b_i - \langle a_i^T, \overline{x} + \delta\, d\rangle$ and the fact that $b_i - \langle a_i^T, \overline{x}\rangle < 0$ when $i \notin I$.

As $\overline{x} = \frac{y}{2} + \frac{z}{2}$, this implies that $\overline{x}$ is not an extreme point of P. $\square$

Theorem

Given a finite set of linear inequalities, there can only be a finite number of extreme points.

Proof.

We have shown that extreme points and vertices are the same, so we prove the results for vertices. Suppose we are given a total of $n + m$ constraints. To obtain a vertex, we need to pick n linearly independent constrains that are tight. There are at most C^n_{m+n} ways of doing this and each subset of n linearly independent constraints gives a unique vertex (as seen previously, the vertex x satisfies $\widetilde{A}x = \widetilde{b}$ where $\widetilde{A}$ is invertible). As a consequence, the are at most C^n_{n+m} vertices. $\square$

More about vertices

For the feasible region $S := \{x \in \mathbb{R}^n \mid Ax \geq b,\ x \geq 0\}$, let

$$x_{n+i} = A_{i\cdot}x - b_i, \quad i = 1, 2, ..., m.$$

where $A_{i\cdot}$ means the ith row of the matrix A, like in Matlab.

In other word, a *vertex* of S is any point $\overline{x} = (\overline{x}_1, \overline{x}_2, ..., \overline{x}_n) \in S$ that satisfies

$$\overline{x}_N = 0,$$

where N is any subset of $\{1, 2, ..., n + m\}$ containing n elements such that the linear functions defined by $x_j, j \in N$, are linearly independent.

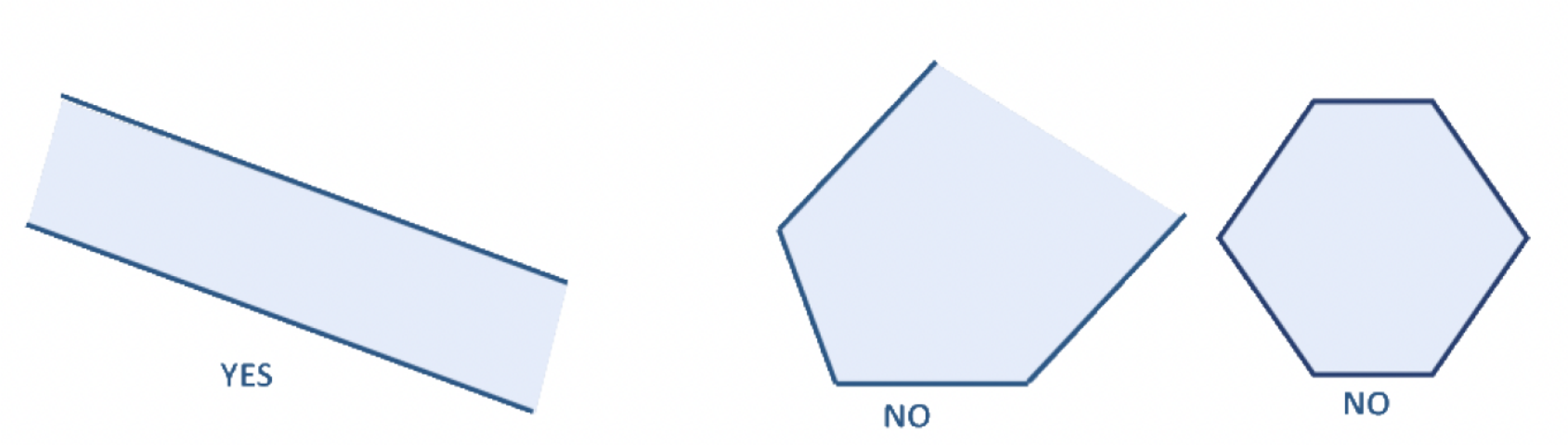
It is important for the n functions in this definition to be linearly independent. If not, then the system of equation $x_N = 0$ has either zero solution infinitely many solutions.

We know now how to characterize mathematically the vertices of the feasible region, BUT we still do not know (it is not already proven and not obvious) why to check for a solution of the linear program only among the set of vertices.

Definition

A polyhedron contains a line if there exists $x \in P$ and $d \in \mathbb{R}^n$, $d \neq 0$, such that

$$x + \lambda d \in P, \quad \forall \lambda \in \mathbb{R}.$$



Existence of extreme point

Consider a nonempty polyhedron P. The following are equivalent:

i) P does not contain a line.

ii) P has at least one extreme point.

Therefore, every bounded polyhedron has an extreme point.

Optimality of extreme points

Theorem

Consider a linear programming in the standard form, i.e.,

$$\min_{x \in \mathbb{R}^n} \quad z = \langle p, x \rangle$$
$$\text{subject to} \quad Ax \geq b \quad x \geq 0,$$

where $p \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, and $A \in \mathbb{R}^{m \times n}$.

Suppose P has at least one extreme point and there exists an optimal solution, then there exists an optimal solution which is at a vertex.

Proof.

Let Q be the set of optimal solutions (assumed to be nonempty). In other words, if v is the optimal value of the linear program, then $Q := \{x \mid A\,x \geq b, z = \langle p, x \rangle = v\}$.

Using the above theorem we know that if P has an extreme point then P has not lines. Since $Q \subset P$, it follows that Q has no lines. Hence Q has an extreme point.

Let x^* be an extreme point of Q. We will show that x^* is also extreme point of P. Once this is prove, since $\langle p, x^* \rangle = v$, we would be done. $\square$

Proof.

Suppose that x^* is not an extreme point of P. Then $\exists\, y \neq x^*$, $z \neq x^*$, $\lambda \in [0,1]$, such that

$$x^* = \lambda\, y + (1-\lambda)\, z.$$

Multiplying by p on both sides, we obtain

$$v = \langle p, x^* \rangle = \lambda \langle p, y \rangle + (1-\lambda)\langle p, z \rangle.$$

Since v is optimal, $\langle p, y \rangle \geq v$ and $\langle p, z \rangle \geq v$. Combined with the previous equality, this implies

$$\langle p, y \rangle = v, \langle p, z \rangle = v.$$

But this means that $y \in Q$ and $z \in Q$. Hence, that x^* is not an extreme point of Q. Contradiction! $\square$

Implication of the theorems

- These theorems show that when looking for an optimal solution, it is enough to examine only the extreme points (which is equivalent to vertices).
- This leads to an algorithm for solving an linear program: if there are m constraints in $\mathbb{R}^n$, then pick all possible subsets of n linearly independent constraints out of the $n+m$. Solve (in worst case) C_{n+m}^n systems of equations of the type $\widetilde{A}x = \widetilde{b}$, where $\widetilde{A}$, $\widetilde{b}$ are the restrictions of A and b to the subset of n constraints. This can be done, e.g., by the gradient method from the previous lectures or by Gaussian elimination. Evaluate the objective function at each solution and pick the best.
- Unfortunately, this algorithm, even though correct, is very inefficient. The reason is that there are too many vertices to explore. For example, consider the constraints $\{-1 \leq x \leq 1\}$, $i = 1, 2, ..., n$. Then we have in general $2\,n$ inequalities, but 2^n extreme points.

- The simplex method (which will be done in the following lectures) is an intelligent algorithm for reducing the number of vertices visited.

Its name is the **simplex algorithm** and in a nutshell, this is all the simplex algorithm does:

- start at a vertex,
- while there is a better neighboring vertex, move to it.

Definition

Two vertices are neighbors if they share $n - 1$ tight constraints.

How to move from vertex to vertex whithout knowing their possitions?

Back to our Simple Example

Linear programs in standard form

$$
\begin{aligned}
\min_{x_1, x_2 \in \mathbb{R}} \quad & z = 3\,x_1 - 6\,x_2 \\
\text{subject to} \quad & x_1 + 2x_2 \geq -1, \\
& 2\,x_1 + x_2 \geq 0, \\
& x_1 - x_2 \geq -1, \\
& x_1 - 4\,x_2 \geq -13, \\
& -4\,x_1 + x_2 \geq -23, \\
& x_1, x_2 \geq 0.
\end{aligned}
$$

Linear programs in canonical form

The first step is to add slack variables to convert the constraints into a set of general equalities combined with nonnegativity requirements on all the variables. The slacks are defined as follows:

$$
\begin{aligned}
x_3 &= x_1 + 2x_2 - 1, \\
x_4 &= 2\,x_1 + x_2, \\
x_5 &= x_1 - x_2 + 1, \\
x_6 &= x_1 - 4\,x_2 + 13, \\
x_7 &= -4\,x_1 + x_2 + 23,
\end{aligned}
$$

Tableau representation of the linear program

		x_1	x_2	1
x_3	$=$	1	2	1
x_4	$=$	2	1	0
x_5	$=$	1	-1	1
x_6	$=$	1	-4	13
x_7	$=$	-4	1	23
z	$=$	3	-6	0

