Aproximarea funcţiilor prin metoda celor mai mici pătrate

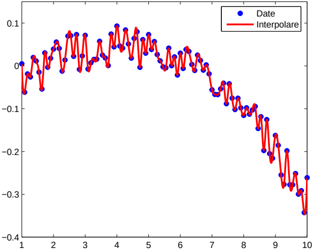
Prof.dr.ing. Gabriela Ciuprina

Universitatea "Politehnica" Bucureşti, Facultatea de Inginerie Electrică

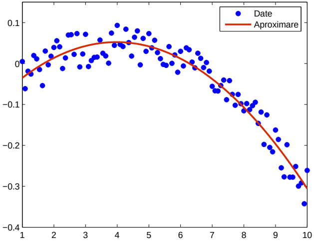
Suport didactic pentru disciplina *Algoritmi Numerici*, 2016-2017

Cuprins

Interpolare vs. aproximare



Interpolare



Aproximare

Se dă un set de date (x_k, y_k), $k = 0, \ldots, m$, unde $y_k = f(x_k)$.
Se caută o aproximare notată $g(x)$, unde $g(x_k) \approx y_k$.

Ideea metodei

Funcţia de aproximare se caută a.î:

$$\|g(x) - f(x)\| \quad \text{minimă}$$

Deoarece f este cunoscută într-un număr discret de puncte $\Rightarrow$ norma discretă (de exemplu norma Euclidiană discretă).

Metoda celor mai mici pătrate (CMMP) caută g a.î:

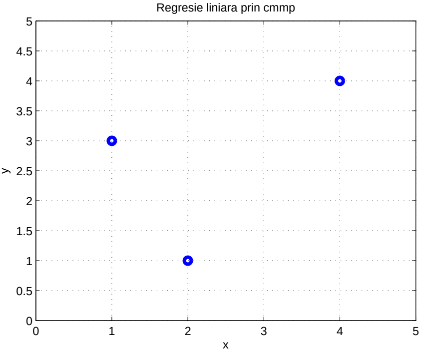
$$E = \sum_{k=0}^{m} (f(x_k) - g(x_k))^2 \quad \text{minimă} \tag{1}$$

Regresia liniară

$$g(x) = c_0 + c_1 x \quad c_0 = ?, c_1 = ?.$$

Exemplu: m = 3: (1,3), (2,1), (4,4).

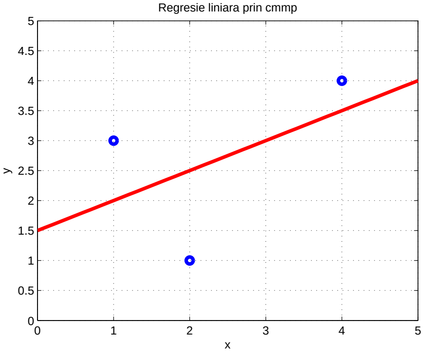
$$\begin{aligned} E(c_0, c_1) &= (y_0 - c_0 - c_1 x_0)^2 + \\ &+ (y_1 - c_0 - c_1 x_1)^2 + \\ &+ (y_2 - c_0 - c_1 x_2)^2 = \\ &= (3 - c_0 - c_1)^2 + \\ &+ (1 - c_0 - 2c_1)^2 + \\ &+ (4 - c_0 - 4c_1)^2. \end{aligned}$$



Regresia liniară

$$\begin{aligned} E(c_0, c_1) &= (y_0 - c_0 - c_1 x_0)^2 + (y_1 - c_0 - c_1 x_1)^2 + (y_2 - c_0 - c_1 x_2)^2 = \\ &= (3 - c_0 - c_1)^2 + (1 - c_0 - 2c_1)^2 + (4 - c_0 - 4c_1)^2. \end{aligned} \tag{2}$$

$$\left\{ \begin{array}{lcl} \frac{\partial E}{\partial c_0} & = & 0, \\ \frac{\partial E}{\partial c_1} & = & 0. \end{array} \right. \Rightarrow \left\{ \begin{array}{lcl} 3c_0 + 7c_1 & = & 8, \\ 7c_0 + 21c_1 & = & 21 \end{array} \right. \Rightarrow c_0 = 1.5, \quad c_1 = 0.5. \tag{3}$$



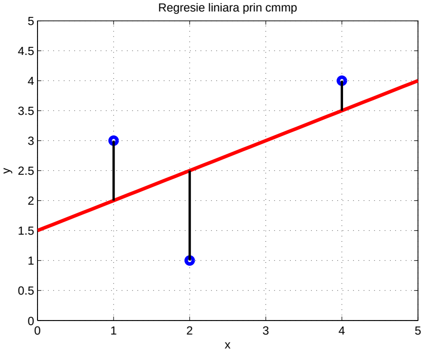
$g(1) = 2, g(2) = 2.5, g(3) = 3.5;$

$\min E = (3 - 2)^2 + (1 - 2.5)^2 + (4 - 3.5)^2$

Regresia liniară

$$\begin{aligned} E(c_0, c_1) &= (y_0 - c_0 - c_1 x_0)^2 + (y_1 - c_0 - c_1 x_1)^2 + (y_2 - c_0 - c_1 x_2)^2 = \\ &= (3 - c_0 - c_1)^2 + (1 - c_0 - 2c_1)^2 + (4 - c_0 - 4c_1)^2. \end{aligned} \tag{2}$$

$$\left\{ \begin{array}{rcl} \frac{\partial E}{\partial c_0} &=& 0, \\ \frac{\partial E}{\partial c_1} &=& 0. \end{array} \right. \Rightarrow \left\{ \begin{array}{rcl} 3c_0 + 7c_1 &=& 8, \\ 7c_0 + 21c_1 &=& 21 \end{array} \right. \Rightarrow c_0 = 1.5, \quad c_1 = 0.5. \tag{3}$$



