

# Laborator 6

## Descompunerea valorilor singulare

### 6.1 Preliminarii

#### 6.1.1 Descompunerea valorilor singulare

Descompunerea valorilor singulare (DVS) a unei matrice este descrisă de următoarea teoremă.

**Teorema 6.1** Pentru orice matrice  $A \in \mathbb{R}^{m \times n}$  există matricele ortogonale  $U \in \mathbb{R}^{m \times m}$  și  $V \in \mathbb{R}^{n \times n}$  astfel încât

$$U^T A V = \Sigma = \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix}, \quad (6.1)$$

unde

$$\Sigma_1 = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_r) \in \mathbb{R}^{r \times r}, \quad (6.2)$$

cu

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0. \quad (6.3)$$

Numerele pozitive  $\sigma_i$ ,  $i = 1 : p$ ,  $p = \min(m, n)$ , ordonate descrescător

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0, \quad \sigma_{r+1} = \sigma_{r+2} = \dots = \sigma_p = 0 \quad (6.4)$$

se numesc *valori singulare* ale matricei  $A$ . Coloanele  $u_j \in \mathbb{R}^m$  ale matricei ortogonale  $U$  se numesc *vectori singulari* stânga ai matricei  $A$ . Coloanele  $v_j \in \mathbb{R}^n$  ale matricei ortogonale  $V$  se numesc *vectori singulari* dreapta ai matricei  $A$ . Relația (6.1) poate fi scrisă alternativ

$$A = U \Sigma V^T = \sum_{j=1}^r \sigma_j u_j v_j^T. \quad (6.5)$$

DVS a unei matrice scoate în evidență numeroase aspecte structurale ale matricei respective și are multiple valențe aplicative.

**Teorema 6.2** Valorile singulare nenule  $\sigma_i$ ,  $i = 1 : r$ , ale unei matrice  $A \in \mathbb{R}^{m \times n}$  sunt rădăcinile pătrate pozitive ale valorilor proprii nenule ale matricei simetrice pozitiv semidefinite

$$B = A^T A \quad (6.6)$$

i.e., dacă  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r > 0$ , sunt cele  $r$  valori proprii nenule ale lui  $B$  atunci

$$\sigma_i = \sqrt{\lambda_i}, \quad i = 1 : r. \quad (6.7)$$

Teorema 6.2 sugerează calculul valorilor singulare ale lui  $A$  prin calculul valorilor proprii ale matricei  $B = A^T A$ . Aceasta procedură, utilă pentru matrice mici, nu este recomandată în general datorită posibilei rele condiționări numerice a matricei  $B$ .

### 6.1.2 Utilizări ale DVS

Prezentăm succint câteva aplicații ale DVS. Pentru detalii și alte aplicații recomandăm consultarea cursului.

#### Calculul rangului unei matrice

Fie  $A = U\Sigma V^T$  DVS a matricei  $A \in \mathbb{R}^{m \times n}$ . Întrucât înmulțirea cu matrice nesingulare nu modifică rangul unei matrice, rangul lui  $A$  este rangul lui  $\Sigma$ . Prin urmare, rangul unei matrice este dat de numărul valorilor sale singulare nenule.

În general, din cauza erorilor de rotunjire, valorile proprii *calculate*  $\hat{\sigma}_i$  vor fi toate nenule, i.e.

$$\hat{\sigma}_1 \geq \hat{\sigma}_2 \geq \dots \geq \hat{\sigma}_p > 0. \quad (6.8)$$

