

(curs 13 - 58)

9.4. TCD modificata si compresia audio

- ne întoarcem la problema semnalelor uni-dimensionale și discutăm abordări de ultimă oră pentru compresia audio
- deși s-ar putea crede că o dimensiune este mai ușor de abordat decât două, provocarea este că sistemul auditiv uman este foarte sensibil în domeniul frecvență, și „artefactele” nedorite introduse prin compresie și decompresie sunt detectate mai ușor
- din acest motiv, este obișnuit ca metodele de compresie a sunetului să folosească trucuri sofisticate menite să ascundă faptul că a avut loc compresia
- prima dată, vom introduce TCD4, o nouă versiune a Transformantei Cosinus Discrete, și aşa-numita Transformată Cosinus Discretă Modificată (TCDM)
- TCDM este reprezentată printr-o matrice care nu este pătratică și astfel, spre deosebire de TCD și TCD4, nu este inversabilă
- totuși, când este aplicată pentru ferestre de timp suprapuse, poate fi folosită pentru a reconstrui complet fluxul de date inițial
- poate fi combinată cu cuantizarea pentru a efectua compresia cu pierdere de informații cu o degradare minimală a calității sunetului
- TCDM stă la baza majorității formatelor curente de compresie audio, cum ar fi MP3, AAC, și WMA

9.4.1. Transformata cosinus discreta modificata

- începem cu o formă ușor diferită a TCD introdusă mai devreme
- există patru versiuni diferite ale TCD care sunt folosite de obicei—noi am folosit versiunea TCD1 pentru compresia imaginilor în secțiunea precedentă
- versiunea TCD4 este cea mai populară pentru compresia audio

Definiția 1

Transformata Cosinus Discretă (versiunea 4) (TCD4) a lui $x = [x_0, \dots, x_{n-1}]^T$ este vectorul n -dimensional

$$y = Ex,$$

unde E este matricea $n \times n$

$$E_{ij} = \sqrt{\frac{2}{n}} \cos \frac{(i + \frac{1}{2})(j + \frac{1}{2})\pi}{n}. \quad (1)$$

- ca în cazul TCD1, matricea E din TCD4 este o matrice reală ortogonală: este pătratică și coloanele ei sunt vectori unitari ortogonali doi căte doi
- de fapt, coloanele lui E sunt vectorii proprii unitari ai matricii reale simetrice $n \times n$

$$\begin{bmatrix} 1 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ & & \ddots & \ddots & \ddots \\ & & & -1 & 2 & -1 \\ & & & & -1 & 3 \end{bmatrix}. \quad (2)$$

- în continuare, observăm două fapte importante despre coloanele matricii TCD4
- presupunem n ca fiind fixat, și considerăm nu doar cele n coloane din TCD4, ci vectorii coloană definiți de (1) pentru orice întreg pozitiv sau negativ j

Lema 1

Notăm prin c_j coloana j a matricii TCD4 (extinse) (1). Atunci (a) $c_j = c_{-1-j}$ pentru orice întreg j (coloanele sunt simetrice în jurul lui $j = -\frac{1}{2}$), și (b) $c_j = -c_{2n-1-j}$ pentru orice întreg j (coloanele sunt antisimetrice în jurul lui $j = n - \frac{1}{2}$).

doar ca sa fie si demonstratia:

- pentru a demonstra partea (a) a lemei, scriem $j = -\frac{1}{2} + (j + \frac{1}{2})$ și $-1 - j = -j + \frac{1}{2} - (j + \frac{1}{2})$
- folosind ecuația (1), obținem

$$\begin{aligned} c_j &= c_{-\frac{1}{2}+(j+\frac{1}{2})} = \sqrt{\frac{2}{n}} \cos \frac{(i+\frac{1}{2})(j+\frac{1}{2})\pi}{n} = \sqrt{\frac{2}{n}} \cos \frac{(i+\frac{1}{2})(-j-\frac{1}{2})\pi}{n} \\ &= c_{-\frac{1}{2}-(j+\frac{1}{2})} = c_{-1-j} \end{aligned}$$

pentru $i = 0, \dots, n-1$

- pentru a demonstra (b), luăm $r = n - \frac{1}{2} - j$; atunci $j = n - \frac{1}{2} - r$ și $2n - 1 - j = n - \frac{1}{2} + r$, și trebuie să arătăm că $c_{n-\frac{1}{2}-r} + c_{n-\frac{1}{2}+r} = 0$
- din formula pentru adunarea unghiurilor la cosinus, avem

$$\begin{aligned} c_{n-\frac{1}{2}-r} &= \sqrt{\frac{2}{n}} \cos \frac{(2i+1)(n-r)\pi}{2n} \\ &= \sqrt{\frac{2}{n}} \cos \frac{2i+1}{2}\pi \cos \frac{(2i+1)r\pi}{2n} + \sqrt{\frac{2}{n}} \sin \frac{2i+1}{2}\pi \sin \frac{(2i+1)r\pi}{2n} \\ c_{n-\frac{1}{2}+r} &= \sqrt{\frac{2}{n}} \cos \frac{(2i+1)(n+r)\pi}{2n} \\ &= \sqrt{\frac{2}{n}} \cos \frac{2i+1}{2}\pi \cos \frac{(2i+1)r\pi}{2n} - \sqrt{\frac{2}{n}} \sin \frac{2i+1}{2}\pi \sin \frac{(2i+1)r\pi}{2n} \end{aligned}$$

- Lema 1 arată că pentru orice întreg j , coloana c_j poate fi exprimată ca una dintre coloanele lui TCD4—și anume, una dintre coloanele c_i pentru $0 \leq i \leq n-1$, după cum se arată în Figura 1, până la o posibilă schimbare de semn

Definiția 2

Fie n un întreg pozitiv par. **Transformata Cosinus Discretă**

Modificată (TCDM) a lui $x = [x_0, \dots, x_{2n-1}]^T$ este vectorul n -dimensional

$$y = Mx, \quad (3)$$

unde M este matricea $n \times 2n$

$$M_{ij} = \sqrt{\frac{2}{n}} \cos \frac{(i+\frac{1}{2})(j+\frac{n}{2}+\frac{1}{2})\pi}{n}, \quad (4)$$

pentru $0 \leq i \leq n-1$ și $0 \leq j \leq 2n-1$.

- observăm diferența majoră față de formele anterioare ale TCD: TCDM a unui vector de dimensiune $2n$ este un vector de dimensiune n
- din acest motiv, TCDM nu este direct inversabilă, dar vom vedea mai târziu că același efect va fi obținut prin suprapunerea vectorilor de dimensiune $2n$
- compararea cu Definiția 1 ne permite să scriem matricea TCDM M în termeni de coloane TCD4 și apoi să simplificăm, folosind Lema 1:

$$\begin{aligned} M &= \left[c_{\frac{n}{2}} \dots c_{\frac{5}{2}n-1} \right] \\ &= \left[c_{\frac{n}{2}} \dots c_{n-1} | c_n \dots c_{\frac{3}{2}n-1} | c_{\frac{3}{2}n} \dots c_{2n-1} | c_{2n} \dots c_{\frac{5}{2}n-1} \right] \\ &= \left[c_{\frac{n}{2}} \dots c_{n-1} | -c_{n-1} \dots -c_{\frac{n}{2}} | -c_{\frac{n}{2}-1} \dots -c_0 | -c_0 \dots -c_{\frac{n}{2}-1} \right]. \quad (5) \end{aligned}$$

- de exemplu, matricea TCDM $n = 4$ este

$$M = [c_2 c_3 | c_4 c_5 | c_6 c_7 | c_8 c_9] = [c_2 c_3 | -c_3 -c_2 | -c_1 -c_0 | -c_0 -c_1].$$

- pentru a simplifica notația, fie A și B jumătatea stângă, și, respectiv, dreapta a matricii TCD4, astfel încât $E = [A|B]$
- definim matricea de permutare formată prin inversarea coloanelor matricii identitate, de la stânga la dreapta:

$$R = \begin{bmatrix} & & 1 \\ & \ddots & \\ 1 & & \end{bmatrix}.$$

- matricea de permutare R inversează coloanele de la dreapta la stânga atunci când este înmulțită cu o matrice la dreapta
- când este înmulțită la stânga, inversează rândurile de sus în jos
- observăm că R este o matrice simetrică ortogonală, deoarece $R^{-1} = R^T = R$
- acum (5) poate fi scrisă mai simplu sub forma

$$M = [B| -BR| -AR| -A], \quad (6)$$

unde AR și BR sunt versiunile lui A și B în care ordinea coloanelor a fost inversată, de la stânga la dreapta

- acțiunea TCDM poate fi exprimată în termeni de TCD4

pentru $i = 0, \dots, n-1$

- deoarece $\cos \frac{1}{2}(2i+1)\pi = 0$ pentru orice întreg i , suma $c_{n-\frac{1}{2}-r} + c_{n-\frac{1}{2}+r} = 0$, ceea ce trebuia să arătăm

...	c_{-4}	c_{-3}	c_{-2}	c_{-1}	c_0	c_1	c_2	c_{n-1}	c_n	c_{2n-1}	c_{2n}	c_{2n+1}	...
...	c_3	c_2	c_1	c_0	c_1	c_2	c_{n-1}	$-c_{n-1}$	$-c_0$	$-c_0$	$-c_1$...	