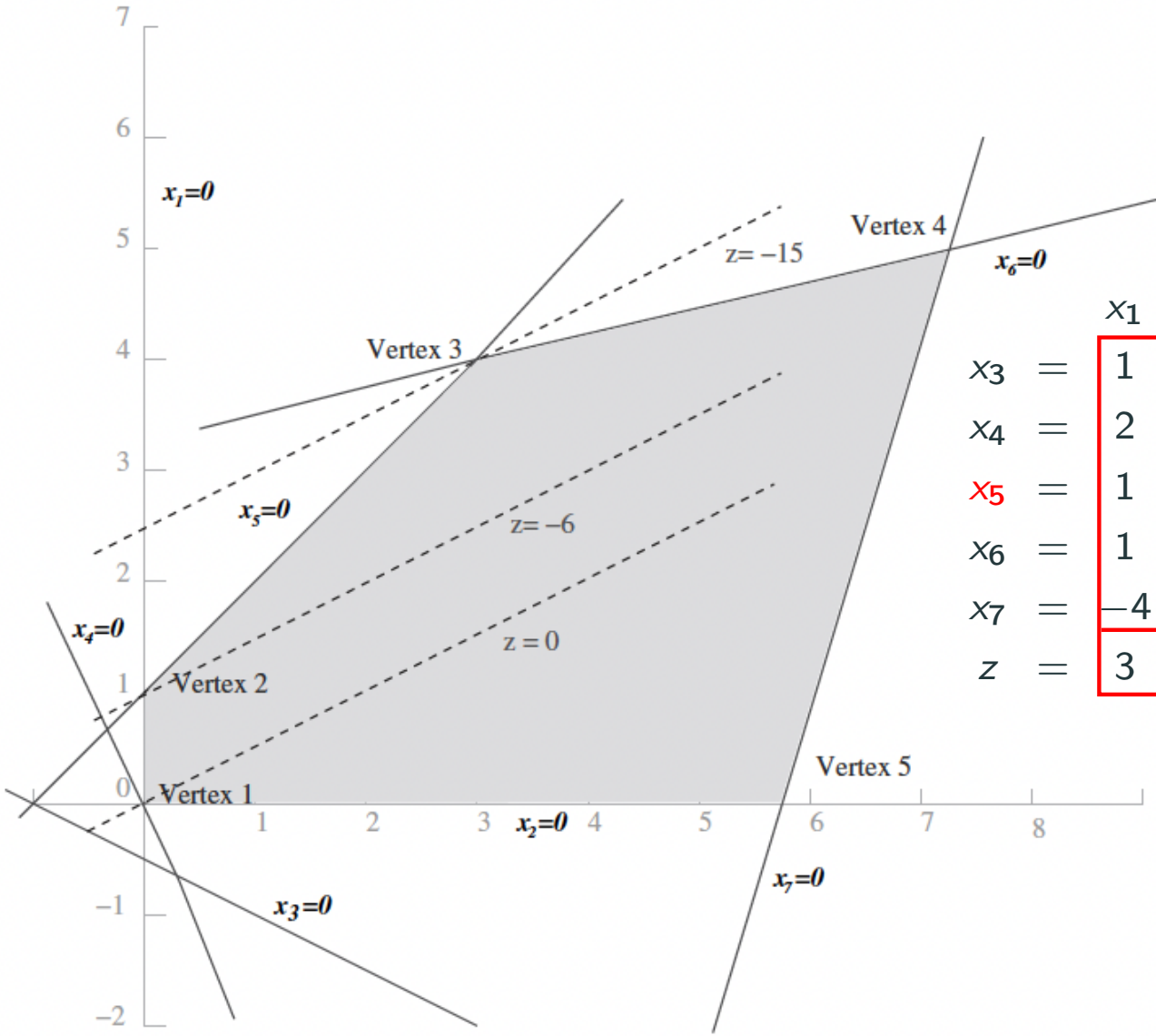
In MATLAB we store this tableau in a structure.

We may set a starting searching point by setting (the nonbasic variables) $x_1 = 0$ and $x_2 = 0$, since the table is feasible, i.e. the corresponding slack variables (basic) remain positive.

The values of the objective in this tableau, $z = 0$, is obtained from the bottom right element.

Let us move from this starting position! I know now why but we have to understand how we could move from a vertex to another vertex?

Tableau representation of the Vertex 1



		x_1	x_2	1
x_3	=	1	2	1
x_4	=	2	1	0
x_5	=	1	−1	1
x_6	=	1	−4	13
x_7	=	−4	1	23
z	=	3	−6	0

How to move in Vertex 2 and stock it in Tableau, i.e., in a structure?

Jordan Exchange

Consider the following simple linear equation in the one-dimensional variables x and y:

$$y = a\,x.$$

The form of the equation indicates that x is the *independent variable* and y is the *dependent variable*: Given a value of x, the equation tells us how to determine the corresponding values of y, i.e.

$$y(x) := a\,x.$$

If we assume that $a \neq 0$, we can reverse the roles of y and x as follows:

$$x = a^{-1}\,y.$$

Jordan Exchange

The Jordan exchange is a generalization of the process above. It deals with the case in which $x \in \mathbb{R}^n$ is a vector of independent variables and $y \in \mathbb{R}^m$ is a vector of dependent variables, and we wish to exchange one of the independent variables with one of the dependent variables.

Let us consider a linear system $y = A\,x$, where $A \in \mathbb{R}^{m \times n}$, and change the roles of a component y_r of y and a component x_s of x. We write this system equation-wise as

$$y_i = A_{i1}\,x_1 + A_{i2}\,x_2 + ... + A_{in}\,x_n, \quad i = 1, 2, ..., m,$$

where the *independent variables* are $x_1, x_2, ..., x_n$ and the *dependent variables* are $y_1, y_2, ..., y_m$.

Jordan Exchange

We can think of the y_i's as linear functions of x_j's, that is

$$y_i(x) = A_{i1}\,x_1 + A_{i2}\,x_2 + \ldots + A_{in}\,x_n, \quad i = 1, 2, \ldots, m$$

or, more succinctly, $y(x) := A\,x$. This system can also be represented in the following *tableau* form:

$$\begin{array}{cc|ccccc} & & x_1 & \cdots & x_s & \cdots & x_n \\ y_1 & = & A_{11} & \cdots & A_{1s} & \cdots & A_{1n} \\ \vdots & & \vdots & \cdots & \vdots & \cdots & \vdots \\ y_r & = & A_{r1} & \cdots & A_{rs} & \cdots & A_{rn} \\ \vdots & & \vdots & \cdots & \vdots & \cdots & \vdots \\ y_m & = & A_{m1} & \cdots & A_{ms} & \cdots & A_{mn} \end{array}$$

We now decribe the *Jordan exchange* or *pivot operation* with regard to the tableau representation. The dependent variable y_r become independent, while x_r changes from being independent to being dependent.

$$
\begin{array}{cc|cccccc}
 & & x_1 & \cdots & x_s & \cdots & x_n \\
y_1 & = & A_{11} & \cdots & A_{1s} & \cdots & A_{1n} \\
\vdots & & \vdots & \cdots & \vdots & \cdots & \vdots \\
y_r & = & A_{r1} & \cdots & A_{rs} & \cdots & A_{rn} \\
\vdots & & \vdots & \cdots & \vdots & \cdots & \vdots \\
y_m & = & A_{m1} & \cdots & A_{ms} & \cdots & A_{mn}
\end{array}
$$

The process is carried out by the following three steps:

a) Solve the rth equation $y_r = A_{r1}x_1 + ... + A_{rs}x_s + ... + A_{rn}x_n$ for x_s in terms of $x_1, ..., x_{s-1}, y_r, x_{s+1}, ..., x_n$. Note that this is possible if and only if $A_{rs} \neq 0$. (A_{rs} is known as *pivot element*.)

b) Substitute for x_s in all the remaining equations.

c) Write the linear dependency in a new tableau as follows

$$\begin{array}{ccc} & x_1 \ \cdots \ x_s \ \cdots \ x_n \\ y_1 & A_{11} \ \cdots \ A_{1s} \ \cdots \ A_{1n} \\ \vdots & \vdots \ \cdots \ \vdots \ \cdots \ \vdots \\ y_r & A_{r1} \ \cdots \ A_{rs} \ \cdots \ A_{rn} \\ \vdots & \vdots \ \cdots \ \vdots \ \cdots \ \vdots \\ y_m & A_{m1} \ \cdots \ A_{ms} \ \cdots \ A_{mn} \end{array} \qquad \begin{array}{ccc} & x_1 \ \cdots \ y_r \ \cdots \ x_n \\ y_1 & B_{11} \ \cdots \ B_{1s} \ \cdots \ B_{1n} \\ \vdots & \vdots \ \cdots \ \vdots \ \cdots \ \vdots \\ x_s & B_{r1} \ \cdots \ B_{rs} \ \cdots \ B_{rn} \\ \vdots & \vdots \ \cdots \ \vdots \ \cdots \ \vdots \\ y_m & B_{m1} \ \cdots \ B_{ms} \ \cdots \ B_{mn} \end{array}$$

To determine the elements B_{ij} in terms of the elemets A_{ij}, let us carry out the algebra specified by the Jordan exchange.

Solution of the rth equation $y_r = A_{r1}x_1 + ... + A_{rs}x_s + ... + A_{rn}x_n$ for x_s gives

$$x_s = \underbrace{\frac{1}{A_{rs}}}_{:=Brs} y_r + \sum_{j=1, j\neq s}^{n} \underbrace{\frac{-A_{rj}}{A_{rs}}}_{:=Brj} x_j.$$

Subtituting of the expression of x_s, i.e.

$$x_s = \underbrace{\frac{1}{A_{rs}}}_{:=Brs} y_r + \sum_{j=1, j\neq s}^{n} \underbrace{\frac{-A_{rj}}{A_{rs}}}_{:=Brj} x_j$$

in the ith equation of the tableau ($i \neq r$), i.e. in
$y_i = A_{i1}x_1 + ... + A_{is}x_s + ... + A_{in}x_n$,

we have the formulae defining the rows $i = 1, 2, ...m$, $i \neq r$:

$$\begin{aligned} y_i &= \sum_{j=1, j\neq s}^{n} A_{ij}\, x_j + A_{is} \left(\frac{1}{A_{rs}} y_r + \sum_{j=1, j\neq s}^{n} \frac{-A_{rj}}{A_{rs}} x_j \right) \\ &= \sum_{j=1, j\neq s}^{n} \underbrace{\left(A_{ij} - \frac{A_{is}A_{rj}}{A_{rs}} \right)}_{:=B_{ij}} x_j + \underbrace{\frac{A_{is}}{A_{rs}}}_{:=B_{is}} y_r, \qquad \forall i \neq r. \end{aligned}$$

We may do multiple pivots in succession. Consider the linear function y defined by $y(x) = Ax$, where $A \in \mathbb{R}^{m \times n}$. After k pivots (with appropriate reordering of rows and columns) denote the initial and kth tableaus as follows:

$$\begin{array}{rcc} & x_{J_1} & x_{J_2} \\ y_{I_1} = & A_{I_1 J_1} & A_{I_1 J_2} \\ y_{I_2} = & A_{I_2 J_1} & A_{I_2 J_2} \end{array} \qquad \begin{array}{rcc} & y_{I_1} & x_{J_2} \\ x_{J_1} = & B_{I_1 J_1} & B_{I_1 J_2} \\ y_{I_2} = & B_{I_2 J_1} & B_{I_2 J_2} \end{array}$$

Here I_1, I_2 is a partition of $\{1, 2, ..., m\}$ and J_1, J_2 is a partition of $\{1, 2, ..., n\}$, with I_1 and J_1 containing the same number of elements.

Both tableaus express the same identity in different ways.

The uniqueness

If the linear function y is defined by $y(x) = Ax$ and also by $y(x) = Bx$, then $A = B$.

Proof.