De aceea în practica numerică se utilizează conceptul de *rang numeric*. Rangul numeric se obține neglijând valorile singulare mici, e.g. inferioare unei toleranțe precizate  $\varepsilon$

$$\hat{\sigma}_i \leq \varepsilon. \quad (6.9)$$

Dacă indicele  $i = r + 1$  este primul pentru care (6.9) este satisfăcută, atunci *rangul numeric* al matricei  $A$  este

$$\text{rang}(A, \varepsilon) = r. \quad (6.10)$$

#### Aproximarea de rang mic a unei matrice

Dându-se o matrice  $A \in \mathbb{R}^{m \times n}$ , ne punem problema aproximării ei cu o matrice  $A_k$  cu rang  $k$ . Mai mult, vrem ca aproximația să fie optimă, adică  $\|A - A_k\|$  să fie minimă. DVS furnizează direct această aproximație prin simpla trunchiere la cele mai mari  $k$  valori singulare:

$$A_k = \sum_{j=1}^k \sigma_j u_j v_j^T = U(:, 1 : k) \Sigma(1 : k, 1 : k) V(:, 1 : k)^T. \quad (6.11)$$

Aproximațiile de rang mic sunt utile pentru simplificarea (comprimarea) unor reprezentări sau pentru a găsi subspații reprezentative de dimensiune mică (analiza componentelor principale, etc.).

### Rezolvarea problemei generale CMMP

Problema generală CMMP constă în rezolvarea sistemelor liniare (determinate, supradeterminate sau subdeterminate)

$$Ax = b \quad (6.12)$$

cu matricea  $A \in \mathbb{R}^{m \times n}$  de rang posibil nemaximal. În general, în cazul

$$\text{rang}A < \min(m, n) \quad (6.13)$$

sistemul linear (6.12) nu are soluții exacte dar are un număr infinit de soluții în sens CMMP. Problema este calculul soluției cu normă euclidiană minimă

$$\|x^*\| = \min_{\text{argmin}_{x \in \mathbb{R}^n} \|b - Ax\|} \|x\|. \quad (6.14)$$

Soluția este dată de următoarea teoremă.

**Teorema 6.3** *Fie  $A \in \mathbb{R}^{m \times n}$  cu  $\text{rang}A = r \leq \min(m, n)$ . Dacă  $U^T A V = \Sigma$  este DVS a lui  $A$  atunci soluția de normă euclidiană minimă pentru (6.12) este dată de*

$$x^* = \sum_{j=1}^r \frac{u_j^T b}{\sigma_j} v_j \quad (6.15)$$

## 6.2 Aplicații ale DVS

### 6.2.1 Compresie de imagini

O imagine în nuanțe de gri poate fi asimilată cu o matrice  $A \in \mathbb{R}^{m \times n}$ , fiecărui element al matricei corespunzându-i un pixel al imaginii. O metodă primară de compresie (inexactă) a imaginilor se bazează pe simpla aplicare a unei DVS trunchiate ca în (6.11), adică utilizând aproximarea de rang  $k$  optimă, prin care valorile singulare mici sunt ignorate. (Atragem atenția că există metode mult mai eficiente de compresie de imagine, dar și mult mai sofisticate.)

Calculând DVS și apoi (6.11) pentru un  $k$  bine ales se poate constata că imaginea  $A_k$  este vizual aproape de  $A$ , dar poate fi descrisă de mult mai puține valori. Dacă, de exemplu, matricea  $A$  este  $512 \times 512$  și constatăm că sunt suficiente  $k = 50$  de valori singulare pentru aproximare, avem nevoie  $2 \cdot 512 \cdot 50 + 50$  valori pentru a reface imaginea aproximată; aceste valori sunt primele  $k$  coloane ale matricelor  $U$  și  $V$ , la care se adaugă valorile singulare. Putem reduce de aproximativ 5 ori memoria necesară pentru imagine.

### 6.2.2 Regresie liniară

Modelele liniare sunt foarte utilizate pentru descrierea unor relații între mărimi fizice. Pentru a simplifica discuția, ne mărginim la cazul în care sunt doar două mărimi,  $\xi$  și  $\eta$ , din care dispunem de  $m$  perechi de măsurători  $(\xi_i, \eta_i)$ ,  $i = 1 : m$  (numite și eşantioane), pe baza cărora trebuie să determinăm modelul liniar (numit deseori model de regresie). Studiem două situații.