Figura 1: Ilustrarea Lemei 1. Coloanele c_0, \dots, c_{n-1} formează matricea $n \times n$ TCD4. Pentru întregii j din afara acestui interval, coloana definită de către c_j din ecuația (1) corespunde uneia dintre cele n coloane ale lui TCD4, prezentată direct sub ea în figură. Aceasta ilustrează Lemă 1.

- vom folosi matricea TCD4 E pentru a construi Transformata Cosinus Discretă Modificată
- presupunem că n este par
- vom crea o nouă matrice, folosind coloanele $c_{\frac{n}{2}}, \dots, c_{\frac{5}{2}n-1}$

- fie

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

un vector $2n$ -dimensional, unde fiecare x_i este un vector de dimensiune $n/2$ (ne amintim că n este par)

- atunci, din caracterizarea lui M din (6), avem

$$\begin{aligned} Mx &= Bx_1 - BRx_2 - ARx_3 - Ax_4 \\ &= [A|B] \begin{bmatrix} -Rx_3 - x_4 \\ x_1 - Rx_2 \end{bmatrix} = E \begin{bmatrix} -Rx_3 - x_4 \\ x_1 - Rx_2 \end{bmatrix}, \end{aligned} \quad (7)$$

unde E este matricea TCD4 $n \times n$ și Rx_2 și Rx_3 reprezintă x_2 și x_3 cu intrările inversate de sus în jos

- acest fapt este foarte util—putem exprima ieșirea lui M în funcție de matricea ortogonală E
- deoarece matricea $n \times 2n$ M a TCDM nu este o matrice pătratică, nu este inversabilă
- știm că, deoarece E este o matrice ortogonală,

$$\begin{aligned} A^T A &= I \\ B^T B &= I \\ A^T B = B^T A &= 0, \end{aligned}$$

unde I reprezintă matricea identitate $n \times n$

- acum suntem pregătiți să calculăm NM , pentru a vedea în ce sens N inversează matricea TCDM M
- presupunem că x este partionat în patru părți, ca mai înainte
- potrivit (7) și (9), ortogonalității lui A și B , și faptului că $R^2 = I$, avem

$$\begin{aligned} NM \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} &= \begin{bmatrix} B^T \\ -RB^T \\ -RA^T \\ -A^T \end{bmatrix} [A(-Rx_3 - x_4) + B(x_1 - Rx_2)] \\ &= \begin{bmatrix} x_1 - Rx_2 \\ -Rx_1 + x_2 \\ x_3 + Rx_4 \\ Rx_3 + x_4 \end{bmatrix}. \end{aligned} \quad (10)$$

- în algoritmii de compresie audio, TCDM este aplicată unor vectori de date care se suprapun
- motivul este că orice „artefacte” datorate capetelor vectorilor vor apărea cu o frecvență fixată, datorită dimensiunii constante a vectorilor
- sistemul auditiv este și mai sensibil la erori periodice decât sistemul vizual; la urma urmei, o eroare cu frecvență fixă este un ton al acelei frecvențe, pe care urechea îl va sesiza
- presupunem că datele vor fi prezentate într-o manieră suprapusă
- fie

$$Z_1 = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \text{ și } Z_2 = \begin{bmatrix} x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

două vectori $2n$ -dimensionali pentru un întreg par n , unde fiecare x_i este un vector de dimensiune $n/2$

- vectorii Z_1 și Z_2 se suprapun pe jumătate din lungimea lor
- deoarece (10) ne arată că

$$NMZ_1 = \begin{bmatrix} x_1 - Rx_2 \\ -Rx_1 + x_2 \\ x_3 + Rx_4 \\ Rx_3 + x_4 \end{bmatrix} \text{ și } NMZ_2 = \begin{bmatrix} x_3 - Rx_4 \\ -Rx_3 + x_4 \\ x_5 + Rx_6 \\ Rx_5 + x_6 \end{bmatrix}, \quad (11)$$

putem reconstrui vectorul n -dimensional $[x_3, x_4]^T$ exact prin medierea jumătății inferioare a lui NMZ_1 și a jumătății superioare a lui NMZ_2 :

$$\begin{bmatrix} x_3 \\ x_4 \end{bmatrix} = \frac{1}{2}(NMZ_1)_{n, \dots, 2n-1} + \frac{1}{2}(NMZ_2)_{0, \dots, n-1}. \quad (12)$$

- această egalitate prezintă modul în care matricea N este folosită pentru a decoda semnalul după ce a fost codat prin M
- acest rezultat este rezumat în Teorema 1

Teorema 1 (Inversarea TCDM prin suprapunere)

Fie M matricea TCDM $n \times 2n$, și $N = M^T$. Fie u_1, u_2, u_3 n -vectori, și luăm

$$v_1 = M \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \text{ și } v_2 = M \begin{bmatrix} u_2 \\ u_3 \end{bmatrix}.$$

Atunci n -vectorii w_1, w_2, w_3, w_4 definiți prin

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = Nv_1 \text{ și } \begin{bmatrix} w_3 \\ w_4 \end{bmatrix} = Nv_2,$$

satisfac $u_2 = \frac{1}{2}(w_2 + w_3)$.

- aceasta este o reconstrucție exactă
- Teorema 1 este folosită de obicei pentru un semnal lung $[u_1, u_2, \dots, u_m]^T$ format din n -vectori concatenați
- TCDM este aplicată pentru perechi adiacente pentru a obține semnalul transformat $[v_1, v_2, \dots, v_{m-1}]^T$
- acum intervine compresia cu pierdere de informații
- vectorii v_i sunt componente de frecvență, astfel că putem alege să păstrăm anumite frecvențe și să estompăm alte frecvențe
- vom urma această direcție în secțiunea următoare
- după compresarea conținutului vectorilor v_i prin cuantizare sau alte mijloace, $[u_2, \dots, u_{m-1}]^T$ poate fi decompresat folosind Teorema 1
- observăm că nu putem recupera u_1 și u_m ; ei trebuie să fie sau părți neimportante ale semnalului sau o bordură care este adăugată înainte

Exemplul 1

- folosiți TCDM suprapusă pentru a transforma semnalul $x = [1, 2, 3, 4, 5, 6]^T$
- apoi inversați transformarea pentru a reconstrui secțiunea din mijloc $[3, 4]^T$
- vom suprapune vectorii $[1, 2, 3, 4]^T$ și $[3, 4, 5, 6]^T$
- fie $n = 2$ și luăm

$$E_2 = \begin{bmatrix} \cos \frac{\pi}{8} & \cos \frac{3\pi}{8} \\ \cos \frac{3\pi}{8} & \cos \frac{9\pi}{8} \end{bmatrix} = \begin{bmatrix} b & c \\ c & -b \end{bmatrix}.$$

aplicând TCDM 2×4 , obținem

$$v_1 = M \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix} = E_2 \begin{bmatrix} -R(3) - 4 \\ 1 - R(2) \end{bmatrix} = E_2 \begin{bmatrix} -7 \\ -1 \end{bmatrix} = \begin{bmatrix} -7b - c \\ b - 7c \end{bmatrix} = \begin{bmatrix} -6.8498 \\ -1.7549 \end{bmatrix},$$

$$v_2 = M \begin{bmatrix} 3 \\ 4 \\ 5 \\ 6 \end{bmatrix} = E_2 \begin{bmatrix} -R(5) - 6 \\ 3 - R(4) \end{bmatrix} = E_2 \begin{bmatrix} -11 \\ -1 \end{bmatrix} = \begin{bmatrix} -11b - c \\ b - 11c \end{bmatrix} = \begin{bmatrix} -10.5454 \\ -3.2856 \end{bmatrix}.$$

semnalul transformat este reprezentat prin

$$[v_1 | v_2] = \begin{bmatrix} -6.8498 & -10.5454 \\ -1.7549 & -3.2856 \end{bmatrix}.$$