Since $(A - B)x = 0$ for all $x \in \mathbb{R}^n$, take x the elements of the canonical base in $\mathbb{R}^n$. We conclude that $A = B$. □

Linear Independence

A key idea in linear algebra is that of *linear dependence*, which is a generalization of the idea of parallel lines.

We define linear dependence of the rows of a matrix A formally as follows:

$$z^T A = 0 \qquad \text{for some nonzero} \quad z \in \mathbb{R}^m.$$

The negation of linear dependence is *linear independence* of the rows of A, which is defined by the implication

$$z^T A = 0 \qquad \Rightarrow \qquad z = 0.$$

Linear Independence

The idea of linear independence extends also to linear functions. The functions $y(x) = Ax$ are linearly independence if and only if the rows of the matrix A are linear independent.

Let us remark that $y(x)$ are linear independent if and only if

$$z^T Ax = 0 \quad \forall x \in \mathbb{R}^n \qquad \Rightarrow \qquad z = 0.$$

Remark

If the m linear functions y_i are linearly independent, then any p of them are also linearly independent, where $p \leq m$.

Proposition

If the linear functions y defined by $y(x) = Ax$, $A \in \mathbb{R}^{m \times n}$, are linearly independent, then $m \leq n$. Furthermore, in the tableau representation, all m dependent y_i's can be made independent; that is, they can be exchanged with m independent x_j's.

Proof.

Suppose that the linear functions $y(x) = Ax$ are linearly independent. Exchange y's and x's in the tableau until no further pivots are possible, at which point we are blocked by a tableau of the following form (after a possible rearrangement of rows and columns):

$$\begin{array}{rcc} & x_{J_1} & x_{J_2} \\ y_{I_1} = & A_{I_1 J_1} & A_{I_1 J_2} \\ y_{I_2} = & A_{I_2 J_1} & A_{I_2 J_2} \end{array} \qquad \begin{array}{rcc} & y_{I_1} & x_{J_2} \\ x_{J_1} = & B_{I_1 J_1} & B_{I_1 J_2} \\ y_{I_2} = & B_{I_2 J_1} & 0 \end{array}$$

If $I_2 \neq \emptyset$, we have that the tableau says that

$$y_{I_2}(x) = B_{I_2 J_1} y_{I_1}(x) \quad \Rightarrow \quad [] B_{I_2 J_1} \quad I] \begin{pmatrix} y_{I_1}(x) \\ y_{I_2}(x) \end{pmatrix} = 0.$$

Note that any row z of $[-B_{I_2 J_1} \quad I]$ is nonzero and the above relation implies that the functions $y(x) = Ax$ are linear dependent.

Hence, we must have $I_2 = \emptyset$, and therefore $m \leq n$ and all the y_i's have been pivoted to the top of the tableau, as required. $\square$

Proposition (Steinitz)

For a given matrix $A \in \mathbb{R}^{m \times n}$, the linear functions y defined by $y(x) = Ax$ are linearly independent if and only if for the corresponding tableau representation all y_i's can be exchanged with m independent x_j's.

Proof The "only if" part follows from the previous Proposition. So, we have to prove only the "if" part.

If all the y_i's can be exchanged to the top of the tableau, then we have (by rearranging rows and columns if necessary) that

$$\begin{array}{c} \\ y \\ \\ \end{array} = \begin{array}{c} \begin{array}{cc} x_{J_1} & x_{J_2} \end{array} \\ \boxed{\begin{array}{cc} A_{\cdot J_1} & A_{\cdot J_2} \end{array}} \end{array} \qquad \begin{array}{c} \\ x_{J_1} \\ \\ \end{array} = \begin{array}{c} \begin{array}{cc} y_{I_1} & x_{J_2} \end{array} \\ \boxed{\begin{array}{cc} B_{\cdot J_1} & B_{\cdot J_2} \end{array}} \end{array}$$

$$\begin{array}{c} \\ y \end{array} \begin{array}{c} \begin{array}{cc} x_{J_1} & x_{J_2} \end{array} \\ \boxed{\begin{array}{cc} A_{.J_1} & A_{.J_2} \end{array}} \end{array} \qquad\qquad \begin{array}{c} \\ x_{J_1} \end{array} \begin{array}{c} \begin{array}{cc} y_{I_1} & x_{J_2} \end{array} \\ \boxed{\begin{array}{cc} B_{.J_1} & B_{.J_2} \end{array}} \end{array}$$

Suppose now that there is some z such that $z^T A = 0$.

We therefore have that $z^T A x = 0$ for all $x \in \mathbb{R}^n$. In the right-hand side tableau above, we may set the independent variables $y = z$, $x_{J_2} = 0$, whereupon $x_{J_1} = B_{.J_1} z$.

For this particular choice of x and y, we have actually $y = Ax$, and so it follows that

$$z^T A x = 0 \;\Rightarrow\; z^T y = 0 \;\Rightarrow\; z^T z = 0 \;\Rightarrow\; z = 0,$$

verifying that the rows of A are linearly independent.

Again the definition of a vertex

For the feasible region $S := \{x \in \mathbb{R}^n \mid A\,x \geq b,\ x \geq 0\}$, let

$$x_{n+i} = A_{i\cdot}x - b_i, \quad i = 1, 2, ..., m.$$

where $A_{i\cdot}$ means the ith row of the matrix A, like in Matlab.

In other word, a *vertex* of S is any point $(x_1, x_2, ..., x_n) \in S$ that satisfies

$$x_N = 0,$$

where N is any subset of $\{1, 2, ..., n + m\}$ containing n elements such that the linear functions defined by $x_j, j \in N$, are linearly independent.

It is important for the n functions in this definition to be linearly independent. If not, then the system of equation $x_N = 0$ has either zero solution infinitely many solutions.

More about vertices

Let us consider a linear program for which $(0, 0, ..., 0) \in \mathbb{R}^n$ belongs to the feasible region. Therefore, taking $N = \{1, 2, ...n\}$, the following table

$$\begin{array}{rcc} & x_N & 1 \\ x_B & A & -b \\ z & p^T & 0 \end{array}$$

describes the vertex $(0, 0, ..., 0)$. And we see directly form this tableau that $(0, 0, ..., 0)$ belongs to the feasible region only if $-b$ has only nonnegative values.

- Starting from one vertex of the domain of feasibility, how it is possible to arrive to the tableau describing another vertex?
  - An idea is to use Jordan exchanges.

More about vertices

Let us consider a linear program for which $(0, 0, ..., 0) \in \mathbb{R}^n$ belongs to the feasible region. Therefore, taking $N = \{1, 2, ...n\}$, the following table

$$\begin{array}{rcc} & x_N & 1 \\ x_B & A & -b \\ z & c^T & 0 \end{array}$$

describes the vertex $(0, 0, ..., 0)$. And we see directly form this tableau that $(0, 0, ..., 0)$ belongs to the feasible region only if $-b$ has only nonnegative values.

- Do we succeed to come to another tableau describing an arbitrary vertex proceeding a finite number of Jordan exchanges?
  - Perhaps, yes, due to the definition of a vertex and as a consequence of the Steiniz's Theorem. Explain!

More about vertices

Let us consider a linear program for which $(0, 0, ..., 0) \in \mathbb{R}^n$ belongs to the feasible region. Therefore, taking $N = \{1, 2, ...n\}$, the following table

$$\begin{array}{rcc} & x_N & 1 \\ x_B & A & -b \\ z & p^T & 0 \end{array}$$

describes the vertex $(0, 0, ..., 0)$. And we see directly form this tableau that $(0, 0, ..., 0)$ belongs to the feasible region only if $-b$ has only nonnegative values.

- By proceeding a Jordan exchange to a feasible table, do we arrive to a feasible tableau?

That means to a tableau of the form

$$\begin{array}{rcc} & x_N & 1 \\ x_B & H & h \\ z & p^T & \alpha \end{array}$$

where N is now another subset of indices of $\{1, 2, ..., n + m\}$ and $h \geq 0$.

A vertex means a feasible tableau.

Theorem

Suppose that $\overline{x}$ is a vertex of S with corresponding index set N. Then if we define

$$\mathcal{A} := [A \ -I], \quad B := \{1, 2, ..., n+m\} \setminus N,$$

then $\overline{x}$ satisfies the relationships

$$\mathcal{A}_{.B} x_B + \mathcal{A}_{.N} x_N = b, \quad x_B \geq 0, \quad x_N = 0,$$

where $\mathcal{A}_{.B}$ is invertible. Moreover, $\overline{x}$ can represented by a tableau of the form

$$\begin{array}{rcc} & x_N & 1 \\ x_B & = & H \quad h \\ z & = & c^T \quad \alpha \end{array}$$

with $h \geq 0$, i.e., by a feasible one.

Proof

By the definition of a vertex it follows that

$$\mathcal{A}_{.B}\overline{x}_B + \mathcal{A}_{.N}\overline{x}_N = b, \quad \overline{x}_B \geq 0, \quad \overline{x}_N = 0.$$

It remains to prove that $\mathcal{A}_{.B}$ is invertible. Suppose there exists a vector z such that $z^T\mathcal{A}_{.B} = 0$. It follows from above that $z^T b = 0$.

By the definition, the functions x_N satisfy

$$\mathcal{A}_{.B}x_B + \mathcal{A}_{.N}x_N = b,$$

and so $z^T\mathcal{A}_{.N}x_n = 0$.

Since x_N are linearly independent, it follows that $z^T\mathcal{A}_{.N} = 0$. Together with the assumption that $z^T\mathcal{A}_{.B} = 0$, this implies that $z^T\mathcal{A} = 0$.

Since $\mathcal{A}$ has the (negative) identity matrix in its columns, this implies that $z = 0$, and thus $\mathcal{A}_{.B}$ is invertible.

Proof

Finally, premultiplying

$$\mathcal{A}_{.B} x_B + \mathcal{A}_{.N} x_N = b,$$

by $\mathcal{A}_{.B}$ and rearranging, we see that

$$x_B = \underbrace{\mathcal{A}_{.B}^{-1} \mathcal{A}_{.N}}_{:=H} x_N + \underbrace{\mathcal{A}_{.B}^{-1} b}_{:=h},$$

which can be viewed in the first m lines of the tableau from conclusion of the theorem.