$g(1) = 2, g(2) = 2.5, g(3) = 3.5;$

$\min E = (3 - 2)^2 + (1 - 2.5)^2 + (4 - 3.5)^2$

Regresia liniară

Set oarecare de date:

$$E(c_0, c_1) = \sum_{k=0}^{m} (y_k - c_0 - c_1 x_k)^2, \tag{4}$$

$$\left\{ \begin{array}{rcl} \frac{\partial E}{\partial c_0} & = & 0, \\ \frac{\partial E}{\partial c_1} & = & 0. \end{array} \right. \Rightarrow \left\{ \begin{array}{rcl} \sum_{k=0}^{m} -2(y_k - c_0 - c_1 x_k) & = & 0, \\ \sum_{k=0}^{m} -2x_k(y_k - c_0 - c_1 x_k) & = & 0. \end{array} \right. \tag{5}$$

$\Rightarrow$

$$\left\{ \begin{array}{rcl} c_0(m+1) + c_1 \sum_{k=0}^{m} x_k & = & \sum_{k=0}^{m} y_k, \\ c_0 \sum_{k=0}^{m} x_k + c_1 \sum_{k=0}^{m} x_k^2 & = & \sum_{k=0}^{m} x_k y_k. \end{array} \right. \tag{6}$$

Sistem normal

Regresia liniară

Soluţia se poate calcula explicit:

$$c_0 = \frac{d_0}{d}, \quad c_1 = \frac{d_1}{d}, \tag{7}$$

unde

$$d = (m+1)\sum_{k=0}^{m} x_k^2 - \left(\sum_{k=0}^{m} x_k\right)^2, \tag{8}$$

$$d_0 = \left(\sum_{k=0}^{m} x_k^2\right)\left(\sum_{k=0}^{m} y_k\right) - \left(\sum_{k=0}^{m} x_k\right)\left(\sum_{k=0}^{m} x_k y_k\right), \tag{9}$$

$$d_1 = (m+1)\sum_{k=0}^{m} x_k y_k - \left(\sum_{k=0}^{m} x_k\right)\left(\sum_{k=0}^{m} y_k\right). \tag{10}$$

Regresia liniară - alt raţionament

Condiţiile de interpolare pentru datele din tabel şi $g(x) = c_0 + c_1 x$:

$$c_0 + c_1 x_k = y_k, \quad k = 1, \ldots, m, \tag{11}$$

$\Rightarrow$ sistem supradeterminat de ecuaţii. Dacă notăm

$$\mathbf{A} = \begin{bmatrix} 1 & x_0 \\ 1 & x_1 \\ \vdots & \vdots \\ 1 & x_m \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_m \end{bmatrix}, \tag{12}$$

atunci (11) se scriu compact

$$\mathbf{Ac} = \mathbf{y}, \tag{13}$$

unde $\mathbf{c} = [c_0 \ c_1]^T$, iar sistemul normal este

$$\mathbf{A}^T\mathbf{Ac} = \mathbf{A}^T\mathbf{y}. \tag{14}$$

Regresia liniară - alt raţionament

Reluăm exemplul simplu: $m = 3$: (1,3), (2,1), (4,4).

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 4 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} 1 & 3 \\ 1 & 1 \\ 1 & 4 \end{bmatrix}, \tag{15}$$

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} 3 & 7 \\ 7 & 21 \end{bmatrix}, \quad \mathbf{A}^T\mathbf{y} = \begin{bmatrix} 8 \\ 21 \end{bmatrix}. \tag{16}$$

$$\left\{ \begin{array}{rcl} 3c_0 + 7c_1 & = & 8, \\ 7c_0 + 21c_1 & = & 21 \end{array} \right. \Rightarrow \quad c_0 = 1.5, \quad c_1 = 0.5. \tag{17}$$

Cazul general

Metoda celor mai mici pătrate nu este limitată la polinoame de gradul întâi.

Exemplu:

$g(x) = c_0 \ln(x) + c_1 \sin(x) + c_2 \sqrt{x}$.

1. Se minimizează funcţia definită de $E = \sum_{k=0}^{m} (f(x_k) - g(x_k))^2$;
2. Se impun condiţiile de minimizare;
3. Se rezolvă un sistem algebric liniar de dimensiune 3.