pentru a inversa TCDM, definim A și B prin

$$E_2 = [A | B] = \begin{bmatrix} b & c \\ c & -b \end{bmatrix},$$

și calculăm

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = Nv_1 = \begin{bmatrix} B^T v_1 \\ -RB^T v_1 \\ -RA^T v_1 \\ -A^T v_1 \end{bmatrix} = \begin{bmatrix} c & -b \\ -c & b \\ -b & -c \\ -b & -c \end{bmatrix} \begin{bmatrix} -7b - c \\ b - 7c \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 7 \\ 7 \end{bmatrix},$$

$$\begin{bmatrix} w_3 \\ w_4 \end{bmatrix} = Nv_2 = \begin{bmatrix} B^T v_2 \\ -RB^T v_2 \\ -RA^T v_2 \\ -A^T v_2 \end{bmatrix} = \begin{bmatrix} c & -b \\ -c & b \\ -b & -c \\ -b & -c \end{bmatrix} \begin{bmatrix} -11b - c \\ b - 11c \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 11 \\ 11 \end{bmatrix},$$

unde am folosit faptul că $b^2 + c^2 = 1$

rezultatul Teoremei 1 ne spune că putem recupera suprapunerea $[3, 4]^T$ folosind expresia

$$u_2 = \frac{1}{2}(w_2 + w_3) = \frac{1}{2} \left(\begin{bmatrix} 7 \\ 7 \end{bmatrix} + \begin{bmatrix} -1 \\ 1 \end{bmatrix} \right) = \begin{bmatrix} 3 \\ 4 \end{bmatrix}.$$

- definiția și utilizarea TCDM sunt mai puțin directe decât folosirea TCD, discutată anterior în acest capitol
- avantajul ei este că permite suprapunerea vectorilor adiacenți în mod eficient
- efectul este de a media contribuțiile a doi vectori, reducând „artefactele” rezultate din tranzitii abrupte care pot fi observate la capete
- ca în cazul TCD, putem filtra sau cuantiza coeficienții transformate înapoi la semnalul original pentru a îmbunătăți sau a compresa semnalul
- în cele ce urmează, vom arăta cum poate fi folosită TCDM pentru compresie prin adăugarea unui pas de cuantizare

9.4.2. Cuantizarea bitilor

- compresia cu pierdere de informații a semnalelor audio se obține prin cuantizarea ieșirii TCDM a semnalului
- în această subsecțiune, vom extinde cuantizarea folosită pentru compresia imaginilor, pentru a permite un control sporit asupra numărului de biți folosiți pentru a reprezenta versiunea compresată a semnalului
- pornim de la intervalul deschis de numere reale $(-L, L)$
- presupunem că scopul este de a reprezenta un număr din $(-L, L)$ prin b biți, și că suntem dispuși să acceptăm o mică eroare
- vom folosi un bit pentru semn și vom cuantiza la un întreg binar pe $b - 1$ biți
- formula de cuantizare este:

Algoritm 1 (Cuantizarea cu b biți a intervalului $(-L, L)$)

Cuantizarea:

$$z = \text{rotunjire} \left(\frac{y}{q} \right), \text{ unde } q = \frac{2L}{2^b - 1}$$

Decuantizarea:

$$\bar{y} = qz \quad (13)$$

- ca exemplu, vom arăta cum se pot reprezenta numerele din intervalul $(-1, 1)$ cu 4 biți
- luăm $q = 2(1)/(2^4 - 1) = 2/15$, și cuantizăm prin q
- numărul $y = -0.3$ este reprezentat prin

$$\frac{-0.3}{2/15} = -\frac{9}{4} \rightarrow -2 \rightarrow -010,$$

și numărul $y = 0.9$ este reprezentat prin

$$\frac{0.9}{2/15} = \frac{27}{4} = 6.75 \rightarrow 7 \rightarrow +111.$$

- decuantizarea inversează acest proces
- versiunea cuantizată a lui -0.3 este decuantizată ca

$$(-2)q = (-2)(2/15) = -4/15 \approx -0.2667,$$

și versiunea cuantizată a lui 0.9 ca

$$(7)q = (7)(2/15) = 14/15 \approx 0.9333.$$

- în ambele cazuri, eroarea de cuantizare este $1/30$

Exemplul 2

- cuantizați ieșirea TCDM din Exemplul 1 cu întregi pe 4 biți
- apoi decuantizați, inversați TCDM, și găsiți eroarea de cuantizare
- toate intrările transformate se află în intervalul $(-12, 12)$
- folosind $L = 12$, cuantizarea cu 4 biți necesită ca $q = 2(12)/(2^4 - 1) = 1.6$
- atunci

$$v_1 = \begin{bmatrix} -6.8498 \\ -1.7549 \end{bmatrix} \rightarrow \begin{bmatrix} \text{rotunjire} \left(\frac{-6.8498}{1.6} \right) \\ \text{rotunjire} \left(\frac{-1.7549}{1.6} \right) \end{bmatrix} \rightarrow \begin{bmatrix} -4 \\ -1 \end{bmatrix} \rightarrow \begin{bmatrix} -100 \\ -001 \end{bmatrix}$$

și

$$v_2 = \begin{bmatrix} -10.5454 \\ -3.2856 \end{bmatrix} \rightarrow \begin{bmatrix} \text{rotunjire} \left(\frac{-10.5454}{1.6} \right) \\ \text{rotunjire} \left(\frac{-3.2856}{1.6} \right) \end{bmatrix} \rightarrow \begin{bmatrix} -7 \\ -2 \end{bmatrix} \rightarrow \begin{bmatrix} -111 \\ -010 \end{bmatrix}.$$

- variabilele transformate v_1, v_2 pot fi stocate ca întregi pe 4 biți, dând un total de 16 biți
- decuantizarea cu $q = 1.6$ este

$$\begin{bmatrix} -4 \\ -1 \end{bmatrix} \rightarrow \begin{bmatrix} -6.4 \\ -1.6 \end{bmatrix} = \overline{v_1}$$

și

$$\begin{bmatrix} -7 \\ -2 \end{bmatrix} \rightarrow \begin{bmatrix} -11.2 \\ -3.2 \end{bmatrix} = \overline{v_2}.$$

- aplicând TCDM inversă, obținem

$$\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = N\overline{v_1} = \begin{bmatrix} -0.9710 \\ 0.9710 \\ 6.5251 \\ 6.5251 \end{bmatrix}$$

$$\begin{bmatrix} w_3 \\ w_4 \end{bmatrix} = N\overline{v_2} = \begin{bmatrix} -1.3296 \\ 1.3296 \\ 11.5720 \\ 11.5720 \end{bmatrix},$$

- și semnalul reconstruit este

$$u_2 = \frac{1}{2}(w_2 + w_3) = \frac{1}{2} \left(\begin{bmatrix} 6.5251 \\ 6.5251 \end{bmatrix} + \begin{bmatrix} -1.3296 \\ 1.3296 \end{bmatrix} \right) = \begin{bmatrix} 2.5977 \\ 3.9274 \end{bmatrix}.$$

- eroarea de cuantizare este diferența dintre semnalul inițial și semnalul reconstruit:

$$\left| \begin{bmatrix} 2.5977 \\ 3.9274 \end{bmatrix} - \begin{bmatrix} 3 \\ 4 \end{bmatrix} \right| = \begin{bmatrix} 0.4023 \\ 0.0726 \end{bmatrix}.$$

10. Valori proprii și valori singulare

- metodele computaționale pentru localizarea valorilor proprii sunt bazate pe ideea fundamentală a iterării de putere, un tip de iterare de punct fix pentru spații proprii
- o versiune sofisticată a acestei idei, numită algoritmul QR, este algoritmul standard pentru determinarea tuturor valorilor proprii ale matricilor tipice
- descompunerea valorilor singulare dezvăluie structura de bază a unei matrice și este foarte folosită în aplicațiile statistice pentru a găsi relații între date
- în acest capitol, trecem în revistă metode pentru calcularea valorilor proprii și vectorilor proprii pentru o matrice pătratică, și a valorilor singulare și a vectorilor singulari pentru o matrice generală

10.1. Metode de tip iterare de putere

>> nu există nicio metodă directă pentru calcularea valorilor proprii

10.1.1. Iterare de putere

- motivarea pentru iterare de putere este aceea că înmulțirea cu o matrice tinde să miște vectorii înspre direcția vectorului propriu dominant

Definiția 3

Fie A o matrice $m \times m$. O **valoare proprie dominantă** a lui A este valoarea proprie λ a cărei normă este mai mare decât toate celelalte valori proprii ale lui A . Dacă există, un vector propriu asociat lui λ se numește **vector propriu dominant**.