Regarding the last line of the tableau, let us remark that the initial expression of the cost, i.e.,
$z = p_1 x_1 + p_2 x_2 + ... + p_n x_n + 0 \cdot x_{n+1} + 0 \cdot x_{n+m}$, may be written generically as

$$z = p_B^T x_B + p_N^T x_N.$$

By substituting

$$x_B = \underbrace{\mathcal{A}_{.B}^{-1}\mathcal{A}_{.N}}_{:=H} x_N + \underbrace{\mathcal{A}_{.B}^{-1} b}_{:=h}$$

in this last form of the cost we obtain

$$z = p_B^T x_B + p_N^T x_N = (-p_B^T \mathcal{A}_{.B}^{-1}\mathcal{A}_{.N} + p_N^T)x_N + p_B^T \mathcal{A}_{.B}^{-1} b,$$

i.e.,

$$c^T = -p_B^T \mathcal{A}_{.B}^{-1}\mathcal{A}_{.N} + p_N^T$$

and

$$\alpha = p_B^T \mathcal{A}_{.B}^{-1} b.$$

Note that $h \geq 0$ since $\overline{x}$ is a point of the feasible region, i.e., $\overline{x} > 0$, but it also satisfies

$$x_B = \underbrace{\mathcal{A}_{.B}^{-1}\mathcal{A}_{.N}}_{:=H} x_N + \underbrace{\mathcal{A}_{.B}^{-1} b}_{:=h}$$

and $\overline{x}_N = 0$. Therefore, $h = \overline{x}_B \geq 0$.

In which vertex it is optimal to move?

General rule

We move from one vertex to another vertex only if the COST decrease!

Hence, only if we obtain something better.

How to move optimal?

Tableau representation of the linear program

		x_1	x_2	1
x_3	$=$	1	2	1
x_4	$=$	2	1	0
x_5	$=$	1	-1	1
x_6	$=$	1	-4	13
x_7	$=$	-4	1	23
z	$=$	3	-6	0

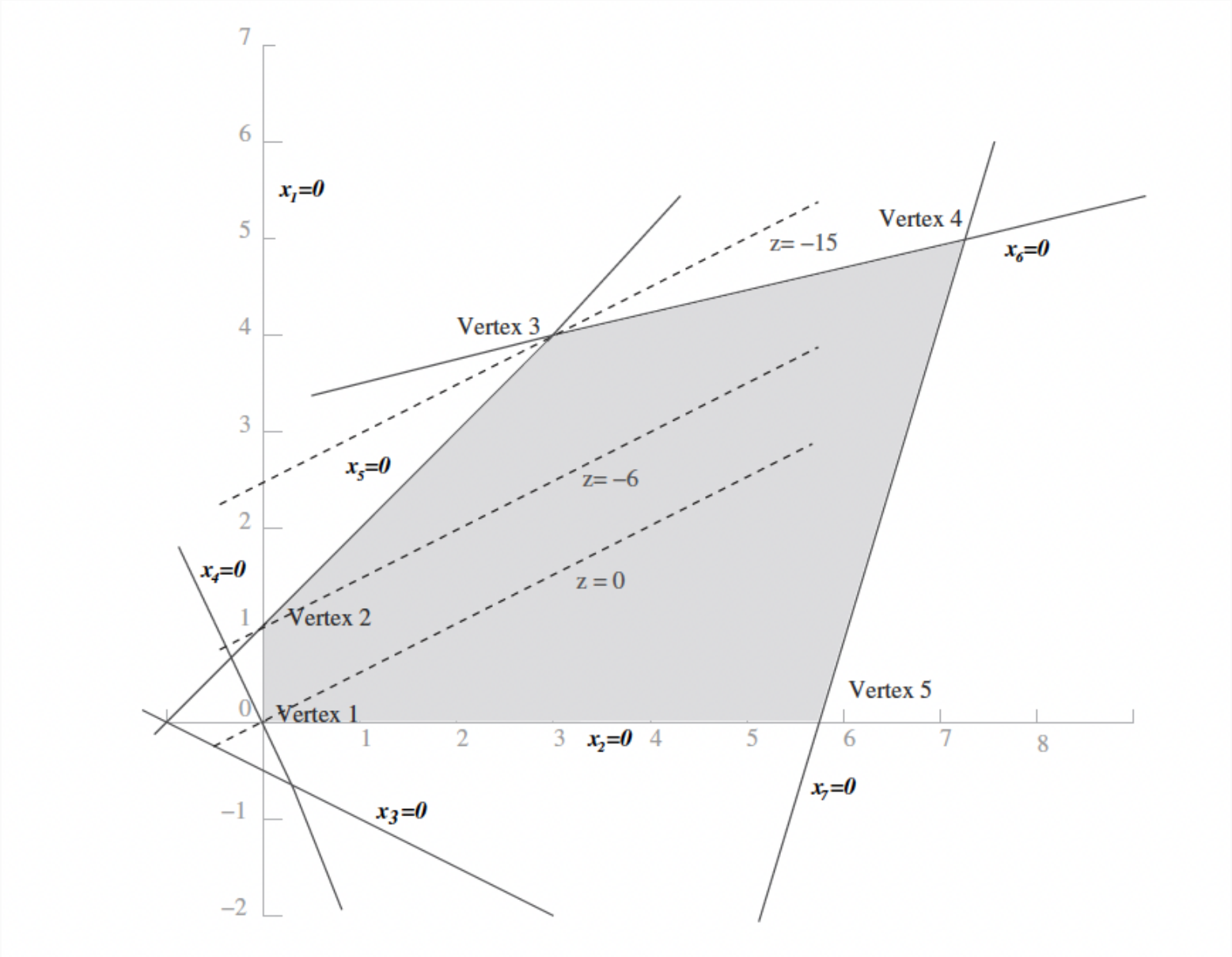
In MATLAB we store this tableau in a structure.

We may set a starting searching point by setting (the nonbasic variables) $x_1 = 0$ and $x_2 = 0$, since the table is feasible, i.e. the corresponding slack variables (basic) remain positive.

The values of the objective in this tableau, $z = 0$, is obtained from the bottom right element.

Let us move from this starting position! I know now why but we have to

How to move optimal?



How to move optimal?

Let us move from this starting position! Where is optimal?

We now seek a pivot - a Jordan exchange of a basic variable with a nonbasic variable - that yields a decrease in the objective z.

The issue is to choose the nonbasic variable which is to become basic, that is, to choose a pivot column in the tableau. In allowing a nonbasic variable to become basic, we are allowing its value to possibly increase from 0 to some positive value. After that, we study which effect will this increase have on z and on the dependence (basic) variable.

In the given example, let us try increasing x_1 from 0. We assign x_1 the nonnegative value λ while holding the other nonbasic variable x_2 at zero; that is

$$x_1 = \lambda, \qquad x_2 = 0.$$

How to move optimal?

The tableau

		x_1	x_2	1
x_3	$=$	1	2	1
x_4	$=$	2	1	0
x_5	$=$	1	-1	1
x_6	$=$	1	-4	13
x_7	$=$	-4	1	23
z	$=$	3	-6	0

tells us how the objective z depends on x_1 and x_2, and so for the values given above we have

$$z = 3\lambda - 6 \cdot 0 = 3\lambda > 0 \qquad \text{for} \qquad \lambda > 0.$$

This expression tells us that z increases as λ increases- the opposite of what we want!

How to move optimal?

Let us try instead choosing x_2 as the variable to increase, and set

$$x_1 = 0, \qquad x_2 = \lambda > 0.$$

The tableau

		x_1	x_2	1
x_3	$=$	1	2	1
x_4	$=$	2	1	0
x_5	$=$	1	-1	1
x_6	$=$	1	-4	13
x_7	$=$	-4	1	23
z	$=$	3	-6	0

tells us how the objective z depends on x_1 and x_2, and so for the values given above we have

$$z = 3 \cdot 0 - 6\,\lambda = -6\,\lambda < 0 \qquad \text{for} \qquad \lambda > 0,$$

thus decreasing z, as we wished.

How to move optimal?

The tableau

		x_1	x_2	1
x_3	$=$	1	2	1
x_4	$=$	2	1	0
x_5	$=$	1	-1	1
x_6	$=$	1	-4	13
x_7	$=$	-4	1	23
z	$=$	3	-6	0

tells us how the objective z depends on x_1 and x_2, and so for the values given above we have

$$z = 3 \cdot 0 - 6\,\lambda = -6\,\lambda < 0 \qquad \text{for} \qquad \lambda > 0,$$

thus decreasing z, as we wished.

The general rule is to choose the pivot column to have a *negative* value in the last row, as this indicates that z will decreases as the variable corresponding to that column increases away from 0.

How to move optimal?

We use the term *pricing* to indicate selection of the pivot column.

We call the label of the pivot column the *entering variable*, as this variable is the one that "enters" the basis as this step of the simplex method.

To determine which of the basis variables is to change places with the entering variable, we examine the effect of increasing the entering variable on each of the basis variables.

The tableau

		x_1	x_2	1
x_3	$=$	1	2	1
x_4	$=$	2	1	0
x_5	$=$	1	-1	1
x_6	$=$	1	-4	13
x_7	$=$	-4	1	23
z	$=$	3	-6	0

indicates the following dependences of the basic variables in the non-basic variables

$$\begin{aligned} x_3 &= 2\,\lambda + 1, \\ x_4 &= \lambda, \\ x_5 &= -\lambda + 1, \\ x_6 &= -4\,\lambda + 13, \\ x_7 &= \lambda + 23. \end{aligned}$$

Since $z = -6\lambda$, we clearly would like to make λ as large as possible to obtain the largest possible decrease in z. On the other hand, we cannot allow λ to become too large, as this would force some of the basic variables to become negative. By enforcing the nonnegativity restrictions on the variables above, we obtain the following restrictions on the value of λ:

$$\begin{aligned}
x_3 &= \quad 2\lambda + 1 \geq 0 \quad \Rightarrow \quad \lambda \geq -1/2 \\
x_4 &= \quad \lambda \geq 0 \quad \Rightarrow \quad \lambda \geq 0 \\
x_5 &= -\lambda + 1 \geq 0 \quad \Rightarrow \quad \lambda \leq 1 \\
x_6 &= -4\lambda + 13 \geq 0 \quad \Rightarrow \quad \lambda \leq 13/14 \\
x_7 &= \quad \lambda + 23 \geq 0 \quad \Rightarrow \quad \lambda \geq -23.
\end{aligned}$$

We see that the largest nonnegative value that λ can take without violating any of these constraints is $\lambda = 1$. Moreover, we observe that the blocking variable-the one that will become negative if we increase λ above its limit of 1-is x_5.

We choose the row for which x_5 is the label as the pivot row and refer to x_5 as the leaving variable—the one that changes from being basic to being nonbasic. The pivot row selection process just outlined is called the ratio test.