Cazul general

În general, se caută o aproximare de forma unui polinom generalizat

$$g(x) = \sum_{j=0}^{n} c_j \varphi_j(x), \tag{18}$$

unde $\varphi_i(x)$ se numesc *funcţii de bază*.
OBS: $n + 1 < m + 1$

$$E(c_0, c_1, \dots, c_n) = \sum_{k=0}^{m} (g(x_k) - f(x_k))^2 = \sum_{k=0}^{m} \left(\sum_{j=0}^{n} c_j \varphi_j(x_k) - y_k \right)^2. \tag{19}$$

Cazul general

$$E(c_0, c_1, \ldots, c_n) = \sum_{k=0}^{m} (g(x_k) - f(x_k))^2 = \sum_{k=0}^{m} \left(\sum_{j=0}^{n} c_j \varphi_j(x_k) - y_k \right)^2 . \tag{20}$$

Punctul de minim al acestei funcţii este un punct critic:

$$\frac{\partial E}{\partial c_i} = 0, \quad i = 0, \ldots, n. \tag{21}$$

Înlocuind expresia (20) în (21) rezultă

$$\sum_{j=0}^{n} \left(\sum_{k=0}^{m} \varphi_i(x_k) \varphi_j(x_k) \right) c_j = \sum_{k=0}^{m} y_k \varphi_i(x_k), \quad i = 0, \ldots, n. \tag{22}$$

Sistem normal dificil de rezolvat dacă φ_k nu sunt alese potrivit.

Cazul general - alt raţionament

Condiţiile de interpolare pentru datele din tabel şi $g(x)$

$$g(x_k) = y_k, \quad k = 1, \ldots, m, \tag{23}$$

$\Rightarrow$ sistem supradeterminat de ecuaţii. Notăm

$$\mathbf{A} = \begin{bmatrix} \varphi_0(x_0) & \varphi_1(x_0) & \cdots & \varphi_n(x_0) \\ \varphi_0(x_1) & \varphi_1(x_1) & \cdots & \varphi_n(x_1) \\ \vdots & \vdots & & \vdots \\ \varphi_0(x_m) & \varphi_1(x_m) & \cdots & \varphi_n(x_m) \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_m \end{bmatrix}, \tag{24}$$

atunci (23) se scriu compact

$$\mathbf{Ac} = \mathbf{y}, \tag{25}$$

unde $\mathbf{c} = [c_0, \ c_1 \ \ldots \ c_{n+1}]^T$. Sistemul normal:

$$\mathbf{A}^T\mathbf{Ac} = \mathbf{A}^T\mathbf{y}. \tag{26}$$

Funcţii de bază clasice

Sistemul normal de rezolvat

$$\mathbf{Nc} = \mathbf{t} \tag{27}$$

are, în cazul *regresiei liniare* forma

$$\mathbf{N} = \begin{bmatrix} m+1 & \sum_{k=0}^{m} x_k \\ \sum_{k=0}^{m} x_k & \sum_{k=0}^{m} x_k^2 \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} \sum_{k=0}^{m} y_k \\ \sum_{k=0}^{m} x_k y_k \end{bmatrix}. \tag{28}$$

Funcţii de bază clasice

Sistemul normal de rezolvat

$$\mathbf{Nc} = \mathbf{t} \tag{29}$$

are, în cazul *regresiei parabolice* $g(x) = c_0 + c_1 x + c_2 x^2$, forma:

$$\mathbf{N} = \left[\begin{array}{ccc} m+1 & \sum_{k=0}^{m} x_k & \sum_{k=0}^{m} x_k^2 \\ \sum_{k=0}^{m} x_k & \sum_{k=0}^{m} x_k^2 & \sum_{k=0}^{m} x_k^3 \\ \sum_{k=0}^{m} x_k^2 & \sum_{k=0}^{m} x_k^3 & \sum_{k=0}^{m} x_k^4 \end{array} \right], \quad \mathbf{t} = \left[\begin{array}{c} \sum_{k=0}^{m} y_k \\ \sum_{k=0}^{m} x_k y_k \\ \sum_{k=0}^{m} x_k^2 y_k \end{array} \right]. \tag{30}$$

OBS:
Generalizarea este uşoara pentru $n > 2$, dar sistemul este slab condiţionat. :(

Funcţii de bază clasice

Sistemul normal de rezolvat

$$\mathbf{Nc} = \mathbf{t} \tag{29}$$

are, în cazul *regresiei parabolice* $g(x) = c_0 + c_1 x + c_2 x^2$, forma:

$$\mathbf{N} = \begin{bmatrix} m+1 & \sum_{k=0}^m x_k & \sum_{k=0}^m x_k^2 \\ \sum_{k=0}^m x_k & \sum_{k=0}^m x_k^2 & \sum_{k=0}^m x_k^3 \\ \sum_{k=0}^m x_k^2 & \sum_{k=0}^m x_k^3 & \sum_{k=0}^m x_k^4 \end{bmatrix}, \quad \mathbf{t} = \begin{bmatrix} \sum_{k=0}^m y_k \\ \sum_{k=0}^m x_k y_k \\ \sum_{k=0}^m x_k^2 y_k \end{bmatrix}. \tag{30}$$

OBS:
Generalizarea este uşoara pentru $n > 2$, dar sistemul este slab condiţionat. :(Remedii ?

Funcţii de bază clasice

Remedii:

1. Folosirea unor alte funcţii de bază;
2. O altă strategie pentru CMMP, care nu rezolvă sistemul normal.

Funcţii de bază - polinoame Chebyshev

În cazul în care se caută polinoamelor algebrice, atunci folosirea polinoamelor Chebyshev, definite recursiv ca:

$$\begin{aligned} T_0(x) &= 1, & (31)\\ T_1(x) &= x, & (32)\\ T_j(x) &= 2xT_{j-1}(x) - T_{j-2}(x), & (33) \end{aligned}$$

generează un sistem de ecuaţii mult mai bine condiţionat.

OBS: Există un algoritm mai eficient de evaluare a aproximării $g(x)$ cu polinoame Chebyshev, decât rezolvarea sistemului normal [Cheney07].

