Factorizarea QR

Valentin-Ioan VINTILĂ

Facultatea de Automatică și Calculatoare - CTI
Universitatea POLITEHNICA București

7 martie 2023 (Lab. 2)

Cuprins

Chestiuni organizatorice (1)

Deoarece a avut loc și primul curs, știm că laboratorul valorează:

- $1p$, dacă dați parțial;
- (probabil) $2p$, dacă **nu** dați parțial.

Indiferent, notarea la laborator se va face astfel:

- **Lucrare** neanunțată din matrice - 50%;
- **Lucrare** neanunțată din funcții - 50%;
- **Bonus** la laborator - până la 50%.

Nota se trunchiază la 100%.

Chestiuni organizatorice (2)

Punctajul pe laborator se anulează COMPLET și IREVOCABIL:

- Dacă nu aveți minim 8 prezențe la laborator;
- **Dacă ați copiat la oricare dintre lucrări (teste/teme).**

În cazul din urmă, va fi înștiințat și cadrul didactic.

Matrice ortogonale

Definiție (matrice ortogonală)

Fie $Q \in \mathbb{R}^{n\times n}$, $n \in \mathbb{N}^*$ o matrice nesingulară. Aceasta se consideră **ortogonală** dacă și numai dacă $Q^{-1} = Q^T$.

Matrice ortogonale - exemplu

Spre exemplu, următoarea matrice este ortogonală:

$$Q = \begin{bmatrix} 0.6 & -0.8 \\ 0.8 & 0.6 \end{bmatrix}$$

Cum demonstrăm? Calculăm Q^T și verificăm că aceasta este de fapt inversa, adică $QQ^T = Q^TQ = I_2$:

$$Q^T = \begin{bmatrix} 0.6 & 0.8 \\ -0.8 & 0.6 \end{bmatrix} \Rightarrow \begin{cases} QQ^T = \begin{bmatrix} 0.6^2 + 0.8^2 & 0 \\ 0 & 0.8^2 + 0.6^2 \end{bmatrix} = I_2 \\ \text{analog, } Q^TQ = I_2 \end{cases}$$

Matrice ortogonale - proprietăți

Având în vedere definiția matricelor ortogonale ($Q^{-1} = Q^T$), reies următoarele proprietăți:

1. $QQ^T = Q^TQ = I_n$ (din proprietățile inversei);
2. $\det(Q) = \pm 1$ (demonstrația la tablă);

Factorizarea QR

Pentru o matrice $A \in \mathbb{R}^{n\times n}$, $n \in \mathbb{N}$, ne propunem să găsim două matrice $Q \in \mathbb{R}^{n\times n}$ și $R \in \mathbb{R}^{n\times n}$ astfel încât:

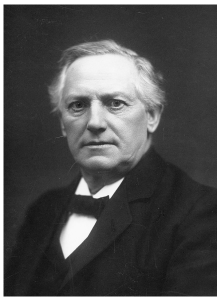
1. Q să fie **ortogonală**;
2. R să fie **superior triunghiulară**;
3. Produsul să revină în A, adică $\boxed{A = QR}$.

Factorizarea Gram-Schmidt - istorie

Din punct de vedere istoric:

- **Jørgen Pedersen Gram** publică în 1883 metoda;
- **Erhard Schmidt** publică o hârtie în 1907 ce face cunoscut algoritmul.

J.P. Gram și E. Schmidt



Jørgen Pedersen Gram (1850-1916)

Erhard Schmidt (1876-1959)

Produs scalar

Definiție (produs scalar)

Fie $u, v \in \mathbb{R}^n$ doi vectori, $n \in \mathbb{N}$. Definim **produsul scalar** al acestora prin:

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^T \mathbf{v}$$

Cu alte cuvinte, dacă $\mathbf{u} = \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix}$ și $\mathbf{v} = \begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}$, atunci:

$$\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{k=1}^{n} u_k v_k$$

Vectori ortogonali

Definiție (vectori ortogonal)

Fie $u, v \in \mathbb{R}^n$ doi vectori, $n \in \mathbb{N}$. Acești vectori se consideră **ortogonali** unul față de celălalt dacă și numai dacă produsul lor scalar este egal cu 0:

$$\langle \mathbf{u}, \mathbf{v} \rangle = \mathbf{u}^T \mathbf{v} = 0$$

În acest caz, se notează $u \perp v$.