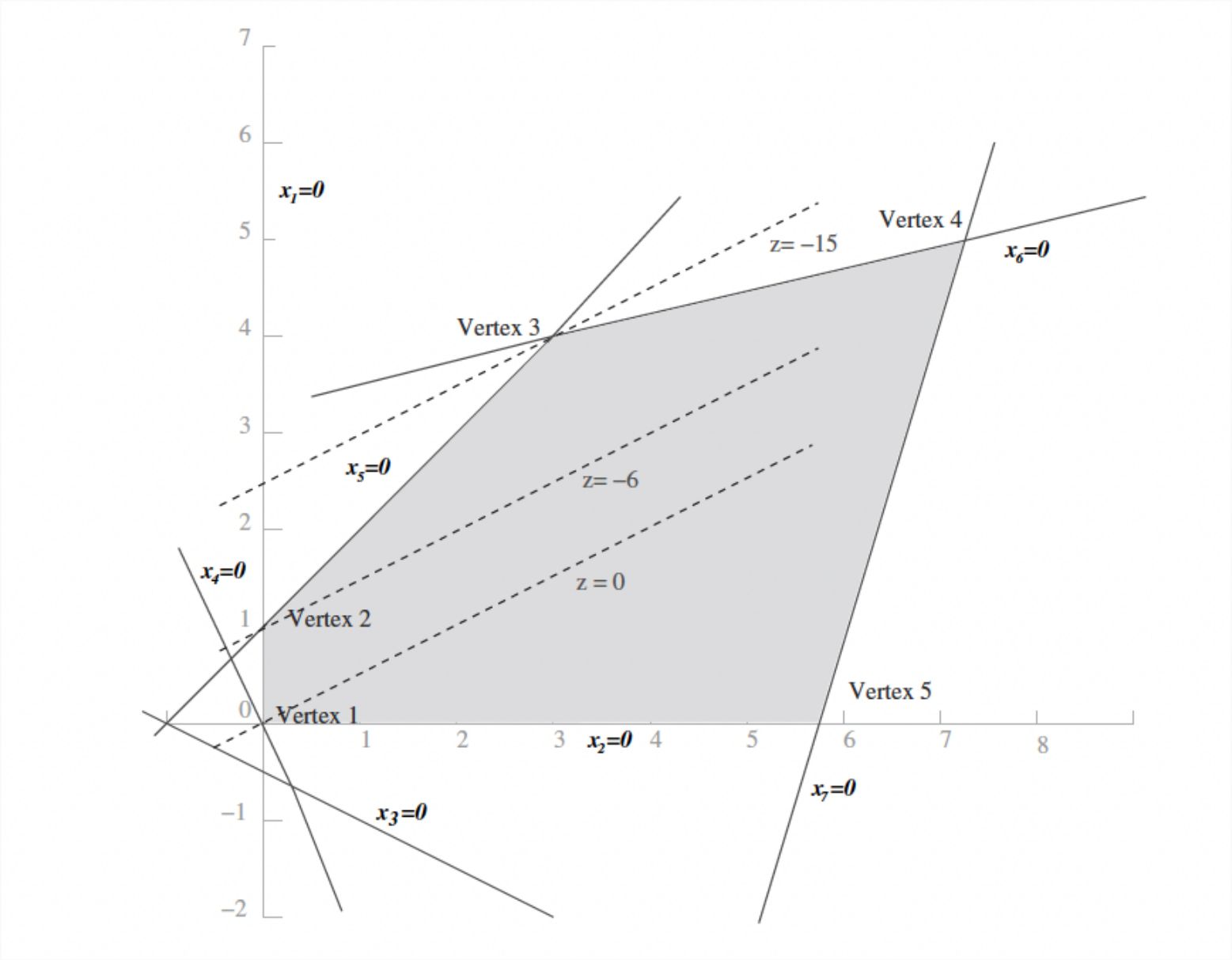
By setting $\lambda = 1$, we have that x_1 and x_5 are zero, while the other variables remain nonnegative.

We obtain the tableau corresponding to this point by performing the Jordan exchange of the row labeled x_5 (row 3) with the column labeled x_2 (column 2). The new tableau is as follows

		x_1	x_5	1
x_3	=	3	−2	3
x_4	=	3	−1	1
x_2	=	1	−1	1
x_6	=	−3	4	9
x_7	=	−3	−1	24
z	=	−3	6	−6

Note that z decreased from 0 to -6 and that the table corresponds to the vertex characterised by $x_1 = 0, x_5 = 0$.

Therefore, we have moved from the vertex $x_1 = 0, x_2 = 0$ to the vertex $x_1 = 0, x_5 = 0$.



Pricing and Ration Test

Given the tableau

$$\begin{array}{rcc} & x_N & 1 \\ x_B & H & h \\ z & c^T & \alpha \end{array}$$

where B represents the current set of basic variables and N represents the current set of nonbasic variables, a pivot step of the simplex method is a Jordan exchange between a basic and nonbasic variable according to the following:

Pricing and Ration Test

Pivot selection rules:

1. **Pricing** (selection of pivot column s): The pivot column is a column s with a negative element in the bottom row. These elements are called *reduced costs*.
2. **Ratio Test** (selection of pivot row r): The pivot row is a row r such that

$$-\frac{h_r}{H_{rs}} = \min_i\{-\frac{h_i}{H_{is}} \mid H_{is} < 0\}.$$

$$\begin{array}{rc} & \begin{array}{cc} x_N & 1 \end{array} \\ \begin{array}{r} x_B \\ z \end{array} & \left[\begin{array}{cc} H & h \\ c^T & \alpha \end{array}\right] \end{array} \quad \rightarrow \quad \begin{array}{rc} & \begin{array}{cc} x_{\widetilde{N}} & 1 \end{array} \\ \begin{array}{r} x_{\widetilde{B}} \\ z \end{array} & \left[\begin{array}{cc} \widetilde{H} & \widetilde{h} \\ \widetilde{c}^T & \widetilde{\alpha} \end{array}\right] \end{array}$$

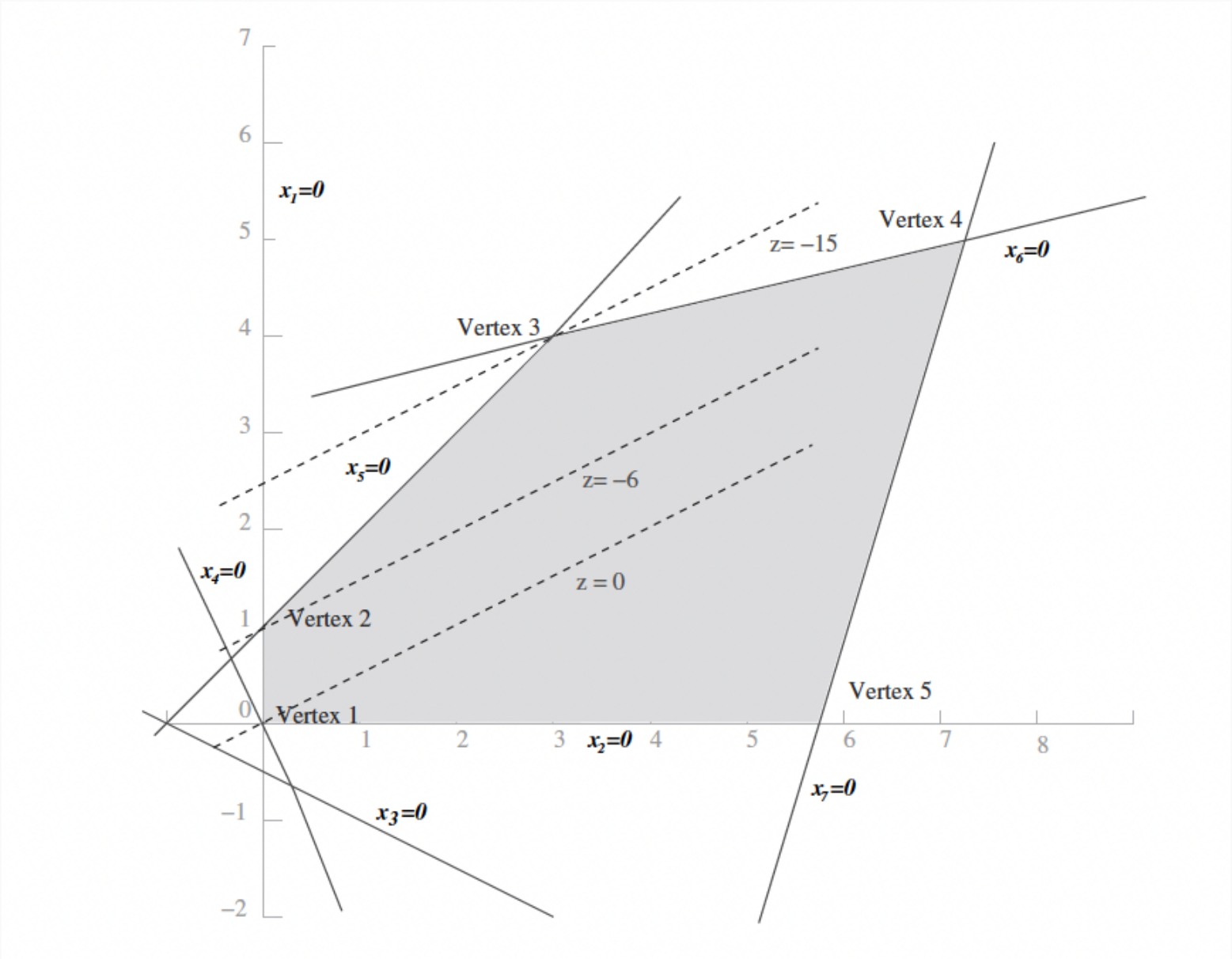
Here, by the Jordan exchange formulas $\widetilde{\alpha} = \alpha - \frac{c_s\, h_r}{H_{rs}} < \alpha$, where the strict inequality follows from the properties of c_s, h_r and H_{rs}.

Note that there is considerable flexibility in selection of the pivot column, as it is often the case that many of the reduced costs are negative.

One simple rule is to choose the column with the most negative reduced cost. This gives the biggest decrease in z per unit increase in the entering variable.

However, since we cannot tell how much we can increase the entering variable until we perform the ratio test, it is not generally true that this choice leads to the best decrease in z on this step, among all possible pivot columns.