Funcţii de bază ortonormale

Ideal (din p.d.v. al condiţionării) - varianta în care matricea coeficienţilor este matricea unitate, adică dacă

$$\sum_{k=0}^{m} \varphi_i(x_k)\varphi_j(x_k) = \delta_{ij}, \qquad \text{unde} \quad \delta_{ij} = \begin{cases} 1 & \text{dacă } i = j \\ 0 & \text{dacă } i \neq j \end{cases}, \tag{34}$$

ceea ce înseamnă că **funcţiile de bază sunt ortonormale**.

$$c_j = \sum_{k=0}^{m} y_k \varphi_j(x_k), \quad j = 0, \ldots, n. \tag{35}$$

Funcţii de bază ortonormale

Dacă notăm cu $\mathcal{G}$ spaţiul vectorial al funcţiilor generat de funcţii de bază $\varphi_j(x)$

$$\mathcal{G} = \left\{ g : \exists\, c_0, c_1, \ldots c_n, \text{ astfel încât } g(x) = \sum_{j=0}^{n} c_j \varphi_j(x) \right\}, \quad (36)$$

atunci o bază ortonormală se poate construi de exemplu printr-o procedură Gram-Schmidt.

Algoritm - paşi principali

Date:

- tabelul de valori;
- valorile în care se doreşte evaluarea polinomului de aproximare.

Se aleg:

- funcţiile de bază (în consecinţă şi gradul polinomului de aproximare).

Se calculează:

- valorile polinomului de aproximare în punctele dorite.

Algoritm - paşi principali

CMMP_SistemNormal

1. Asamblează $\mathbf{A}$ şi $\mathbf{y}$
2. Calculează $\mathbf{N} = \mathbf{A}^T\mathbf{A}$ şi $\mathbf{t} = \mathbf{A}^T\mathbf{y}$
3. Rezolvă sistemul pătrat $\mathbf{Nc} = \mathbf{t} \Rightarrow \mathbf{c}$
4. Evaluează polinomul de aproximare $g(x)$ în punctul dorit

Algoritm - paşi principali

Obs:

- În cazuri particulare, se poate evita calculul explicit al coeficienţilor **c**.
- Se poate ca tabelul de valori să includă numere complexe. În acest caz:

$$\mathbf{N} = \mathbf{A}^H \mathbf{A}, \qquad \mathbf{t} = \mathbf{A}^H \mathbf{y}$$

  unde $\mathbf{A}^H$ (notată uneori şi cu $\mathbf{A}^*$) este *transpusa hermitiană* adica $(a^H)_{ij} = \overline{a_{ji}}$.
- Sistemul normal $\mathbf{A}^H \mathbf{A}\mathbf{c} = \mathbf{A}^H \mathbf{y}$ este nesingular dcaă şi numai dacă **A** este de rang complet [Trefethen97].

1. Netezirea datelor

- Q: În cazul unor date experimentale, ce grad să aibă polinomul de aproximare ?
- A: Am putea creşte gradul polinomului până când avem o aproximare suficient de netedă a datelor.
- Q: Cum să stabilim cantitativ gradul de netezire?
- A: Folosind criterii din statistică.

1. Netezirea datelor

- Q: În cazul unor date experimentale, ce grad să aibă polinomul de aproximare ?
- A: Am putea creşte gradul polinomului până când avem o aproximare suficient de netedă a datelor.
- Q: Cum să stabilim cantitativ gradul de netezire?
- A: Folosind criterii din statistică.

1. Netezirea datelor

- Q: În cazul unor date experimentale, ce grad să aibă polinomul de aproximare ?
- A: Am putea creşte gradul polinomului până când avem o aproximare suficient de netedă a datelor.
- Q: Cum să stabilim cantitativ gradul de netezire?
- A: Folosind criterii din statistică.

1. Netezirea datelor

- Q: În cazul unor date experimentale, ce grad să aibă polinomul de aproximare ?
- A: Am putea creşte gradul polinomului până când avem o aproximare suficient de netedă a datelor.
- Q: Cum să stabilim cantitativ gradul de netezire?
- A: Folosind criterii din statistică.

1. Netezirea datelor

Forsythe [1957]:

- Se construiesc funcţii de bază ortogonale adaptate setului de date;
- Se pp. că valorile din tabel sunt afectate de erori

$$y_i = p_N(x_i) + \varepsilon_i, \qquad i = 0, \ldots, m$$

  unde ε_i sunt presupune variabile aleatoare independente distribuite normal.
- Pentru fiecare aproximare de grad n determinată se calculează varianţa

$$\sigma_n^2 = \frac{1}{m-n} \sum_{i=0}^{m} [y_i - p_n(x_i)]^2$$

.

1. Netezirea datelor

Teoria statistică:
Dacă tendinţa datelor din tabel e într-adevăr polinom de gradul N atunci are loc

$$\sigma_0^2 > \sigma_1^2 > \cdots \sigma_N^2 = \sigma_{N+1}^2 = \cdots \sigma_m^2$$

.
Algoritmul va calcula aproximări de ordin crescător până când varianţa nu se mai modifică. Pentru detalii ale acestui algoritm, consultaţi [Cheney07, subcapitolul 12.2].