Norma vectorilor

Definiție (norma Euclidiană)

Fie $\mathbf{u} \in \mathbb{R}^n$ un vector, $n \in \mathbb{N}$. Definim **norma** sa (Euclidiană) prin formula:

$$||\mathbf{u}|| = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}$$

Dacă $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ sunt ortogonali ($\mathbf{u} \perp \mathbf{v}$) și $||\mathbf{u}|| = ||\mathbf{v}|| = 1$, acești vectori se numesc **ortonormați**.

Algoritmul Gram-Schmidt - vectorial (1)

Să considerăm o mulțime de vectori din subspațiul $\mathbb{R}^n$, unde $n \in \mathbb{N}^*$, ce formează o bază $B = \{\mathbf{a_1}, \ldots, \mathbf{a_n}\}$.

Factorizarea Gram-Schmidt își propune să genereze o bază ortonormată $Q = \{\mathbf{q_1}, \ldots, \mathbf{q_n}\}$ în funcție de baza B.

Fiecare vector $\mathbf{a}_i$ din baza B, $\forall i = \overline{1, n}$, va trece printr-un proces de transformare, descris foarte simplist prin intermediul acestor doi pași:

$$\boxed{\mathbf{a_i} \to \mathbf{v_i} \to \mathbf{q_i}}, \ \forall i = \overline{1, n}$$

Algoritmul Gram-Schmidt - vectorial (2)

Vrem $\mathbf{a_i} \to \mathbf{v_i} \to \mathbf{q_i}$ - respectăm următorul set de reguli:

1. Fiecare vector $\mathbf{a_i}$, $\forall i = \overline{1, n}$, se va transforma pe rând, adică întâi a_1, apoi a_2, apoi a_3 etc.;
2. Vectorul $\mathbf{v_i}$ ($i > 1$) va proveni din $\mathbf{a_i}$ și va fi ortogonal cu fiecare vector $\mathbf{v_1}, \ldots, \mathbf{v_{i-1}}$ aflat înaintea sa; prin convenție, $\mathbf{v_1} = \mathbf{a_1}$;
3. Vectorul $\mathbf{q_i}$ va reprezenta normalizarea lui $\mathbf{v_i}$, așadar $\mathbf{q_i} = \frac{\mathbf{v_i}}{||\mathbf{v_i}||}$.

Corolar. $\mathbf{v_i} \perp \mathbf{v_j} \Leftrightarrow \mathbf{v_i} \perp \mathbf{q_j}$

Algoritmul Gram-Schmidt - exemplu (1)

Știm $B = \left\{ \begin{bmatrix} 3 \\ 4 \\ 0 \end{bmatrix}, \begin{bmatrix} 6 \\ 0 \\ 8 \end{bmatrix}, \begin{bmatrix} 0 \\ 7 \\ 0 \end{bmatrix} \right\}$. Vrem $Q = \{\mathbf{q_1}, \mathbf{q_2}, \mathbf{q_3}\}$.

Pasul 1. Calcularea lui $\mathbf{q_1}$:

- $\mathbf{q}_1 = \frac{\mathbf{v_1}}{||\mathbf{v_1}||} = \begin{bmatrix} 3/5 \\ 4/5 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.8 \\ 0 \end{bmatrix}$

Algoritmul Gram-Schmidt - exemplu (2)

Știm $B = \left\{ \begin{bmatrix} 3 \\ 4 \\ 0 \end{bmatrix}, \begin{bmatrix} 6 \\ 0 \\ 8 \end{bmatrix}, \begin{bmatrix} 0 \\ 7 \\ 0 \end{bmatrix} \right\}$. Vrem $Q = \{\mathbf{q_1} = \begin{bmatrix} 0.6 \\ 0.8 \\ 0 \end{bmatrix}, \mathbf{q_2}, \mathbf{q_3}\}$.