1. Relația dintre cele două mărimi este

$$\eta = f(\xi) = \alpha\xi + \gamma, \quad (6.16)$$

adică  $\eta$  este o funcție afină de  $\xi$ . Coeficienții  $\alpha$  și  $\gamma$  sunt necunoscuți și trebuie determinați din măsurătorile disponibile. Presupunem că valorile  $\xi_i$  sunt exacte, dar valorile  $\eta_i$  sunt afectate de zgomot. Putem calcula coeficienții din (6.16) minimizând norma erorii

$$\sum_{i=1}^m (\eta_i - \alpha\xi_i - \gamma)^2 = \left\| \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_m \end{bmatrix} - \begin{bmatrix} \xi_1 & 1 \\ \xi_2 & 1 \\ \vdots & \vdots \\ \xi_m & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \gamma \end{bmatrix} \right\|^2 = \|b - Ax\|^2. \quad (6.17)$$

Această problemă se rezolvă cu algoritmul CMMP studiat în Laboratorul 4.

2. Modelul este

$$g(\xi, \eta) = \alpha\xi + \beta\eta + \gamma = 0. \quad (6.18)$$

Diferența față de (6.16) este prezența coeficientului  $\beta$ , care poate avea inclusiv valoarea zero. Deci modelul (6.18) este mai general decât (6.16). Deoarece egalitatea (6.18) se păstrează când înmulțim cu o constantă, trebuie să normalizăm coeficienții, adică să impunem condiția

$$\alpha^2 + \beta^2 + \gamma^2 = 1. \quad (6.19)$$

Presupunem acum că zgomotul afectează întreaga funcție  $g$ . Atenție, această ipoteză este diferită de cea de la problema CMMP anterioară: ambele variabile pot fi afectate de zgomot. Pentru a estima coeficienții necunoscuți, minimizăm eroarea

$$\sum_{i=1}^m |g(\xi_i, \eta_i)|^2 = \left\| \begin{bmatrix} \xi_1 & \eta_1 & 1 \\ \xi_2 & \eta_2 & 1 \\ \vdots & \vdots & \vdots \\ \xi_m & \eta_m & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} \right\|^2 = \|Ax\|^2 \quad (6.20)$$

cu restricția (6.19), adică  $\|x\| = 1$ . Când sunt suficient de multe măsurători, rangul matricei  $A$  din (6.20) este  $n = 3$ . Presupunem că DVS (6.1) a fost calculat. Necunoscuta  $x$  poate fi exprimată în baza formată de coloanele lui  $V$  prin

$$x = \sum_{i=1}^n z_i v_i = Vz. \quad (6.21)$$

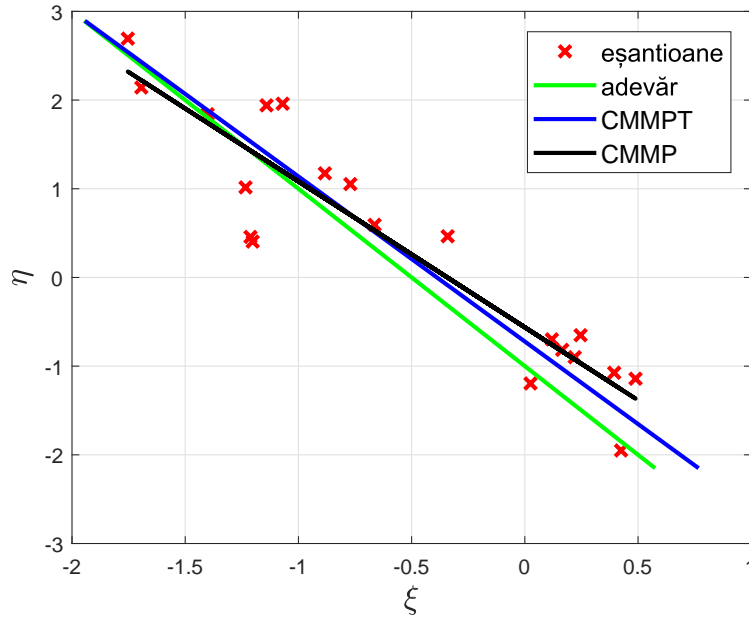


Figure 6.1: Modele liniare pentru un set de măsurători.