Returning to the example, we are now interested how (or if it is needed) to move from vertex $x_1 = 0, x_5 = 0$ to another vertex.



		x_1	x_5	1
x_3	$=$	3	-2	3
x_4	$=$	3	-1	1
x_2	$=$	1	-1	1
x_6	$=$	-3	4	9
x_7	$=$	-3	-1	24
z	$=$	-3	6	-6

We see that column 1, the one labeled x_1, is the only possible choice for pivot column. The ratio test indicates that row 4, labeled by x_6, should be the pivot row. We thus obtain

		x_6	x_5	1
x_3	$=$	-1	2	12
x_4	$=$	-1	3	10
x_2	$=$	-0.33	0.33	4
x_1	$=$	-0.33	1.33	3
x_7	$=$	1	-5	15
z	$=$	1	2	-15

		x_6	x_5	1
x_3	$=$	-1	2	12
x_4	$=$	-1	3	10
x_2	$=$	-0.33	0.33	4
x_1	$=$	-0.33	1.33	3
x_7	$=$	1	-5	15
z	$=$	1	2	-15

In this tableau, all reduced costs are positive, and so the pivot column selection procedure does not identify an appropriate column.

This is as it should be, because this tableau is optimal! For any other feasible point than the one indicated by this tableau, we would have $x_6 \geq 0$ 0 and $x_5 \geq 0$, giving an objective $z = x_6 + 2\,x_5 - 15 \geq -15$.

Hence, we cannot improve z over its current value of -15 by allowing either x_5 or x_6 to enter the basis, and so the tableau is optimal.

		x_6	x_5	1
x_3	$=$	-1	2	12
x_4	$=$	-1	3	10
x_2	$=$	-0.33	0.33	4
x_1	$=$	-0.33	1.33	3
x_7	$=$	1	-5	15
z	$=$	1	2	-15

The values of the basic variables can be read from the last column of the optimal tableau. We are particularly interested in the values of the two variables x_1 and x_2 from the original standard formulation of the problem; they are $x_1 = 3$ and $x_2 = 4$.

In general, we have an optimal tableau when both the last column and the bottom row are nonnegative.

Note: when talking about the last row or last column, we do not include in our considerations the bottom right element of the tableau, the one indicating the current value of the objective. Its sign is irrelevant to the optimization process.

Simplex method

Simplex method

We consider that the linear program is in the canonical form.

We split the simplex method in two phases:

- Phase I: finds a starting point that satisfies the constraints, i.e., it identifies a starting vertex. Note that we cannot always start the initial tableau, given by the definition of the slacks variables, since these tableau correspond to the origin $(0, 0, ...0) \in \mathbb{R}^n$ and it does not belong always to the feasible region.
- Phase II: starts with a feasible tableau and applies the pivots needed to move to an optimal tableau, thus solving the linear program.

Simplex Algorithm

Simplex Method:

1. Construct an initial tableau. If the problem is in *standard form*, this process amounts to simply *adding slack variables*.
2. If the tableau is not feasible, apply a *Phase I* procedure to *generate a feasible tableau*, if one exists (TO DO in the next lectures). Let us consider that a starting vertex is found, i.e., we know a feasible tableau. For now we shall assume the origin $x_N = 0$ is feasible, but only to can implement Phase II, firstly.
3. Use the pricing rule to determine the pivot column s. If none exists, **STOP**; (a): **tableau is optimal**.
4. Use the ratio test to determine the pivot row r. If none exists, **STOP**; (b): **tableau is unbounded**.
5. Exchange $x_{B(r)}$ and $x_{N(s)}$ using a Jordan exchange on H_{rs} .
6. Go to Step 3.

The Phase II Procedure: optimal

Phase II comprises Steps 3 through 6 of the method above—that part of the algorithm that occurs after an initial feasible tableau has been identified.

The method terminates in one of two ways.

- Stop (a) indicates optimality. This occurs when the last row is nonnegative. In this case, there is no benefit to be obtained by letting any of the nonbasic variables x_N increase away from zero. We can verify this claim mathematically by writing out the last row of the tableau, which indicates that the objective function is

$$z = c^T x_N + \alpha.$$

  When $c \geq 0$ and $x_N \geq 0$, we have $z \geq \alpha$. Therefore, the point corresponding to the tableau, $x_B = h$ and $x_N = 0$, is optimal, with objective function value α.

The Phase II Procedure: unbounded

- The second way that the method above terminates, Stop (b), occurs when a column with a negative cost c_s has been identified, but the ratio test fails to identify a pivot row. This situation can occur only when all the entries in the pivot column $H_{\cdot s}$ are nonnegative.
  In this case, by allowing $x_{N(s)}$ to grow larger, without limit, we will be decreasing the objective function to $-\infty$ without violating feasibility.

$$\begin{array}{rcc} & x_N & 1 \\ x_B & H & h \\ z & c^T & \alpha \end{array} =$$

Unbounded

In other words, by setting $x_{N(s)} = \lambda$ for any positive value λ, we have for the basic variables x_B that

$$x_B(\lambda) = H_{\cdot s}\lambda + h \geq h \geq 0,$$

so the full set of variables $x(\lambda) \in \mathbb{R}^{m+n}$ is defined by the formula

$$x_j(\lambda) = \begin{cases} \lambda & \text{if} \quad j = N(s), \\ H_{is}\lambda + h_i & \text{if} \quad j = B(i), \\ 0 & \text{if} \quad j \in N \setminus \{N(s)\}, \end{cases}$$

which is feasible for all $\lambda \geq 0$.

The Phase II Procedure: unbounded

The objective function for $x(\lambda)$ is

$$z = c^T x_N(\lambda) + \alpha = c_s \lambda + \alpha,$$

which tends to $-\infty$ as $\lambda \to \infty$. Thus the set of points $x(\lambda)$ for $\lambda \geq 0$ identifies a ray of feasible points along which the objective function approaches $-\infty$.

More than one solution?

Linear programs may have more than one solution. In fact, given any collection of solutions $x^1, x^2, ..., x^k$, any other point in the *convex hull* of these solutions, defined by

$$\{x \,|\, x = \sum_{i=1}^{K} \alpha_i x^i, \sum_{i=1}^{K} \alpha_i = 1, \alpha_i \geq 0, i = 1, 2, ..., K\}$$

is also a solution.

To prove this claim, we need to verify that any such x is feasible with respect to the constraints $Ax \geq b$, $x \geq 0$ and also that x achieves the same objective value as each of the solutions $x^i, i = 1, 2, ..., K$. First, note that

$$A\,x = \sum_{i=1}^{K} \alpha_i A\,x^i \geq \sum_{i=1}^{K} \alpha_i\, b = b,$$

and so the inequality constraint is satisfied. Since x is a nonnegative combination of the nonnegative vectors x, it is also nonnegative, and so the constraint $x \geq 0$ is also satisfied.

Finally, since each x_i is a solution, we have that $p^T x^i = z$ for some scalar z_{opt} and all $i = 1, 2, ..., K$. Hence,

$$p^T x = \sum_{i=1}^{K} \alpha_i p^T x^i = \sum_{i=1}^{K} \alpha_i z_{opt} = z_{opt}.$$

Since x is feasible and attains the optimal objective value z_{opt}, we conclude that x is a solution, as claimed.

Phase II can be extended to identify multiple solutions by performing additional pivots on columns with zero reduced costs after an optimal tableau has been identified.

$$\begin{array}{rcc} & x_N & 1 \\ x_B = & H & h \\ z = & c^T & \alpha \end{array}$$

The Phase I Procedure

In all the problems we have examined to date, the linear program has been stated in standard form, and the tableau constructed from the problem data has been feasible.

This situation occurs when $x_i = 0,\ i = 1, 2, \ldots, n$ is feasible with respect to the constraints, that is, when the right-hand side b of all the inequality constraints is nonpositive.

$$\begin{array}{rcc} & x_N & 1 \\ x_B = & A & -b \\ z = & p^T & 0 \end{array}$$

In general, however, this need not be the case, and we are often faced with the task of identifying a feasible initial point (that is, a feasible tableau), so that we can go ahead and apply the Phase II procedure already described.

The Phase I Procedure

The process of identifying an initial feasible tableau is called Phase I.

Phase I entails the solution of a linear program that is different from, though closely related to, the problem we actually wish to solve.

It is easy to identify an initial feasible tableau for the modified problem, and its eventual solution tells us whether the original problem has a feasible tableau or not.

If the original problem has a feasible tableau, it can be easily constructed from the tableau resulting from Phase I.

The Phase I Procedure

The Phase I problem contains one *additional variable* x_0, a set of constraints that is the same as the original problem except for the addition of x_0 to some of them, and an *objective function* of x_0 itself.

It can be stated as follows:

$$\min_{x_0, x_1, x_2, \ldots, x_n \in \mathbb{R}} \quad z_0 = x_0$$

$$\begin{aligned}
\text{subject to} \quad x_{n+i} &= A_{i1}x_1 + \ldots + A_{in}x_n - b_i + x_0 \quad \text{if} \quad b_i > 0 \\
x_{n+i} &= A_{i1}x_1 + \ldots + A_{in}x_n - b_i \quad \text{if} \quad b_i \leq 0
\end{aligned}$$

$$x_1, x_2, \ldots, x_{n+m}, x_0 \geq 0.$$

The variable x_0 is an *artificial variable*.

Note that the objective z_0 is bounded below by 0, since x_0 is constrained to be nonnegative.

The Phase I Procedure

$$\min_{x_0, x_1, x_2, \ldots, x_n \in \mathbb{R}} \quad z_0 = x_0$$

$$\begin{array}{lrcccccccl} \text{subject to} & x_{n+i} & = & A_{i1}x_1 + & \ldots & + A_{in}x_n & - b_i & + x_0 & \text{if} & b_i > 0 \\ & x_{n+i} & = & A_{i1}x_1 + & \ldots & + A_{in}x_n & - b_i & & \text{if} & b_i \leq 0 \end{array}$$

$$x_1, x_2, \ldots, x_{n+m}, x_0 \geq 0.$$

We note a number of important facts about this problem:

- We can obtain a feasible point for the auxiliary problem by setting $x_0 = \max(\max_{1 \leq i \leq n} b_i, 0)$ and $x_N = 0$ for $N = \{1, 2, \ldots, n\}$. The dependent variables x_B, where $B = \{n+1, n+2, \ldots, n+m\}$, then take the following initial values:

  $$b_i > 0 \;\Rightarrow\; x_{n+i} = A_{i\cdot}x - b_i + x_0 = -b_i + \max_{1 \leq j \leq m, b_j > 0} b_i \geq -b_i + b_i = 0,$$

  $$b_i \leq 0 \;\Rightarrow\; x_{n+i} = A_{i\cdot}x - b_i = -b_i \geq 0.$$

  so that $x_B \geq 0$, and these components are also feasible.