- matricea

$$A = \begin{bmatrix} 1 & 3 \\ 2 & 2 \end{bmatrix}$$

are o valoare proprie dominantă 4 cu vectorul propriu $[1, 1]^T$, și o valoare proprie care are normă mai mică, -1, cu vectorul propriu asociat $[-3, 2]^T$

- să observăm care este rezultatul înmulțirii matricii A cu un vector „aleator”, de exemplu $[-5, 5]^T$:

$$\begin{aligned}x_1 &= Ax_0 = \begin{bmatrix} 1 & 3 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} -5 \\ 5 \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \end{bmatrix} \\x_2 &= A^2x_0 = \begin{bmatrix} 1 & 3 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} 10 \\ 0 \end{bmatrix} = \begin{bmatrix} 10 \\ 20 \end{bmatrix} \\x_3 &= A^3x_0 = \begin{bmatrix} 1 & 3 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} 10 \\ 20 \end{bmatrix} = \begin{bmatrix} 70 \\ 60 \end{bmatrix} \\x_4 &= A^4x_0 = \begin{bmatrix} 1 & 3 \\ 2 & 2 \end{bmatrix} \begin{bmatrix} 70 \\ 60 \end{bmatrix} = \begin{bmatrix} 250 \\ 260 \end{bmatrix} = 260 \begin{bmatrix} \frac{25}{26} \\ 1 \end{bmatrix}.\end{aligned}$$

- înmulțirea repetată cu matricea A a unui vector inițial aleator a avut ca efect mișcarea vectorului foarte aproape de vectorul propriu dominant al lui A
- aceasta nu este o coincidență, după cum se poate vedea din exprimarea lui x_0 ca o combinație liniară a vectorilor proprii

$$x_0 = 1 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 2 \begin{bmatrix} -3 \\ 2 \end{bmatrix},$$

$$\det(A - \lambda I_m) = 0$$

$$\Rightarrow \lambda$$

• și ia fiecare $\lambda \Rightarrow \beta_{\lambda} \Rightarrow v_{\lambda}$

$\beta_{\lambda} = \{v_{\lambda}\}$ de la λ

- și refacerea calculului în lumina acestei scrieri:

$$\begin{aligned}x_1 &= Ax_0 = 4 \begin{bmatrix} 1 \\ 1 \end{bmatrix} - 2 \begin{bmatrix} -3 \\ 2 \end{bmatrix} \\x_2 &= A^2 x_0 = 4^2 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 2 \begin{bmatrix} -3 \\ 2 \end{bmatrix} \\x_3 &= A^3 x_0 = 4^3 \begin{bmatrix} 1 \\ 1 \end{bmatrix} - 2 \begin{bmatrix} -3 \\ 2 \end{bmatrix} \\x_4 &= A^4 x_0 = 4^4 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 2 \begin{bmatrix} -3 \\ 2 \end{bmatrix} \\&= 256 \begin{bmatrix} 1 \\ 1 \end{bmatrix} + 2 \begin{bmatrix} -3 \\ 2 \end{bmatrix}.\end{aligned}$$

- ideea este că vectorul propriu corespunzător valorii proprii care este cea mai mare în modul va domina calculul după câțiva pași
- în acest caz, valoarea proprie 4 este cea mai mare, și calculul se apropie din ce în ce mai mult de vectorul propriu corespunzător $[1, 1]^T$
- pentru a nu lăsa numerele să crească necontrolat, este necesar să normalizăm vectorul la fiecare pas

- o modalitate de a face acest lucru este de a împărți vectorul curent prin norma lui înainte de fiecare pas
- aceste două operații, normalizarea și înmulțirea cu A constituie metoda iterăției de putere
- pe măsură ce pașii oferă aproximări îmbunătățite ale vectorilor proprii, cum găsim aproximările valorilor proprii?
- pentru a pune întrebarea mai general, presupunem că o matrice A și o aproximare a unui vector propriu sunt cunoscute
- care este cea mai bună aproximare pentru valoarea proprie asociată?
- vom apela la cele mai mici pătrate
- considerăm ecuația valorii proprii $x\lambda = Ax$, unde x este o aproximare a unui vector propriu și λ este necunoscut
- privită în acest fel, matricea coeficienților este matricea $n \times n$ $x^T x$
- ecuațiile normale ne spun că răspunsul în sensul celor mai mici pătrate este soluția lui $x^T x\lambda = x^T Ax$, sau

$$\lambda = \frac{x^T Ax}{x^T x}, \quad (14)$$

cunoscut sub numele de **cât Rayleigh**

- fiind dată o aproximare a unui vector propriu, câtul Rayleigh este cea mai bună aproximare a valorii proprii corespunzătoare
- aplicând câtul Rayleigh la vectorul propriu normalizat adaugă o aproximare a valorii proprii la iterăția de putere

Algoritmul 2 (Iterația de putere)

Dându-se vectorul inițial x_0

for $j = 1, 2, 3, \dots$

$$u_{j-1} = x_{j-1} / \|x_{j-1}\|_2$$

$$x_j = Au_{j-1}$$

$$\lambda_j = u_{j-1}^T Au_{j-1}$$

end

$$u_j = x_j / \|x_j\|_2$$

- pentru a găsi vectorul propriu dominant al matricii A , începem cu un vector inițial
- fiecare iterăție constă din normalizarea vectorului curent și înmulțirea cu A
- câtul Rayleigh este folosit pentru a approxima valoarea proprie

10.1.2. Convergența iterăției de putere

- vom demonstra convergența iterăției de putere în anumite condiții impuse valorilor proprii
- deși aceste condiții nu sunt absolut generale, ele servesc pentru a arăta de ce metoda înregistrează o reușită în cel mai clar caz posibil
- mai târziu, vom asambla metode de găsire a valorilor proprii din ce în ce mai sofisticate, construite pe baza conceptului de iterăție de putere, care pot fi folosite pentru matrici mai generale

Teorema 2

Fie A o matrice $m \times m$ cu valorile proprii reale $\lambda_1, \dots, \lambda_m$, care satisfac $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_m|$. Presupunem că vectorii proprii ai lui A generează \mathbb{R}^m . Pentru aproape orice vector inițial, iterăția de putere converge liniar către un vector asociat lui λ_1 , cu rata constantă de convergență $S = |\lambda_2/\lambda_1|$.

- fie v_1, \dots, v_n vectorii proprii care formează o bază a lui \mathbb{R}^n , cu valorile proprii corespunzătoare, respectiv, $\lambda_1, \dots, \lambda_n$
- exprimăm vectorul inițial în această bază sub forma

$$x_0 = c_1 v_1 + \dots + c_n v_n, \text{ pentru anumiți coeficienți } c_i$$

- afirmația „pentru aproape orice vector inițial” înseamnă că putem presupune că $c_1, c_2 \neq 0$
- aplicând iterația de putere, obținem

$$\begin{aligned} Ax_0 &= c_1 \lambda_1 v_1 + c_2 \lambda_2 v_2 + \cdots + c_n \lambda_n v_n \\ A^2 x_0 &= c_1 \lambda_1^2 v_1 + c_2 \lambda_2^2 v_2 + \cdots + c_n \lambda_n^2 v_n \\ A^3 x_0 &= c_1 \lambda_1^3 v_1 + c_2 \lambda_2^3 v_2 + \cdots + c_n \lambda_n^3 v_n \\ &\vdots \end{aligned}$$

cu normalizare la fiecare pas

- pe măsură ce numărul de pași $k \rightarrow \infty$, primul termen din partea dreaptă va domina, indiferent de cum este realizată normalizarea, deoarece

$$\frac{A^k x_0}{\lambda_1^k} = c_1 v_1 + c_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k v_2 + \cdots + c_n \left(\frac{\lambda_n}{\lambda_1}\right)^k v_n.$$

- presupunerea că $|\lambda_1| > |\lambda_i|$ pentru $i > 1$ implică faptul că toți termenii mai puțin primul vor converge la zero cu rata de convergență $S \leq |\lambda_2/\lambda_1|$, și anume cu exact această rată, deoarece $c_2 \neq 0$
- prin urmare, metoda converge la un multiplu al vectorului propriu dominant v_1 , cu valoarea proprie λ_1
- termenul „aproape orice” din concluzia teoremei înseamnă că mulțimea de vectori inițiali x_0 pentru care iterația eșuează este o mulțime de dimensiune mai mică din \mathbb{R}^m
- mai exact, iterația va converge cu rata specificată dacă x_0 nu se află în reuniunea planelor de dimensiune $m - 1$ generate de $\{v_1, v_3, \dots, v_m\}$ și $\{v_2, v_3, \dots, v_m\}$

Rata de convergență
 $S \leq |\frac{\lambda_2}{\lambda_1}|$

10.1.3. Iterația de putere inversă

- iterația de putere este limitată la localizarea valorii proprii cu norma cea mai mare
- dacă iterația de putere este aplicată inversei matricii, cea mai mică valoare proprie poate fi găsită

Lema 2

Fie $\lambda_1, \lambda_2, \dots, \lambda_m$ valorile proprii ale matricii $m \times m A$. (a) Valorile proprii ale matricii inverse A^{-1} sunt $\lambda_1^{-1}, \lambda_2^{-1}, \dots, \lambda_m^{-1}$, presupunând că această inversă există. Vectorii proprii sunt aceiași cu cei ai lui A . (b) Valorile proprii ale matricii $A - sI$ sunt $\lambda_1 - s, \lambda_2 - s, \dots, \lambda_m - s$ și vectorii proprii sunt aceiași cu cei ai lui A .