Observăm că  $\|x\| = \|z\|$  deoarece  $V$  este ortogonală. Ținând seama că  $z = V^T x$ , problema (6.20) devine

$$\|Ax\|^2 = \|U\Sigma VV^T z\|^2 = \|\Sigma z\|^2 = \sum_{i=1}^n \sigma_i^2 z_i^2. \quad (6.22)$$

Având în vedere că  $\|z\| = 1$  și valorile singulare sunt ordonate descrescător, valoarea minimă în (6.21) se obține când  $z_n = 1$  și  $z_1 = \dots = z_{n-1} = 0$ . Din (6.21) rezultă că  $x = v_n$ , adică soluția este vectorul singular dreapta corespunzător valorii singulare celei mai mici. (Reamintim că norma lui este egală cu 1, pentru că este o coloană a matricei ortogonale  $V$ .)

Metoda descrisă mai sus se numește și CMMP totale (CMMPT), deoarece zgomotul afectează întreaga matrice  $A$  din (6.20), în timp ce în (6.17) doar termenul liber  $b$  este afectat. Figura 6.1 ilustrează rezultatele obținute cu cele două metode. Dreapta verde este cea obținută pentru valorile adevărate ale parametrilor. Dreapta neagră, obținută cu metoda CMMP, minimizează suma pătratelor distanțelor până la eșantioane, distanțele fiind măsurate pe axa verticală. Dreapta albastră, obținută cu CMMPT, minimizează suma pătratelor distanțelor până la eșantioane, în sensul propriu de distanță de la o dreaptă la un punct. Insistăm asupra faptului că metoda se alege în funcție de felul modelului, nu de eventuala verificare vizuală a modului prin care dreapta trece printre puncte.



și  $V \in \mathbb{R}^{n \times n}$  și matricea bidiagonală  $J$  astfel încât  $J = U^T AV$ . Matricea  $J$  suprascrie matricea  $A$  dar se rețin numai elementele diagonale ale lui  $J$  în vectorul  $f \in \mathbb{R}^n$  și cele supradiagonale în vectorul  $g \in \mathbb{R}^{n-1}$ . Se folosesc vectorii de lucru  $u$  și  $v$ , precum și scalarii  $\beta$  și  $\gamma$ , pentru definirea reflectorilor curenți  $U_k = I_m - uu^T/\beta$  și  $V_{k+1} = I_n - vv^T/\gamma$ .

1.  $U = I_m$
2.  $V = I_n$
3. **pentru**  $k = 1 : n$ 
  - % Calculează  $U_k$
  - 1.  $\sigma = \text{sign}(a_{kk})\sqrt{\sum_{i=k}^m a_{ik}^2}$
  - 2.  $u_k = a_{kk} + \sigma$
  - 3. **pentru**  $i = k + 1 : m$ 
    - 1.  $u_i = a_{ik}$
  - 4.  $\beta = u_k \sigma$
  - 5.  $a_{kk} \leftarrow -\sigma$
  - %  $A \leftarrow U_k A$
  - 6. **pentru**  $j = k + 1 : n$ 
    - 1.  $\tau = (\sum_{i=k}^m u_i a_{ij})/\beta$
    - 2. **pentru**  $i = k : m$ 
      - 1.  $a_{ij} \leftarrow a_{ij} - \tau u_i$
  - %  $U \leftarrow U U_k$
  - 7. **pentru**  $i = 1 : m$ 
    - 1.  $\tau = (\sum_{j=k}^m U(i, j) u_j)/\beta$
    - 2. **pentru**  $j = k : m$ 
      - 1.  $U(i, j) \leftarrow U(i, j) - \tau u_j$
8. **dacă**  $k < n - 1$ 
  - % Calculează  $V_{k+1}$
  - 1.  $\sigma = \text{sign}(a_{k,k+1})\sqrt{\sum_{j=k+1}^n a_{kj}^2}$
  - 2.  $v_{k+1} = a_{k,k+1} + \sigma$
  - 3. **pentru**  $j = k + 2 : n$ 
    - 1.  $v_j = a_{kj}$
  - 4.  $\gamma = v_{k+1} \sigma$
  - %  $A \leftarrow A V_{k+1}$
  - 5.  $a_{k,k+1} \leftarrow -\sigma$
  - 6. **pentru**  $i = k + 1 : m$ 
    - 1.  $\tau = (\sum_{j=k+1}^n a_{ij} v_j)/\gamma$
    - 2. **pentru**  $j = k + 1 : n$ 
      - 1.  $a_{ij} \leftarrow a_{ij} - \tau v_j$
  - %  $V \leftarrow V V_{k+1}$
  - 7. **pentru**  $i = 1 : n$ 
    - 1.  $\tau = (\sum_{j=k+1}^n V(i, j) v_j)/\gamma$
    - 2. **pentru**  $j = k + 1 : n$ 
      - 1.  $V(i, j) \leftarrow V(i, j) - \tau v_j$