2. Cazul unei funcţii date prin cod

Dacă $f : [a, b] \to \mathbb{R}$ e dată prin cod, atunci mărimea de minimizat foloseşte o normă continuă

$$E(c_0, c_1, \ldots, c_n) = \int_a^b [g(x) - f(x)]^2 \, \mathrm{d}x. \tag{37}$$

În anumite aplicaţii e de dorit că aproximarea să se potrivească mai bine pe anumite intervale, caz în care se folosesc funcţii pondere $w(x)$:

$$E(c_0, c_1, \ldots, c_n) = \int_a^b [g(x) - f(x)]^2 w(x) \, \mathrm{d}x. \tag{38}$$

2. Cazul unei funcţii date prin cod

Sistemul normal de ecuaţii devine

$$\sum_{j=0}^{n} \left[\varphi_i(x)\varphi_j(x)w(x)\,\mathrm{d}x\right] c_j = \int_a^b f(x)\varphi_i(x)w(x)\,\mathrm{d}x, \quad i = 0, \ldots, n. \tag{39}$$

Au fost propuse mai multe seturi de astfel de funcţii ortogonale şi ponderi potrivite.

3. Probleme de CMMP neliniare

Până acum polinoamele de interpolare = combinaţii liniare de coeficienţi necunoscuţi.

Dar dacă:

Pentru setul de date (x_k, y_k), $k = 0, m$ se caută $g(x) = e^{cx}$?

Metoda CMMP minimizează funcţia

$$E(c) = \sum_{k=0}^{m} \left(e^{cx_k} - y_k\right)^2 \tag{40}$$

Minimul are loc pentru $E'(c) = 0$, deci

$$2\sum_{k=0}^{m} \left(e^{cx_k} - y_k\right) e^{cx_k} x_k = 0, \tag{41}$$

3. Probleme de CMMP neliniare

Pentru setul de date (x_k, y_k), $k = 0, m$ se caută $g(x) = e^{cx}$. CMMP duce la

$$\sum_{k=0}^{m} (e^{cx_k} - y_k)\, e^{cx_k} x_k = 0, \tag{42}$$

ecuaţie neliniară în $c \Rightarrow$

- rezolvare de ecuaţii neliniare;
- metode de minimizare.

3. Probleme de CMMP neliniare

Pentru setul de date (x_k, y_k), $k = \overline{0, m}$ se caută $g(x) = e^{cx}$.
Acestea sunt **probleme de CMMP neliniare**.
OBS:
Reformulare ca o problemă de CMMP liniară:
Pentru setul de date $(x_k, \ln(y_k))$, $k = \overline{0, m}$ se caută $h(x) = cx$.
Valoarea c din problema reformulată nu este soluţia ecuaţiei neliniare (42), dar aproximarea $e^{h(x)}$ s-ar putea să fie satisfăcătoare.

Recapitularea ideilor de până acum

- Se caută aproximarea unor date dintr-un tabel de valori;
- CMMP găseşte o funcţie a.î. distanţa dintre ea şi tabelul de date să fie minimă (în normă Euclidiană);
- Aceasta problemă este echivalentă cu rezolvarea unui sistem de ecuaţii supradimensionat $\mathbf{Ac} = \mathbf{y}$;
- Acesta se poate aduce la un sistem de ecuaţii normal, cu matricea coeficienţilor pătrată;
- Condiţionarea sistemului normal depinde de alegerea funcţiilor de bază;

Recapitularea ideilor de până acum

- Polinoamele algebrice clasice se pot folosi pentru regresii liniare sau parabolice;
- Pentru ordine mai mari e mai bine să se folosească polinoame Chebyshev;
- O procedură eficientă poate folosi polinoame ortogonale adaptate setului de date;
- Găsirea ordinului optim se poate facând o analiză statistică.

Recapitularea ideilor de până acum

CMMP poate fi privită ca o metodă de rezolvare a unui sistem supradeterminat, care a fost notat

$$\mathbf{Ac} = \mathbf{y}.$$

Minimizarea normei Euclidiene discrete este echivalentă cu minimizarea normei reziduului

$$\mathbf{r} = \mathbf{Ac} - \mathbf{y}.$$

În cele ce urmează, vom prezenta alţi algoritmi CMMP, care nu asamblează sistemul normal.

Recapitularea ideilor de până acum

Problema formulată va fi aceea a rezolvării unui sistem supradeterminat de ecuaţii, pe care îl vom formula cu notaţia standard:

$$\mathbf{Ax} = \mathbf{b},$$

A - matrice dreptunghiulară de dimensiuni $(m+1) \times (n+1)$
b - vector coloană de dimensiune $(n+1)$
x - soluţia care se caută în sensul CMMP.

Recapitularea ideilor de până acum

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

x reprezintă punctul pentru care se atinge minimul expresiei

$$E(x_0, x_1, \ldots, x_n) = \sum_{k=0}^{m} \sum_{j=0}^{n} \left(a_{kj}x_j - b_k\right)^2$$

Dacă notăm reziduul (care depinde de **x**)

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

atunci

$$\|\mathbf{r}\|_2^2 = E(x_0, x_1, \ldots, x_n)$$

Un altă interpretare algebrică

$$\mathbf{Ax} = \mathbf{b}$$

$$\mathbf{r} = \mathbf{b} - \mathbf{Ax}$$

Rezolvarea unui sistem supradeterminat în sensul CMMP este echivalentă cu minimizarea normei Euclidiene a reziduului.