- (a) $Av = \lambda v$ implică $v = \lambda A^{-1} v$, și, prin urmare, $A^{-1} v = (1/\lambda)v$
- observăm că vectorul propriu este neschimbat
- (b) scădem s/v din ambele părți ale lui $Av = \lambda v$
- atunci $(A - sI)v = (\lambda - s)v$ este definiția valorii proprii pentru $(A - sI)$, și din nou același vector propriu poate fi folosit

- potrivit Lemei 2, valoarea proprie cu norma cea mai mare a matricii A^{-1} este inversa valorii proprii cu norma cea mai mică a matricii A
- aplicarea iterației de putere pentru matricea inversă, urmată de inversarea valorii proprii a lui A^{-1} , ne dă valoarea proprie cu norma cea mai mică a matricii A
- pentru a evita calculul explicit al inversei lui A , rescriem aplicarea iterației de putere pentru A^{-1} , și anume,

$$x_{k+1} = A^{-1}x_k, \quad (15)$$

în forma echivalentă

$$Ax_{k+1} = x_k, \quad (16)$$

care este apoi rezolvată pentru a găsi x_{k+1} prin eliminare gaussiană

- acum știm cum să găsim cea mai mare și cea mai mică valoare proprie a unei matrici
- cu alte cuvinte, pentru o matrice 100×100 , suntem 2% gata
- cum găsim celelalte 98 de procente?
- o abordare este sugerată de Lema 2(b)
- putem să facem oricare dintre celelalte valori proprii să fie cea mai mică prin deplasarea lui A cu o valoare apropiată valorii proprii
- dacă se întâmplă să știm că există o valoare proprie în apropierea lui 10 (de exemplu, 10.05), atunci $A - 10I$ are o valoare proprie $\lambda = 0.05$
- dacă este valoarea proprie cu norma cea mai mică a lui $A - 10I$, atunci iterația de putere inversă $x_{k+1} = (A - 10I)^{-1}x_k$ o va localiza
- și anume, iterația de putere inversă va converge la inversul $1/(0.05) = 20$, după care inversăm înapoi la 0.05 și adunăm înapoi deplasarea, obținând 10.05
- acest truc va localiza valoarea proprie care este cea mai mică după deplasare—adică valoarea proprie care se află cel mai aproape de această deplasare; pentru a rezuma, avem

Algoritmul 3 (Iterația de putere inversă)

Dându-se vectorul inițial x_0 și deplasarea s

```

for  $j = 1, 2, 3, \dots$ 
     $u_{j-1} = x_{j-1}/\|x_{j-1}\|_2$ 
    Rezolvăm  $(A - sI)x_j = u_{j-1}$ 
     $\lambda_j = u_{j-1}^T x_j$ 
end
 $u_j = x_j/\|x_j\|_2$ 

```

- pentru a găsi valoarea proprie a lui A care se află cel mai aproape de numărul real s, aplicăm iterația de putere pentru $(A - sI)^{-1}$ pentru a obține valoarea proprie cu norma cea mai mare b a lui $(A - sI)^{-1}$
- iterațiile de putere ar trebui făcute prin eliminare gaussiană pentru $(A - sI)y_{k+1} = x_k$
- atunci $\lambda = b^{-1} + s$ este valoarea proprie a lui A aflată cel mai aproape de s
- vectorul propriu asociat lui λ este dat direct de calcul

Exemplul 3

- presupunem că A este o matrice 5×5 cu valorile proprii $-5, -2, 1/2, 3/2, 4$
- aflați valoarea proprie și rata de convergență așteptată când aplicăm
 - (a) iterația de putere
 - (b) iterația de putere inversă cu deplasarea $s = 0$
 - (c) iterația de putere inversă cu deplasarea $s = 2$
- (a) iterația de putere cu un vector inițial aleator va converge către valoarea proprie cu norma cea mai mare -5 , cu rata de convergență $S = |\lambda_2|/|\lambda_1| = 4/5$

- (b) iterația de putere inversă (fără deplasare) va converge către valoarea proprie cea mai mică, $1/2$, pentru că inversa sa 2 este mai mare decât celelalte inverse $-1/5, -1/2, 2/3$, și $1/4$
- rata de convergență va fi raportul dintre cele mai mari două valori proprii ale matricii inverse, $S = (2/3)/2 = 1/3$
- (c) iterația de putere inversă cu deplasarea $s = 2$ va localiza valoarea proprie cea mai apropiată de 2, care este $3/2$
- motivul este că, după deplasarea valorilor proprii la $-7, -4, -3/2, -1/2$, și 2, cea mai mare dintre inverse este -2
- după inversare obținem $-1/2$ și adunând înapoi deplasarea $s = 2$, obținem $3/2$
- rata de convergență este din nou raportul $(2/3)/2 = 1/3$

10.1.4. Iterația câtului Rayleigh

- câtul Rayleigh poate fi folosit împreună cu iterația de putere inversă
- știm că aceasta converge la vectorul propriu asociat valorii proprii cu cea mai mică distanță până la deplasarea s , și convergența este rapidă dacă distanța este mică
- dacă la un anumit pas am și aproxiarea unei valori proprii, aceasta ar putea fi folosită drept deplasarea s , pentru a accelera convergența
- folosind câtul Rayleigh ca actualizare a deplasării în iterația de putere inversă ne conduce la iterația câtului Rayleigh (ICR)

Algoritm 4 (Iterația câtului Rayleigh)

```
Dându-se vectorul inițial  $x_0$ 
for  $j = 1, 2, 3, \dots$ 
   $u_{j-1} = x_{j-1} / \|x_{j-1}\|_2$ 
   $\lambda_{j-1} = u_{j-1}^T A u_{j-1}$ 
  Rezolvăm  $(A - \lambda_{j-1} I)x_j = u_{j-1}$ 
end
 $u_j = x_j / \|x_j\|_2$ 
```

- câtă vreme iterația de putere inversă converge liniar, iterația câtului Rayleigh este pătratic convergentă pentru valori proprii simple (care nu se repetă) și va converge cubic dacă matricea este simetrică
- aceasta înseamnă că puțini pași sunt necesari pentru a converge la eroarea de rotunjire efectuată în calculator pentru această metodă
- după convergență, matricea $A - \lambda_{j-1} I$ este singulară și nu mai pot fi făcuți alți pași ai metodei
- prin urmare, încercarea și eroarea trebuie folosite pentru a opri iterația chiar înainte ca acest lucru să se întâmple
- observăm că s-a produs o creștere în complexitate pentru ICR
- iterația de putere inversă necesită doar o factorizare LU; dar pentru ICR, fiecare pas necesită o nouă factorizare, deoarece deplasarea s-a modificat
- chiar și așa, iterația câtului Rayleigh este metoda cu convergența cea mai rapidă dintre cele pe care le-am prezentat în această secțiune pentru a găsi câte o valoare proprie
- în secțiunea următoare, vom discuta modalități de a găsi toate valorile proprii ale unei matrici în același calcul
- ideea de bază va rămâne iterația de putere—doar detaliile organizationale vor deveni mai sofisticate

10.2. Algoritmul QR

- scopul acestei secțiuni este de a dezvolta metode pentru găsirea tuturor valorilor proprii deodată
- începem cu o metodă care funcționează pentru matrici simetrice, iar mai apoi o vom suplimenta pentru a funcționa în general
- matricile simetrice sunt cel mai ușor de tratat pentru că valorile lor proprii sunt reale și vectorii lor proprii unitari formează o bază ortonormată a lui \mathbb{R}^m
- aceasta motivează aplicarea iterării de putere cu m vectori în paralel, în care ne vom asigura că vectorii sunt ortogonali doi câte doi