The Phase I Procedure

$$\min_{x_0, x_1, x_2, \ldots, x_n \in \mathbb{R}} \quad z_0 = x_0$$

$$\begin{aligned}
\text{subject to} \quad x_{n+i} &= A_{i1}x_1 + \ldots + A_{in}x_n - b_i + x_0 \quad \text{if} \quad b_i > 0 \\
x_{n+i} &= A_{i1}x_1 + \ldots + A_{in}x_n - b_i \quad \text{if} \quad b_i \leq 0
\end{aligned}$$

$$x_1, x_2, \ldots, x_{n+m}, x_0 \geq 0.$$

A number of important facts about this problem:

- If there exists a point $\overline{x}$ that is feasible for the original problem, then the point $(x_0, x) = (0, \overline{x})$ is feasible for the Phase I problem. (It is easy to check this fact by verifying that $x_{n+i} \geq 0$ for $i = 1, 2, \ldots, m$.)

The Phase I Procedure

$$\min_{x_0, x_1, x_2, \ldots, x_n \in \mathbb{R}} \quad z_0 = x_0$$

$$\begin{array}{lrcccccccl} \text{subject to} & x_{n+i} & = & A_{i1}x_1 + & \ldots & + A_{in}x_n & - b_i & + x_0 & \text{if} & b_i > 0 \\ & x_{n+i} & = & A_{i1}x_1 + & \ldots & + A_{in}x_n & - b_i & & \text{if} & b_i \leq 0 \end{array}$$

$$x_1, x_2, \ldots, x_{n+m}, x_0 \geq 0.$$

A number of important facts about this problem:

- Conversely, if $(0, x)$ is a solution of the Phase I problem, then x is feasible for the original problem. We see this by examining the constraint set for the Phase I problem and noting that

$$b_i > 0 \;\Rightarrow\; 0 \leq x_{n+i} = A_{i\cdot}x - b_i + x_0 = A_{i\cdot}x - b_i,$$

$$b_i \leq 0 \;\Rightarrow\; 0 \leq x_{n+i} = A_{i\cdot}x - b_i,$$

  so that $Ax \geq b$.

The Phase I Procedure

$$\min_{x_0, x_1, x_2, \ldots, x_n \in \mathbb{R}} \quad z_0 = x_0$$

$$\begin{aligned}
\text{subject to} \quad x_{n+i} &= A_{i1}x_1 + \ldots + A_{in}x_n - b_i + x_0 \quad \text{if} \quad b_i > 0 \\
x_{n+i} &= A_{i1}x_1 + \ldots + A_{in}x_n - b_i \quad \text{if} \quad b_i \leq 0
\end{aligned}$$

$$x_1, x_2, \ldots, x_{n+m}, x_0 \geq 0.$$

A number of important facts about this problem:

- If (x_0, x) is a solution of the Phase I problem and x_0 is *strictly positive*, then the original problem must be infeasible. This fact follows immediately from the observations above: If the original problem were feasible, it would be possible to find a feasible point for the Phase I problem with objective zero.

We can set up this starting point by forming the initial tableau for the auxiliary problem, in the usual way and performing a "special pivot." We select the x_0 column as the pivot column and choose the pivot row to be a row with the most negative entry in the last column of the tableau.

After the special pivot, the tableau contains only nonnegative entries in its last column, and the simplex method can proceed, using the usual rules for pivot column and row selection.

An example

$$
\begin{aligned}
\min_{x1, x_2 \in \mathbb{R}} \quad & z = 4\,x_1 + 5\,x_2 \\
\text{subject to} \quad & x_1 + x_2 \geq -1, \\
& x_1 + 2\,x_2 \geq 1, \\
& 4\,x_1 + 2\,x_2 \geq 8, \\
& -x_1 - x_2 \geq -3, \\
& -x_1 + x_2 \geq 1, \\
& x_1, x_2 \geq 0.
\end{aligned}
$$

We start by loading the data into a tableau and then adding a column for the artificial variable x_0 and the Phase I objective z_0.

		x_1	x_2	x_0	1
x_3	$=$	1	1	0	1
x_4	$=$	1	2	1	-1
x_5	$=$	4	2	1	-8
x_6	$=$	-1	-1	0	3
x_7	$=$	-1	1	1	-1
z	$=$	4	5	0	0
z_0	$=$	0	0	1	0

$\rightarrow$

		x_1	x_2	x_5	1
x_3	$=$	1	1	0	1
x_4	$=$	-3	0	1	7
x_0	$=$	-4	-2	1	8
x_6	$=$	-1	-1	0	3
x_7	$=$	-5	-1	1	7
z	$=$	4	5	0	0
z_0	$=$	-4	-2	1	8

Since the objective of the auxiliary problem is bounded below (by zero), it can terminate only at an optimal tableau. Two possibilities then arise.

- The optimal objective z_0 is strictly positive. In this case, we conclude that the original problem is infeasible, and so we terminate without going to Phase II.
- The optimal objective z_0 is zero. In this case, x_0 must also be zero, and the remaining components of x are a feasible initial point for the original problem. We can construct a feasible table for the initial problem from the optimal tableau for the Phase I problem as follows.

  - First, if x_0 is still a dependent variable in the tableau (that is, one of the row labels), perform a Jordan exchange to make it an independent variable. (Since $x_0 = 0$, this pivot will be a degenerate pivot, and the values of the other variables will not change.)
  - Next, delete the column labeled by x_0 and the row labeled by z_0 from the tableau.

  The tableau that remains is feasible for the original problem, and we can proceed with Phase II, as described above.

Phase I:

1. If $b \leq 0$, then $x_B = -b$, $x_N =$ is a feasible point corresponding to the initial tableau and no Phase I is required. Skip to Phase II.
2. If $b > 0$, introduce the artificial variable x0 (and objective function $z_0 = x_0$) and set up the Phase I auxiliar problem and the corresponding tableau.
3. Perform the "special pivot" of the x_0 column with a row corresponding to the most negative entry of the last column to obtain a feasible tableau for Phase I.
4. Apply standard simplex pivot rules until an optimal tableau for the Phase I problem is attained.
   - If the optimal value (for z_0) is positive, stop: The original problem has no feasible point.
   - Otherwise, perform an extra pivot (if needed) to move x_0 to the top of the tableau.
5. Strike out the column corresponding to x_0 and the row corresponding to z_0 and proceed to Phase II.

What about our exemple?

$$
\begin{array}{rcccc|c}
 & & x_1 & x_2 & x_0 & 1 \\
x_3 & = & 1 & 1 & 0 & 1 \\
x_4 & = & 1 & 2 & 1 & -1 \\
x_5 & = & 4 & 2 & 1 & -8 \\
x_6 & = & -1 & -1 & 0 & 3 \\
x_7 & = & -1 & 1 & 1 & -1 \\
\hline
z & = & 4 & 5 & 0 & 0 \\
z_0 & = & 0 & 0 & 1 & 0
\end{array}
\rightarrow
\begin{array}{rcccc|c}
 & & x_1 & x_2 & x_5 & 1 \\
x_3 & = & 1 & 1 & 0 & 1 \\
x_4 & = & -3 & 0 & 1 & 7 \\
x_0 & = & -4 & -2 & 1 & 8 \\
x_6 & = & -1 & -1 & 0 & 3 \\
x_7 & = & -5 & -1 & 1 & 7 \\
\hline
z & = & 4 & 5 & 0 & 0 \\
z_0 & = & -4 & -2 & 1 & 8
\end{array}
$$

$$
\begin{array}{rcccc|c}
 & & x_1 & x_6 & x_5 & 1 \\
x_3 & = & 0 & -1 & 0 & 4 \\
x_4 & = & -3 & 0 & 1 & 7 \\
x_0 & = & -2 & 2 & 1 & 2 \\
x_2 & = & -1 & -1 & 0 & 3 \\
x_7 & = & -4 & 1 & 1 & 4 \\
\hline
z & = & -1 & -5 & 0 & 15 \\
z_0 & = & -2 & 2 & 1 & 2
\end{array}
\rightarrow
\begin{array}{rcccc|c}
 & & x_0 & x_6 & x_5 & 1 \\
x_3 & = & 0 & -1 & 0 & 4 \\
x_4 & = & 1.5 & -3 & -0.5 & 4 \\
x_1 & = & -0.5 & 1 & 0.5 & 1 \\
x_2 & = & 0.5 & -2 & -0.5 & 2 \\
x_7 & = & 2 & -3 & -1 & 0 \\
\hline
z & = & 0.5 & -6 & -0.5 & 14 \\
z_0 & = & 1 & 0 & 0 & 0
\end{array}
$$

		x_0	x_6	x_5	1
x_3	=	0	-1	0	4
x_4	=	1.5	-3	-0.5	4
x_1	=	-0.5	1	0.5	1
x_2	=	0.5	-2	-0.5	2
x_7	=	2	-3	-1	0
z	=	0.5	-6	-0.5	14
z_0	=	1	0	0	0

$\rightarrow$

		x_7	x_5	1
x_3	=	1/3	1/3	4
x_4	=	1	0.5	4
x_1	=	11/3	1/6	1
x_2	=	2/3	1/6	2
x_6	=	$-1/3$	$-1/3$	0
z	=	2	1.5	14

Although feasible, this tableau is not optimal for Phase II. Simplex rules lead us to perform the following degenerate pivot:

		x_7	x_5	1
x_3	=	-1	0	4
x_4	=	-3	-0.5	4
x_1	=	1	0.5	1
x_6	=	-2	-0.5	2
x_7	=	-3	-1	0
z	=	-6	-0.5	14

Finite Termination

Why cycling?

Again the definition of a vertex

For the feasible region $S := \{x \in \mathbb{R}^n \mid A\,x \geq b,\ x \geq 0\}$, let

$$x_{n+i} = A_{i\cdot}x - b_i, \quad i = 1, 2, ..., m.$$

where $A_{i\cdot}$ means the ith row of the matrix A, like in Matlab.