$\Leftrightarrow$

Rezolvarea unui sistem supradeterminat în sensul CMMP este echivalentă cu găsirea unei soluţii pentru care reziduul este ortogonal pe imaginea transformării liniare **Ax**

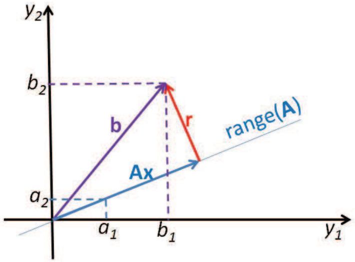
$$\mathbf{r} \perp \mathrm{range}(\mathbf{A})$$

Un altă interpretare algebrică

Cazul $m = 1, n = 0$ (2 ec., 1 nec., pp. numere reale):

$$\begin{aligned} a_1 x &= b_1 \\ a_2 x &= b_2 \end{aligned} \tag{43}$$

$$\text{range}(\mathbf{A}) = \left\{ \mathbf{y} \in \mathbb{R}^2 : \exists x \in \mathbb{R}, y_1 = a_1 x, y_2 = a_2 x \right\}$$



Factorizarea QR

Factorizare QR redusă

$$\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$$

$\mathbf{A}$ - matrice dreptunghiulară $(m+1) \times (n+1)$
$\hat{\mathbf{Q}}$ - matrice dreptunghiulară $(m+1) \times (n+1)$, cu coloane ortonormale
$\hat{\mathbf{R}}$ - matrice pătrată $(n+1) \times (n+1)$, superior triunghiulară

$\hat{\mathbf{Q}}^H \hat{\mathbf{Q}} = \mathbf{I}_{n+1}$
$r_{ii} > 0$, $r_{ij} = 0$ pentru $j < i$.

Factorizarea QR

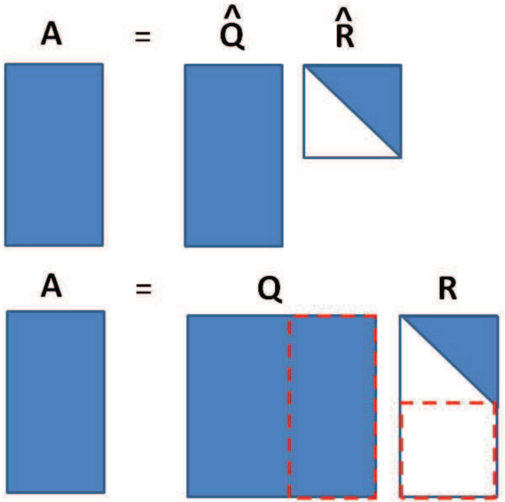
Factorizare QR completă

$$\mathbf{A} = \mathbf{QR}$$

$\mathbf{A}$ - matrice dreptunghiulară $(m+1) \times (n+1)$
$\hat{\mathbf{Q}}$ - matrice pătrată $(m+1) \times (m+1)$, unitară
$\hat{\mathbf{R}}$ - matrice dreptunghiulară $(m+1) \times (n+1)$, superior triunghiulară

$\hat{\mathbf{Q}}^H \hat{\mathbf{Q}} = \mathbf{I}_{m+1}$
$r_{ii} > 0$, $r_{ij} = 0$ pentru $j < i$.

Factorizarea QR



Ideea QR

$$\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$$

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

$$\hat{\mathbf{Q}}\hat{\mathbf{R}}\mathbf{x} = \mathbf{b}$$

$$\hat{\mathbf{Q}}^H\hat{\mathbf{Q}}\hat{\mathbf{R}}\mathbf{x} = \hat{\mathbf{Q}}^H\mathbf{b}$$

Deoarce

$$\hat{\mathbf{Q}}^H\hat{\mathbf{Q}} = \mathbf{I}$$

rezulta

$$\hat{\mathbf{R}}\mathbf{x} = \hat{\mathbf{Q}}^H\mathbf{b}$$

Algoritm - paşi principali

$$\mathbf{Ax} = \mathbf{b}$$

Date:

- matrice dreptunghiulară, cu coeficienţi complecşi $\mathbf{A}$;
- vectorul termenilor liberi $\mathbf{b}$

Se calculează:

- soluţia sistemului supradeterminat $\mathbf{x}$ în sensul CMMP.

Algoritm - paşi principali

CMMP_QR

1. Calculează factorizarea QR redusă a matricei $\mathbf{A}$: $\mathbf{A} = \hat{\mathbf{Q}}\hat{\mathbf{R}}$
2. Calculează vectorul $\mathbf{t} = \hat{\mathbf{Q}}^H \mathbf{b}$
3. Rezolvă sistemul pătrat, superior triunghiular $\hat{\mathbf{R}}\mathbf{x} = \mathbf{t} \Rightarrow \mathbf{x}$

Algoritm - calculul factorizării

Procedură de ortogonalizare Gram-Schmidt aplicată coloanelor matricei **A**

$$\mathbf{A} = \mathbf{R}_1\mathbf{R}_2\cdots\mathbf{R}_{n+1} = \hat{\mathbf{Q}}$$

apoi

$$\hat{\mathbf{R}} = \mathbf{R}_{n+1}^{-1}\mathbf{R}_n^{-1}\cdots\mathbf{R}_1^{-1}$$