10.2.1. Iterația simultană

- presupunem că începem cu m vectori inițiali ortogonali doi câte doi v_1, \dots, v_m
- după un pas din iterăția de putere aplicată fiecărui vector, nu avem garanția că Av_1, \dots, Av_m sunt ortogonali doi câte doi
- de fapt, după înmulțiri repetitive cu A , toți vor converge către vectorul propriu dominant, potrivit Teoremei 2
- pentru a evita această situație, vom re-ortogonaliza mulțimea de m vectori la fiecare pas
- înmulțirea simultană cu A a celor m vectori se poate scrie eficient sub forma produsului matricial

$$A[v_1 | \dots | v_m].$$

- după cum am văzut în Capitolul 5, pasul de ortogonalizare poate fi privit ca factorizarea produsului rezultat sub forma QR
- dacă vectorii bazei elementare sunt folosiți ca vectori inițiali, atunci primul pas al iterăției de putere urmată de re-ortogonalizare este $AI = \overline{Q}_1 R_1$, sau

$$\left[\begin{array}{c|c|c|c} A & \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} & A & \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} & \cdots & A & \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \end{array} \right] = \left[\begin{array}{c|c|c|c} \overline{q}_1^1 & \dots & \overline{q}_m^1 \end{array} \right] \left[\begin{array}{cccc} r_{11}^1 & r_{12}^1 & \cdots & r_{1m}^1 \\ r_{22}^1 & & & \vdots \\ \ddots & & & \vdots \\ & & & r_{mm}^1 \end{array} \right]. \quad (17)$$

- vectorii \overline{q}_i^1 pentru $i = 1, \dots, m$ reprezintă noua mulțime ortogonală de vectori unitari din cadrul iterăției de putere
- în continuare, repetăm acest pas:

$$\begin{aligned} \overline{AQ}_1 &= \left[\overline{Aq}_1^1 | \overline{Aq}_2^1 | \dots | \overline{Aq}_m^1 \right] \\ &= \left[\overline{q}_1^2 | \overline{q}_2^2 | \dots | \overline{q}_m^2 \right] \left[\begin{array}{cccc} r_{11}^2 & r_{12}^2 & \cdots & r_{1m}^2 \\ r_{22}^2 & & & \vdots \\ \ddots & & & \vdots \\ & & & r_{mm}^2 \end{array} \right] \\ &= \overline{Q}_2 R_2. \end{aligned} \quad (18)$$

- cu alte cuvinte, am dezvoltat o formă matricială a iterăției de putere care caută să găsească toți cei m vectori proprii ai unei matrici simetrice simultan

Algoritm 5 (Iterația simultană normalizată)

```

Luăm  $\overline{Q}_0 = I$ 
for  $j = 1, 2, 3, \dots$ 
     $\overline{AQ}_j = \overline{Q}_{j+1} R_{j+1}$ 
end

```

- la pasul j , coloanele lui Q_j sunt aproximări ale vectorilor proprii ai lui A , și intrările diagonale $r_{11}^j, \dots, r_{mm}^j$ sunt aproximări ale valorilor proprii
- vom numi acest algoritm iterăția simultană normalizată (ISN)
- există o modalitate și mai compactă de a implementa iterăția simultană normalizată
- luăm $\overline{Q}_0 = I$

- atunci ISN are loc după cum urmează:

$$\begin{aligned} \overline{AQ_0} &= \overline{Q_1 R_1} \\ \overline{AQ_1} &= \overline{Q_2 R_2} \\ \overline{AQ_2} &= \overline{Q_3 R_3} \\ &\vdots \end{aligned} \tag{19}$$

- considerăm iterată similară $Q_0 = I$, și

$$\begin{aligned} A_0 &\equiv AQ_0 = Q_1 R'_1 \\ A_1 &\equiv R_1 Q_1 = Q_2 R'_2 \\ A_2 &\equiv R_2 Q_2 = Q_3 R'_3 \\ &\vdots \end{aligned} \tag{20}$$

pe care îl vom numi **algoritmul QR nedeplasat**

- singura diferență este că matricea A nu mai este necesară după primul pas; ea este înlocuită de matricea curentă R_k
- comparând (19) și (20) ne arată că putem alege $Q_1 = \overline{Q_1}$ și $R_1 = R'_1$ în (19)
- mai mult, deoarece

$$\overline{Q_2 R_2} = \overline{AQ_1} = Q_1 R'_1 \overline{Q_1} = Q_1 R'_1 Q_1 = Q_1 Q_2 R'_2, \tag{21}$$

putem alege $\overline{Q_2} = Q_1 Q_2$ și $R_2 = R'_2$ în (19)

- dacă am ales $Q_{k-1} = Q_1 \cdots Q_{k-1}$ și $R_{j-1} = R'_{j-1}$, atunci

$$\begin{aligned} \overline{Q_j R_j} &= \overline{AQ_{j-1}} = AQ_1 \cdots Q_{j-1} \\ &= \overline{Q_2 R_2 Q_3 \cdots Q_{j-1}} \\ &= \overline{Q_2 Q_3 R_3 Q_3 \cdots Q_{j-1}} \\ &= Q_1 Q_2 Q_3 Q_4 R_4 Q_4 \cdots Q_{j-1} \\ &= \cdots = Q_1 \cdots Q_j R_j, \end{aligned} \tag{22}$$

și putem defini $\overline{Q_j} = Q_1 \cdots Q_j$ și $R_j = R'_j$ în (19)

- prin urmare, algoritmul QR nedeplasat efectuează aceleasi calcule ca iterată simultană normalizată, cu o notație ușor diferită
- observăm de asemenea că

$$A_{j-1} = Q_j R_j = Q_j R_j Q_j Q_j^T = Q_j A_j Q_j^T, \tag{23}$$

astfel că toate matricile A_j sunt asemenea și au aceeași mulțime de valori proprii

Teorema 3

Presupunem că A este o matrice simetrică $m \times m$ cu valorile proprii λ_i care satisfac $|\lambda_1| > |\lambda_2| > \dots > |\lambda_m|$. Algoritmul QR nedeplasat converge liniar către vectorii proprii și valorile proprii ale lui A . Pe măsură ce $j \rightarrow \infty$, A_j converge către o matrice diagonală care conține valorile proprii pe diagonala principală și $\overline{Q_j} = Q_1 \cdots Q_j$ converge la o matrice ortogonală ale cărei coloane sunt vectorii proprii.

- iterata simultană normalizată, practic același algoritm, converge în aceleasi condiții
- observăm că algoritmul QR nedeplasat ar putea să eșueze chiar și pentru matrici simetrice dacă ipotezele teoremei nu sunt satisfăcute
- deși algoritmul QR nedeplasat este o versiune îmbunătățită a iterării de putere, condițiile cerute de Teorema 3 sunt stricte, și câteva îmbunătățiri sunt necesare pentru a face acest algoritm de găsire a valorilor proprii să funcționeze într-un context mai general—de exemplu, în cazul matricilor nesimetrice
- o problemă, care apare și pentru matrici simetrice, este că algoritmul QR nedeplasat nu funcționează garantat în situația unui caz de egalitate pentru vectorul propriu dominant
- un exemplu pentru această situație este

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

care are valorile proprii 1 și -1

- o altă formă de „egalitate” are loc când valorile proprii sunt complexe
- valorile proprii ale matricii nesimetrice

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

sunt i și $-i$, ambele de modul complex 1

- nimic din definirea algoritmului QR nedeplasat nu ne permite să calculăm valori proprii complexe
- mai mult, algoritmul QR nedeplasat nu folosește trucul iterării de putere inverse
- am arătat că iterata de putere poate fi accelerată semnificativ folosind acest truc, și dorim să găsim o modalitate de a aplica această idee noi noastre implementări
- aceste rafinări sunt aplicate în cele ce urmează, după introducerea scopului algoritmului QR, care este acela de a reduce matricea A la forma ei Schur reală

10.2.2. Forma Schur reală și algoritmul QR

- modalitatea în care algoritmul QR găsește valorile proprii ale matricii A este de a găsi o matrice asemenea ei, ale cărei valori proprii sunt evidente
- un exemplu de astfel de matrice este forma Schur reală

Definiția 4

O matrice T are **forma Schur reală** dacă este superior triangulară, cu posibila excepție a unor blocuri de dimensiune 2×2 de pe diagonala principală.