4. pentru  $i = 1 : n - 1$ 
  1.  $f_i = a_{ii}$
  2.  $g_i = a_{i,i+1}$
5.  $f_n = a_{nn}$

## 6.4 Sarcini de lucru

### 6.4.1 A. In laborator

1. Generați o matrice  $A \in \mathbb{R}^{m \times n}$  care să aibă rangul numeric  $r$  prin

$$A = BC + \sigma D$$

unde  $B \in \mathbb{R}^{m \times r}$ ,  $C \in \mathbb{R}^{r \times n}$  și  $D \in \mathbb{R}^{m \times n}$  sunt matrice aleatoare generate cu `randn` iar  $\sigma$  este un scalar mic. Puteți lua, de exemplu,  $m = 10$ ,  $n = 6$ ,  $r = 3$ . Folosind funcția `svd`, calculați valorile singulare ale lui  $A$  pentru  $\sigma = 0$  (este rangul egal cu  $r$ ?). Apoi creșteți  $\sigma$  și recalculați valorile singulare. Obțineți rezultatele așteptate? Cum trebuie ales  $\varepsilon$  în (6.10) astfel încât rangul numeric să fie într-adevăr  $r$ .

2. Scrieți un program MATLAB prin care încărcați o imagine (de exemplu `baboon.bmp`) într-o matrice  $A$  și salvați imaginile corespunzătoare aproximărilor (6.11) cu  $k = 10, 20, \dots$ . Folosiți funcțiile `imread` și `imwrite`. Vizualizați imaginile și constatați că aproximările sunt din ce în ce mai bune.
3. Folosind programul `gen_dreapta`, ai cărui parametri “adevărați” îi puteți modifica după dorință, generați  $m = 20$  de puncte conform modelului (6.18) (cu `zgomot`). Folosind metodele CMMP și CMMPT, calculați parametrii modelelor (6.16), respectiv (6.18). Pentru verificare, comparați direct cu parametrii “adevărați”. Pentru confirmare vizuală, desenați o figură similară cu Figura 6.1.
4. Scrieți un program MATLAB pentru implementarea algoritmului JQ de bidiagonalizare, cu sintaxa  $[f, g, U, V] = \mathbf{JQ}(A)$ . Testarea se va realiza prin calculul normei reziduuului  $A - UJV^T$ .

### 6.4.2 B. Acasă

- 1 Adaptați metoda de “compresie” cu DVS pentru cazul imaginilor color, în care fiecare pixel este codat prin trei numere, reprezentând intensitățile culorilor RGB. O imagine este deci memorată în trei matrice.
- 2 Cum procedați în cazul în care modelul (6.18) devine neliniar? Un exemplu simplu este  $g(\xi, \eta) = \alpha\xi + \beta\eta + \gamma\xi\eta + \delta$ . Observați că modelul este în continuare liniar în parametri.



- 3 Scrieți un program MATLAB de calcul al soluției CMMP (6.15) a unui sistem liniar în cazul cel mai general.