Sau, mai eficient $\Rightarrow$

Algoritm - calculul factorizării

Procedură Householder

$$\mathbf{Q}_{n+1}\mathbf{Q}_n \cdots \mathbf{Q}_1\mathbf{A} = \mathbf{R}$$

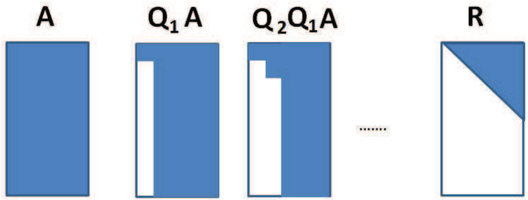
apoi

$$\mathbf{Q} = \mathbf{Q}_1^H\mathbf{Q}_2^H \cdots \mathbf{Q}_{n+1}^H$$

Obs

- $\mathbf{Q}_k$ - matrice elementare unitare, urmăresc eliminarea elementelor ca la Gauss;
- Se obţine factorizarea QR completă;

Algoritm - calculul factorizării



Trecerea la factorizarea QR redusă se obţine prin ultimelor coloane din **Q** şi a ultimelor linii din **R**.

Factorizarea SVD

SVD = *singular value decomposition* (desc. în valori singulare)

Factorizare SVD redusă

$$\mathbf{A} = \hat{\mathbf{U}}\hat{\mathbf{S}}\mathbf{V}^H$$

$\mathbf{A}$ - matrice dreptunghiulară $(m+1) \times (n+1)$ (presupusă de rang complet)
$\hat{\mathbf{U}}$ - matrice dreptunghiulară $(m+1) \times (n+1)$, cu coloane ortonormale
$\hat{\mathbf{S}}$ - matrice pătrată $(n+1) \times (n+1)$, diagonală
$\mathbf{V}$ - matrice pătrată $(n+1) \times (n+1)$, unitară

$\hat{\mathbf{V}}^H\hat{\mathbf{V}} = \mathbf{I}_{n+1}$
$s_{ii} > 0$, $s_{ij} = 0$ pentru $i \neq j$.

Factorizarea SVD

SVD = *singular value decomposition* (desc. în valori singulare)

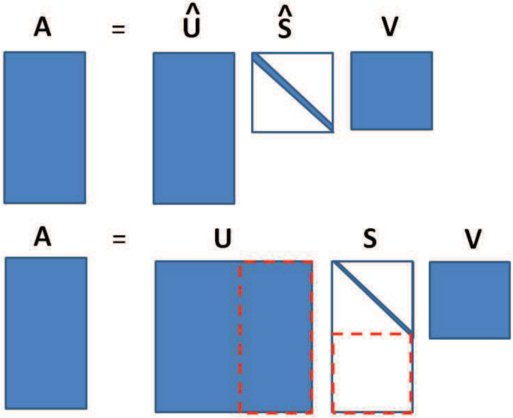
Factorizare SVD completă

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^H$$

A - matrice dreptunghiulară $(m+1) \times (n+1)$ (presupusă de rang complet)
U - matrice pătrată $(m+1) \times (m+1)$, unitară
S - matrice dreptunghiulară $(m+1) \times (n+1)$, diagonală
V - matrice pătrată $(n+1) \times (n+1)$, unitară

$\hat{\mathbf{V}}^H\hat{\mathbf{V}} = \mathbf{I}_{n+1}$; $\hat{\mathbf{U}}^H\hat{\mathbf{U}} = \mathbf{I}_{m+1}$
$s_{ii} > 0$, $s_{ij} = 0$ pentru $i \neq j$.

Factorizarea SVD



Ideea SVD

$$\mathbf{A} = \hat{\mathbf{U}}\hat{\mathbf{S}}\mathbf{V}^H$$

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

$$\hat{\mathbf{U}}\hat{\mathbf{S}}\mathbf{V}^H\mathbf{x} = \mathbf{b}$$

$$\hat{\mathbf{U}}\hat{\mathbf{S}}\mathbf{V}^H\mathbf{x} = \hat{\mathbf{U}}\hat{\mathbf{U}}^H\mathbf{b}$$

$$\hat{\mathbf{S}}\mathbf{V}^H\mathbf{x} = \hat{\mathbf{U}}^H\mathbf{b}$$

$$\hat{\mathbf{S}}(\mathbf{V}^H\mathbf{x}) = \hat{\mathbf{U}}^H\mathbf{b}$$

$$\hat{\mathbf{S}}\mathbf{w} = \hat{\mathbf{U}}^H\mathbf{b}$$

unde am notat

$$\mathbf{V}^H\mathbf{x} = \mathbf{w}$$

$$\mathbf{V}^H\mathbf{x} = \mathbf{V}^H\mathbf{V}\mathbf{w}$$

$$\mathbf{x} = \mathbf{V}\mathbf{w}$$

Algoritm - paşi principali

$$\mathbf{Ax} = \mathbf{b}$$

Date:

- matrice dreptunghiulară, cu coeficienţi complecşi **A**;
- vectorul termenilor liberi **b**

Se calculează:

- soluţia sistemului supradeterminat **x** în sensul CMMP.

Algoritm - paşi principali

CMMP_SVD

1. Calculează descompunerea redusă în valori singulare a matricei $\mathbf{A}$: $\mathbf{A} = \hat{\mathbf{U}}\hat{\mathbf{S}}\mathbf{V}^H$
2. Calculează vectorul $\mathbf{t} = \mathbf{U}^H\mathbf{b}$
3. Rezolvă sistemul diagonal $\hat{\mathbf{S}}\mathbf{w} = \mathbf{t} \Rightarrow \mathbf{w}$
4. Calculează $\mathbf{x} = \mathbf{V}\mathbf{w}$.