- de exemplu, o matrice de forma

$$\begin{bmatrix} x & x & x & x & x \\ & x & x & x & x \\ & & x & x & x \\ & & & x & x \\ & & & & x \end{bmatrix}$$

are forma Schur reală

- valorile proprii ale unei matrici în această formă sunt valorile proprii ale blocului diagonal—intrările diagonale când blocul este de dimensiune 1×1 , sau valorile proprii ale blocului de dimensiune 2×2 în cazul respectiv
- în orice caz, valorile proprii ale matricii pot fi calculate ușor
- valoarea acestei definiții este dată de faptul că fiecare matrice pătratică cu intrări reale este asemenea cu o matrice de această formă
- aceasta este concluzia următoarei teoreme:

Teorema 4

Fie A o matrice pătratică având intrări reale. Atunci există o matrice ortogonală Q și o matrice T care are forma Schur reală, astfel încât $A = Q^T T Q$.

- asa-numita factorizare Schur a matricii A este o factorizare care pune în evidență valorile proprii, ceea ce înseamnă că dacă o putem efectua, vom cunoaște valorile proprii și vectorii proprii
- algoritmul QR complet mută iterativ o matrice arbitrară A înspre factorizarea sa Schur folosind o serie de transformări de asemănare
- vom folosi ideea iterăției de putere inverse cu deplasări și vom adăuga ideea deflației pentru a dezvolta algoritmul QR deplasat
- versiunea deplasată este ușor de scris
- fiecare pas constă din aplicarea deplasării, efectuarea unei factorizări QR, și apoi efectuarea unei deplasări înapoi
- mai precis,

$$\begin{aligned} A_0 - sI &= Q_1 R_1 \\ A_1 &= R_1 Q_1 + sI. \end{aligned} \tag{24}$$

- observăm că

$$\begin{aligned} A_1 - sI &= R_1 Q_1 \\ &= Q_1^T (A_0 - sI) Q_1 \\ &= Q_1^T A_0 Q_1 - sI \end{aligned}$$

implică faptul că A_1 este asemenea cu A_0 și astfel are aceleași valori proprii

- repetăm acest pas, generând un sir A_k de matrici, toate asemenea cu $A = A_0$
- care sunt alegerile bune pentru deplasarea s ?
- această întrebare ne conduce la conceptul de **deflație** pentru calculul valorilor proprii
- vom alege deplasarea ca fiind intrarea din dreapta jos a matricii A_k
- aceasta va face ca iterăția, pe măsură ce converge la forma Schur reală, să schimbe rândul de jos într-un rând de zerouri, cu excepția intrării din dreapta jos
- după ce această intrare a devenit convergentă la o valoare proprie, vom efectua o deflație a matricii prin eliminarea ultimului rând și a ultimei coloane
- apoi continuăm cu găsirea celorlalte valori proprii

10.3. Descompunerea valorilor singulare

- imaginea sferei unitate din \mathbb{R}^m printr-o matrice $m \times m$ este un elipsoid
- acest fapt interesant evidențiază descompunerea valorilor singulare, care are multe aplicații în analiza matricială în general și în special pentru compresie

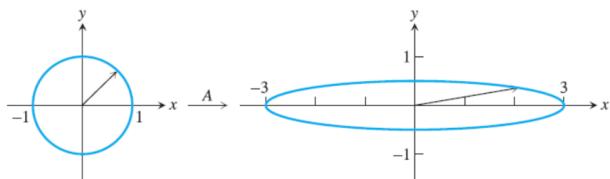


Figura 2: Imaginea cercului unitate printr-o matrice 2×2 . Cercul unitate din \mathbb{R}^2 este transformat în elipsa cu axe semimajore $(3, 0)$ și $(0, 1/2)$ prin matricea A din (25).

- Figura 2 este o ilustrare a elipsei care corespunde matricii

$$A = \begin{bmatrix} 3 & 0 \\ 0 & \frac{1}{2} \end{bmatrix}. \quad (25)$$

- în Figura 2, ne gândim la vectorul v corespunzător fiecărui punct de pe cercul unitate, înmulțit cu A , și apoi reprezentăm capătul vectorului rezultat Av
- rezultatul final este elipsa prezentată în figură
- pentru a descrie elipsa, putem folosi o mulțime ortonormală de vectori pentru a defini baza unui sistem de coordonate

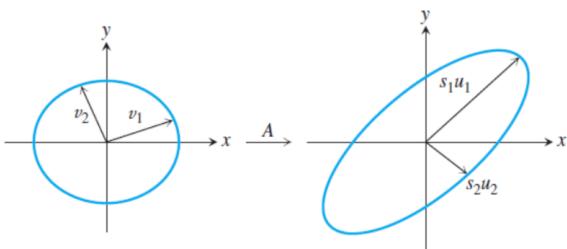


Figura 3: Elipsa asociată cu o matrice. Orice matrice $2 \times 2 A$ poate fi privită din următoarea perspectivă simplă: există un sistem de coordonate $\{v_1, v_2\}$ pentru care A trimite $v_1 \rightarrow s_1 u_1$ și $v_2 \rightarrow s_2 u_2$, unde $\{u_1, u_2\}$ este un alt sistem de coordonate și s_1, s_2 sunt numere nenegative. Această figură se extinde la \mathbb{R}^m pentru o matrice $m \times m$.

- vom vedea în Teorema 5 că pentru orice matrice $m \times n A$, există mulțimile ortonormale $\{u_1, \dots, u_m\}$ și $\{v_1, \dots, v_n\}$, împreună cu numerele nenegative $s_1 \geq \dots \geq s_n \geq 0$, care satisfac

$$\begin{aligned} Av_1 &= s_1 u_1 \\ Av_2 &= s_2 u_2 \\ &\vdots \\ Av_n &= s_n u_n. \end{aligned} \quad (26)$$

- vectorii sunt prezenți în Figura 3
- vectorii v_i se numesc **vectorii singulari drepti** ai matricii A , vectorii u_i sunt **vectorii singulari stângi** ai lui A , și numerele s_i sunt **valorile singulare** ale lui A
- terminologia pentru acești vectori este puțin ciudată, dar motivele vor deveni clare în cele ce urmează
- acest fapt folosit explică de ce o matrice 2×2 transformă cercul unitate într-o elipsă
- putem să ne gândim la vectorii v_i ca fiind baza unui sistem de coordonate rectangular pe care A acționează într-un mod simplu: produce vectorii bazei unui nou sistem de coordonate, și anume vectorii u_i , cu anumite deformări cuantificate de scalarii s_i
- vectorii bazei deformată $s_i u_i$ sunt axele semimajore ale elipsei, după cum se arată în Figura 3

Exemplul 4

- găsiți valorile singulare și vectorii singulari pentru matricea (25) reprezentată în Figura 2
- evident, matricea mărește cu un factor de 3 în direcția x și micșorează cu un factor de $1/2$ în direcția y
- vectorii singulari și valorile singulare ale lui A sunt

$$\begin{aligned} A \begin{bmatrix} 1 \\ 0 \end{bmatrix} &= 3 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\ A \begin{bmatrix} 0 \\ 1 \end{bmatrix} &= \frac{1}{2} \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \end{aligned} \quad (27)$$

- vectorii $3[1, 0]^T$ și $\frac{1}{2}[0, 1]^T$ formează axele semimajore ale elipsei
- vectorii singulari drepti sunt $[1, 0]^T$, $[0, 1]^T$, și vectorii singulari stângi sunt $[1, 0]^T$, $[0, 1]^T$
- valorile singulare sunt 3 și $1/2$

Exemplul 5

- găsiți valorile singulare și vectorii singulari ai matricii

$$A = \begin{bmatrix} 0 & -\frac{1}{2} \\ 3 & 0 \\ 0 & 0 \end{bmatrix}.$$

- aceasta este o ușoară variație a Exemplului 4
- matricea interschimbă axele x și y , schimbând și scara, și adaugă o axă z , de-a lungul căreia nu se întâmplă nimic
- vectorii singulari și valorile singulare ale lui A sunt

$$\begin{aligned} Av_1 &= A \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 3 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = s_1 u_1 \\ Av_2 &= A \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} = s_2 u_2. \end{aligned} \quad (28)$$

- vectorii singulari drepti sunt $[1, 0]^T$, $[0, 1]^T$, vectorii singulari stângi sunt $[0, 1, 0]^T$, $[-1, 0, 0]^T$
- valorile singulare sunt 3, $1/2$
- observăm că trebuie întotdeauna ca s_i să fie un număr nenegativ, și orice semn contrar este absorbit în vectorii u_i și v_i

- există o modalitate standard de a ține evidența acestor informații, într-o factorizare matricială a matricii $m \times n A$
- formăm o matrice $m \times m U$ ale cărei coloane sunt vectorii singulari stângi u_i , o matrice $n \times n V$ ale cărei coloane sunt vectorii singulari drepti v_i , și o matrice diagonală $m \times n S$ ale cărei intrări diagonale sunt valorile singulare s_i
- atunci **descompunerea valorilor singulare** (DVS) a matricii $m \times n A$ este

$$A = USV^T. \quad (29)$$

- Exemplul 5 are reprezentarea DVS

$$\begin{bmatrix} 0 & -\frac{1}{2} \\ 3 & 0 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ 0 & \frac{1}{2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (30)$$

- deoarece U și V sunt matrici pătratice cu coloane ortonormale, ele sunt matrici ortogonale
- observăm că a trebuit să adăugăm o treia coloană u_3 la U pentru a completa baza lui \mathbb{R}^3
- în final, terminologia poate fi acum explicată
- vectorii $u_i(v_i)$ sunt vectorii singulari stângi (drepti) deoarece aceștia apar în partea respectivă din reprezentarea matricială (29)

10.3.1. Gasirea DVS in general

- am arătat două exemple simple de DVS
- pentru a arăta că DVS există pentru o matrice generală A , avem nevoie de următoarea lemă:

Lema 3

Fie A o matrice $m \times n$. Valorile proprii ale lui $A^T A$ sunt nenegative.