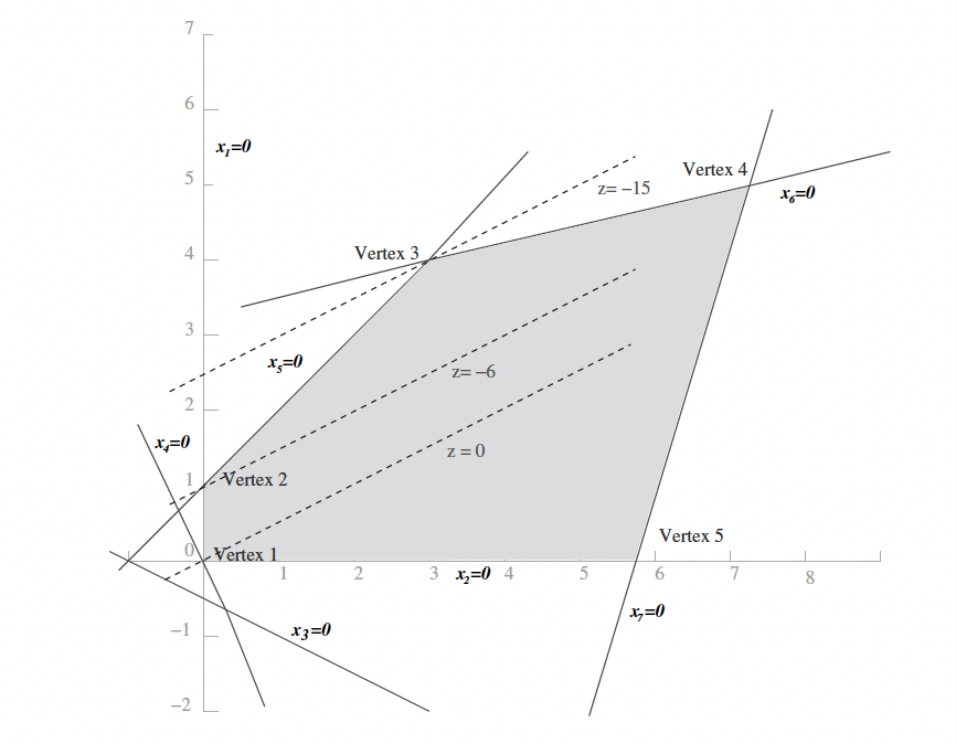
In other word, a *vertex* of S is any point $(x_1, x_2, ..., x_n) \in S$ that satisfies

$$x_N = 0,$$

where N is any subset of $\{1, 2, ..., n + m\}$ containing n elements such that the linear functions defined by $x_j, j \in N$, are linearly independent.

The set N that corresponds to a particular vertex may not be uniquely defined; that is, the same vertex may be specified by more than one set N.

See the vertex 1 from out example!



It is defined by

- $x_1 = 0$ and $x_2 = 0$;
- $x_1 = 0$ and $x_4 \equiv 2\,x_1 + x_2 = 0$;
- $x_2 = 0$ and $x_4 \equiv 2\,x_1 + x_2 = 0$.

So, by 3 different tableaux. Note that $(x_1, x_2) \mapsto x_1$ and $(x_1, x_2) \mapsto 2\,x_1 + x_2$ are linearly independent, and so the other two.

A vertex that can be specified by more than one set N is sometimes called a *degenerate vertex*.

Therefore, there is no reason why to affirm that the algorithm do not remain in a cycle in Vertex 1, by characterising it through 3 different tableaux.

Degenerate/nondegenerate tableau

Definition

A feasible tableau is *degenerate* if the last column contains any zero elements. If the elements in the last column are all strictly positive, the tableau is nondegenerate. A linear program is said to be nondegenerate if all feasible tableaus for that linear program are nondegenerate.

For instance, see

		x_7	x_5	1
x_3	$=$	-1	0	4
x_4	$=$	-3	-0.5	4
x_1	$=$	1	0.5	1
x_6	$=$	-2	-0.5	2
x_7	$=$	-3	-1	0
z	$=$	-6	-0.5	14

Geometrically, a tableau is nondegenerate if the vertex it defines is at the intersection of exactly n hyperplanes of the form $x_j = 0$; namely, those hyperplanes defined by $j \in N$.

Consequently, a linear program is nondegenerate if each of the vertices of the feasible region for that linear program is defined uniquely by a set N.

We encounter degenerate tableaus during the simplex method when there is a tie in the ratio test for selection of the pivot row. After the pivot is performed, zeros appear in the last column of the row(s) that tied but were not selected as pivots. The finite termination of the simplex method under these assumptions is now shown.

The nondegenerate case

Theorem

If a linear program is feasible and nondegenerate, then starting at any feasible tableau, the objective function strictly decreases at each pivot step. After a finite number of pivots the method terminates with an optimal point or else identifies a direction of unboundedness.

Proof.

At every iteration, we must have a nonoptimal, optimal, or unbounded tableau. In the latter two cases, termination occurs. In the first case, the following transformation occurs when we pivot on an element H_{rs} , for which $h_r > 0$ (nondegeneracy) and $c_s < 0$ (by pivot selection), and $H_{rs} < 0$ (by the ratio test):

$$\begin{array}{rcc} & x_N & 1 \\ x_B & = & \boxed{\begin{array}{cc} H & h \\ c^T & \alpha \end{array}} \\ z & = & \end{array} \quad \rightarrow \quad \begin{array}{rcc} & x_{\widetilde{N}} & 1 \\ x_{\widetilde{B}} & = & \boxed{\begin{array}{cc} \widetilde{H} & \widetilde{h} \\ \widetilde{c}^T & \widetilde{\alpha} \end{array}} \\ z & = & \end{array} \qquad \square$$

Proof

$$\begin{array}{rcc} & x_N & 1 \\ x_B = & H & h \\ z = & c^T & \alpha \end{array} \quad \rightarrow \quad \begin{array}{rcc} & x_{\widetilde{N}} & 1 \\ x_{\widetilde{B}} = & \widetilde{H} & \widetilde{h} \\ z = & \widetilde{c}^T & \widetilde{\alpha} \end{array}$$

Here, by the Jordan exchange formulas $\widetilde{\alpha} = \alpha - \frac{c_s h_r}{H_{rs}} < \alpha$, where the strict inequality follows from the properties of c_s, h_r and H_{rs}.

Hence, we can never return to the tableau with objective α, since this would require us to increase the objective at a later iteration, something the simplex method does not allow. Since we can only visit each possible tableau at most once, and since there are only a finite number of possible tableaus, the method must eventually terminate at either an optimal or an unbounded tableau.

In fact, a bound on the number of possible tableaus is obtained by determining the number ofways to choose the nonbasic setN (with n indices) from the index set $\{1, 2, ..., m + n\}$ which, by elementary combinatorics, is C^n_{m+n}.

The general case

Beale (1955) shown that for a degenerate linear program, reasonable rules for selecting pivots can fail to produce finite termination.

Finite termination depends crucially on the rule used to select pivot columns (in the event of more than one negative entry in the last row) and on the rule for selecting the pivot row (in the event of a tie in the ratio test). As shown above, even apparently reasonable rules can fail to produce finite termination.

We now modify the pivot selection rule of the simplex method to overcome this problem.

Smallest-subscript rule

This rule was introduced by Bland (1977) and is commonly called Bland's rule or the smallest-subscript rule.

Pivot selection rules:

1. **Pricing** (selection of pivot column s): The pivot column is the smallest $N(s)$ of nonbasic variable indices such that column s has a negative element in the bottom row (reduced cost).
2. **Ratio Test** (selection of pivot row r): The pivot row is the smallest $B(r)$ of basic variable indices such that row r satisfies

$$-\frac{h_r}{H_{rs}} = \min_i \{-\frac{h_i}{H_{is}} \mid H_{is} < 0\}.$$

In other words, among all possible pivot columns (those with negative reduced costs), we choose the one whose label has the smallest subscript. Among all possible pivot rows (those that tie for the minimum in the ratio test), we again choose the one whose label has the smallest subscript.

The following theorem establishes finiteness of the simplex method using the smallest-subscript rule, without any nondegeneracy assumption. The proof closely follows the one given by Chvátal (1983).

Theorem

If a linear program is feasible, then starting at any feasible tableau, and using the smallest-subscript anticycling rule, the simplex method terminates after a finite number of pivots at an optimal or unbounded tableau.

Theorem

For a linear program, the two-phase simplex method with the smallest-subscript anticycling rule terminates after a finite number of pivots with a conclusion that the problem is infeasible, or at an optimal or unbounded tableau.

Proof

Linear Programs in Nonstandard Form

Transforming Constraints and Variables

The complete process for solving a general linear program involves the following steps:

1. Convert maximization problems into minimization problems.

$$\max\langle p, x\rangle \qquad \Leftrightarrow \qquad \min -\langle p, x\rangle$$

2. Replace equations in inequalities

$$ax + b = 0 \qquad \Leftrightarrow \qquad ax + b \leq 0 \quad \text{and} \quad ax + b \geq 0.$$

3. Transform less-than inequalities into greater-than inequalities.

$$a\,x - b \leq 0 \qquad \Leftrightarrow \qquad -a\,x + b \geq 0.$$

Transforming Constraints and Variables

4. Use substitution to convert generally bounded variables into nonnegative.

$$
\begin{aligned}
0 \le x \le b \quad &\Leftrightarrow \quad x - b \le 0 \quad \Leftrightarrow \quad -x + b \ge 0, \\
x \le b \quad &\Leftrightarrow \quad x - b \le 0 \quad \Leftrightarrow \quad \underbrace{-x + b}_{:=y} \ge 0, \\
x \ge b \quad &\Leftrightarrow \quad \underbrace{x - b}_{:=y} \ge 0.
\end{aligned}
$$

5. Use substitution to convert free variables into nonnegative.

$$
x \in \mathbb{R} \quad \Leftrightarrow \quad x = y^+ - y^-, \qquad y^+, y^- \ge 0.
$$

6. Replace bounded variables and free variables and equations from the formulation.

Transforming Constraints and Variables

7. If the tableau is infeasible, apply the Phase I method to generate a feasible tableau. If Phase I terminates with a positive objective function value, stop and declare the problem infeasible.
8. Apply Phase II pivots to determine unboundedness or an optimal tableau.
9. Recover the values of the original variables if substitution was applied.
10. If the problem was originally a maximization, negate the objective value in the final tableau to give the optimal value of the original problem.

A final result about the Simplex Method

Theorem

Given any linear program, suppose we apply Scheme I together with the two-phase simplex method using the smallest-subscript anticycling rule. Then, after a finite number of pivots, the algorithm either terminates with a conclusion that the problem is infeasible or else arrives at an optimal or unbounded tableau.

Succes în călătoria voastră!