Pentru detalii, consultaţi [Trefethen97],[Dumitrescu98].

Pseudoinversa unei matrice dreptunghiulare

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad \Rightarrow \mathbf{x} = \mathbf{A}^{+}\mathbf{b}$$

Pseudoinversa unei matrice diagonale este tot o matrice diagonală având pe diagonală inversele:

$$\mathbf{D} = \begin{bmatrix} d_1 & 0 & 0 \\ 0 & d_2 & 0 \end{bmatrix} \quad \Rightarrow \quad \mathbf{D}^{+} = \begin{bmatrix} 1/d_1 & 0 \\ 0 & 1/d_2 \\ 0 & 0 \end{bmatrix} \tag{44}$$

Pseudoinversa unei matrice dreptunghiulare

$$\mathbf{Ax} = \mathbf{b} \quad \Rightarrow \mathbf{x} = \mathbf{A}^{+}\mathbf{b}$$

$$\hat{\mathbf{U}}\hat{\mathbf{S}}\mathbf{V}^{H}\mathbf{x} = \mathbf{b}$$
$$\hat{\mathbf{S}}\mathbf{V}^{H}\mathbf{x} = \hat{\mathbf{U}}^{H}\mathbf{b}$$
$$\mathbf{V}^{H}\mathbf{x} = \hat{\mathbf{S}}^{+}\hat{\mathbf{U}}^{H}\mathbf{b}$$
$$\mathbf{x} = \mathbf{V}\hat{\mathbf{S}}^{+}\hat{\mathbf{U}}^{H}\mathbf{b}$$

Rezultă că

$$\mathbf{A}^{+} = \mathbf{V}\hat{\mathbf{S}}^{+}\hat{\mathbf{U}}^{H}$$

Pseudoinversa unei matrice dreptunghiulare

Important:

- Pseudoinversa este unică dacă descompunerea în valori singulare se face astfel încat acestea să fie ordonate în S: $\sigma_1 \geq \sigma_2 \geq ....$ [Cheney]
- În rezolvarea în sens CMMP a problemei $\mathbf{Ax} = \mathbf{b}$ prin descompunere SVD, este util să se analizeze valorile singulare ale matricei.
  Este bine ca valorile singulare mai mici decât o anumită toleranţă să fie anulate.
  O astfel de abordare face ca metoda să fie cea mai stabilă din toate variantele.

Pseudoinversa unei matrice dreptunghiulare

Proprietăţi ale pseudoinversei (proprietăţi Penrose):

1. $\mathbf{A} = \mathbf{A}\mathbf{A}^{+}\mathbf{A}$
2. $\mathbf{A}^{+} = \mathbf{A}^{+}\mathbf{A}\mathbf{A}^{+}$
3. $\mathbf{A}\mathbf{A}^{+} = (\mathbf{A}\mathbf{A}^{+})^{T}$
4. $\mathbf{A}^{+}\mathbf{A} = (\mathbf{A}^{+}\mathbf{A})^{T}$

Comparaţii între algoritmi

1. Din punct de vedere al timpului de calcul:
   - CMMP prin rezolvarea sistemului normal: $O(mn^2 + n^3/3)$;
   - CMMP prin QR: $O(2mn^2 + 2n^3/3)$;
   - CMMP prin SVD: $O(2mn^2 + 11n^3)$;
2. Din punct de vedere al stabilităţii:
   - CMMP prin rezolvarea sistemului normal - de folosit doar pentru regresii liniare sau parabolice;
   - CMMP prin QR - stabil dacă rang(**A**) = n+1;
   - CMMP prin SVD - trebuie folosit dacă **A** are deficienţa de rang.

Matlab: `qr`, `svd`, `pinv`.

Referinţe

Minimal:
[AN] Gabriela Ciuprina,
Algoritmi numerici pentru calcule ştiintifice în ingineria electrică
Editura MatrixROM, 2013, pag. 143-168.
[Ioan98] D. Ioan et al., *Metode numerice in ingineria electrica*, Ed. Matrix Rom, Bucuresti, 1998. (Capitolul 13)
Alte recomandări:
[Cheney] Ward Cheney and David Kincaid, *Numerical Mathematics and Computing*, Brooks/Cole publishing Company,2000.
[Trefethen97] Lloyd N. Trefethen, David Bau III, *Numerical Linear Algebra*, SIAM 1997.
[Dumitrescu98] Bogdan Dumitrescu, Corneliu Popeea, Boris Jora, *Metode de calcul numeric matriceal. Algoritmi fundamentali*, Editura All, 1998.

Tema 8 (din categoria "examen")

1. Pregătiţi date sintetice asfel: alegeţi un polinom de grad 10 şi perturbaţi-l cu valori aleatoare distribuite normal. Generaţi un tabel de date cu 50 de puncte;
2. Implementaţi şi testaţi procedura de aproximare cu determinarea optimă a gradului polinomului de aproximare.

Notă: puteţi folosi orice fel de funcţii `matlab` doriţi.
Scrieţi un scurt raport care să reflecte rezolvarea temei. Este obligatoriu ca raportul să aibă: o pagină de titlu, un cuprins generat automat, o lista de referinţe. Daţi o structură coerentă raportului.