- fie v un vector propriu unitar al lui $A^T A$, și $A^T A v = \lambda v$
- atunci

$$0 \leq \|Av\|^2 = v^T A^T A v = \lambda v^T v = \lambda.$$

- pentru o matrice $m \times n A$, matricea $n \times n A^T A$ este simetrică, astfel că vectorii ei proprii sunt ortogonali și valorile ei proprii sunt reale
- Lema 3 ne arată că valorile proprii sunt numere reale nenegative astfel că pot fi exprimate sub forma $s_1^2 \geq \dots \geq s_n^2$, unde mulțimea ortonormală corespunzătoare de vectori proprii este $\{v_1, \dots, v_n\}$
- aceasta ne dă deja două treimi din DVS
- folosim următoarea procedură pentru a găsi u_i pentru $1 \leq i \leq m$:
 - dacă $s_i \neq 0$, definim u_i prin ecuația $s_i u_i = Av_i$
 - dacă $s_i = 0$, alegem u_i ca fiind un vector unitar arbitrar care trebuie să fie ortogonal pe u_1, \dots, u_{i-1}
- se poate verifica faptul că această alegere implică ortogonalitatea vectorilor unitari u_1, \dots, u_m , care formează astfel o altă bază ortonormată a lui \mathbb{R}^m
- de fapt, u_1, \dots, u_m formează o mulțime ortonormată de vectori proprii ai lui AA^T ; rezumând, am demonstrat următoarea Teoremă:

Teorema 5

Fie A o matrice $m \times n$. Atunci există două baze ortonormate $\{v_1, \dots, v_n\}$ a lui \mathbb{R}^n , și $\{u_1, \dots, u_m\}$ a lui \mathbb{R}^m , și numerele reale $s_1 \geq \dots \geq s_n \geq 0$ astfel încât $Av_i = s_i u_i$ pentru $1 \leq i \leq \min\{m, n\}$. Coloanele lui $V = [v_1 | \dots | v_n]$, vectorii singulari drepti, reprezintă mulțimea de vectori proprii ortonormali ai lui $A^T A$; coloanele lui $U = [u_1 | \dots | u_m]$, vectorii singulari stângi, reprezintă mulțimea de vectori proprii ortonormali ai lui AA^T .

- DVS nu este unică pentru o matrice dată A
- în ecuația de definiție $Av_1 = s_1 u_1$, de exemplu, înlocuind pe v_1 cu $-v_1$ și pe u_1 cu $-u_1$ nu schimbă egalitatea, dar schimbă matricile U și V
- concluzionăm din această teoremă că imaginea sferei unitate de vectori este un elipsoid de vectori, centrat în origine, cu axe semimajore $s_i u_i$
- Figura 3 arată că cercul unitate de vectori este transformat într-o elipsă cu axele $\{s_1 u_1, s_2 u_2\}$
- pentru a găsi unde merge Ax pentru un vector x , putem scrie $x = a_1 v_1 + a_2 v_2$ (unde $a_1 v_1$ ($a_2 v_2$) este proiecția lui x pe direcția v_1 (v_2)), și apoi $Ax = a_1 s_1 u_1 + a_2 s_2 u_2$
- reprezentarea matricială (29) rezultă direct din Teorema 5
- definim S ca fiind matricea diagonală $m \times n$ ale cărei intrări sunt $s_1 \geq \dots \geq s_{\min\{m, n\}} \geq 0$
- definim U ca fiind matricea ale cărei coloane sunt u_1, \dots, u_m , și V ca fiind matricea ale cărei coloane sunt v_1, \dots, v_n
- observăm că $USV^T v_i = s_i u_i$ pentru $i = 1, \dots, m$
- deoarece matricile A și USV^T sunt egale de-a lungul vectorilor bazei v_1, \dots, v_n , ele sunt matrici $m \times n$ identice

Exemplul 6

- găsiți valorile singulare și vectorii singulari ai matricii 2×2

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix}. \quad (31)$$

- valorile proprii ale lui

$$A^T A = \begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix},$$

aranjate în ordine descrescătoare, sunt $v_1 = [0, 1]^T$, $s_1^2 = 2$; și $v_2 = [1, 0]^T$, $s_2^2 = 0$

- valorile singulare sunt $\sqrt{2}$ și 0
- potrivit procedurii anterioare, u_1 este definit prin

$$\begin{aligned} \sqrt{2}u_1 &= Av_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \\ u_1 &= \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}, \end{aligned}$$

- și $u_2 = [1/\sqrt{2}, -1/\sqrt{2}]^T$ este ales pentru a fi ortogonal pe u_1
- DVS este

$$\begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} \sqrt{2}/2 & \sqrt{2}/2 \\ -\sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \quad (32)$$

- conform comentariului despre neunicitatea DVS care a urmat Teoremei 5, o altă DVS perfect valabilă pentru această matrice este

$$\begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} -\sqrt{2}/2 & \sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \quad (33)$$

- imaginea cercului unitate prin A este segmentul de dreapta $y[1, -1]^T$, unde y se află între -1 și 1
- deci acțiunea lui A este de a aplatiza cercul unitate într-o elipsă uni-dimensională cu axele semimajore $\sqrt{2}[\sqrt{2}/2, -\sqrt{2}/2]$ și 0

10.3.2. Gasirea DVS în cazul matricilor simetrice

- găsirea DVS a unei matrici $m \times m$ simetrice se reduce la găsirea vectorilor proprii și a valorilor proprii
- știm că există o mulțime ortonormală de vectori proprii
- deoarece vectorii proprii sunt transformați în ei însăși (cu o scalare λ , care este valoare proprie), satisfacerea ecuației (26) este ușoară: doar ordonăm valorile proprii descrescător în funcție de norme

$$|\lambda_1| \geq |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_m|, \quad (34)$$

și le folosim pe post de valori singulare $s_1 \geq s_2 \geq \dots$

- pentru v_i , folosim vectorii proprii unitari în ordinea corespunzătoare valorilor proprii din (34), și folosim

$$u_i = \begin{cases} +v_i & \text{dacă } \lambda_i \geq 0 \\ -v_i & \text{dacă } \lambda_i < 0 \end{cases}. \quad (35)$$

- schimbarea de semn din (35) compensează orice semn minus pierdut prin luarea valorilor absolute în (34)

Exemplul 7

- găsiți valorile singulare și vectorii singulari ai matricii

$$A = \begin{bmatrix} 0 & 1 \\ 1 & \frac{3}{2} \end{bmatrix}. \quad (36)$$

- perechile valoare proprie/vector propriu sunt 2, $[1, 2]^T$ și $-\frac{1}{2}$, $[-2, 1]^T$
- definim v_i din vectorii proprii unitari și u_i din (35):

$$\begin{aligned} Av_1 &= A \begin{bmatrix} \frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix} = 2 \begin{bmatrix} \frac{1}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} \end{bmatrix} = s_1 u_1 \\ Av_2 &= A \begin{bmatrix} \frac{2}{\sqrt{5}} \\ -\frac{1}{\sqrt{5}} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \frac{2}{\sqrt{5}} \\ -\frac{1}{\sqrt{5}} \end{bmatrix} = s_2 u_2. \end{aligned} \quad (37)$$

- DVS este

$$\begin{bmatrix} 0 & 1 \\ 1 & \frac{3}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{5}} & -\frac{2}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{2}{\sqrt{5}} \\ \frac{2}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \end{bmatrix}. \quad (38)$$

- observăm că a trebuit să schimbăm semnul pentru a defini u_2 , după cum cere (35)