Pasul 2. Calcularea lui $\mathbf{q_2}$:

- $\mathbf{v_2} \perp \mathbf{v_1} \Rightarrow \mathbf{v_2} \perp \mathbf{q_1} \Leftrightarrow \langle \mathbf{v_2}, \mathbf{q_1} \rangle = 0$ (din corolar);
- $\mathbf{a_2}$ (poate) depinde de $\mathbf{q_1}$, deci:

$$\mathbf{a_2} = \mathbf{v_2} + \alpha_{21}\mathbf{q_1} \Rightarrow \mathbf{v_2} = \mathbf{a_2} - \alpha_{21}\mathbf{q_1}$$

- Din aceste ecuații:

$$\begin{aligned} \langle \mathbf{v_2}, \mathbf{q_1} \rangle = 0 &\Rightarrow \langle \mathbf{a_2} - \alpha_{21}\mathbf{q_1}, \mathbf{q_1} \rangle = 0 \\ &\Rightarrow \langle \mathbf{a_2}, \mathbf{q_1} \rangle - \alpha_{21} \cdot \langle \mathbf{q_1}, \mathbf{q_1} \rangle = 0 \end{aligned}$$

- Știm însă $||\mathbf{q_1}|| = 1 \Rightarrow \alpha_{21} = \langle \mathbf{a_2}, \mathbf{q_1} \rangle = 3.6$

Algoritmul Gram-Schmidt - exemplu (3)

Știm $B = \left\{ \begin{bmatrix} 3 \\ 4 \\ 0 \end{bmatrix}, \begin{bmatrix} 6 \\ 0 \\ 8 \end{bmatrix}, \begin{bmatrix} 0 \\ 7 \\ 0 \end{bmatrix} \right\}$. Vrem $Q = \{\mathbf{q_1} = \begin{bmatrix} 0.6 \\ 0.8 \\ 0 \end{bmatrix}, \mathbf{q_2}, \mathbf{q_3}\}$.

Pasul 2. Calcularea lui $\mathbf{q_2}$:

- $\mathbf{v_2} = \mathbf{a_2} - \alpha_{21}\mathbf{q_1}$
- $\alpha_{21} = \langle \mathbf{a_2}, \mathbf{q_1} \rangle = 3.6$
- $\boxed{\mathbf{v_2} = \mathbf{a_2} - \langle \mathbf{a_2}, \mathbf{q_1} \rangle \cdot \mathbf{q_1}}$ sau, numeric, $\mathbf{v_2} = \begin{bmatrix} 3.84 \\ -2.88 \\ 8 \end{bmatrix}$
- $\mathbf{q_2} = \frac{\mathbf{v_2}}{||\mathbf{v_2}||} \approx \begin{bmatrix} 0.4116 \\ -0.3078 \\ 0.8575 \end{bmatrix}$

Algoritmul Gram-Schmidt - exemplu (4)

Știm $B = \left\{ \begin{bmatrix} 3 \\ 4 \\ 0 \end{bmatrix}, \begin{bmatrix} 6 \\ 0 \\ 8 \end{bmatrix}, \begin{bmatrix} 0 \\ 7 \\ 0 \end{bmatrix} \right\}$. Vrem $Q = \{\mathbf{q_1} = \begin{bmatrix} 0.6 \\ 0.8 \\ 0 \end{bmatrix}, \mathbf{q_2} \approx \begin{bmatrix} 0.4116 \\ -0.3078 \\ 0.8575 \end{bmatrix}, \mathbf{q_3}\}$. Analog, pentru $\mathbf{q_3}$:

- $\mathbf{v_3} \perp \mathbf{v_1}$ și $\mathbf{v_3} \perp \mathbf{v_2}$ sau, prin corolar, $\mathbf{v_3} \perp \mathbf{q_1}$ și $\mathbf{v_3} \perp \mathbf{q_2}$, deci:

$$\mathbf{a_3} = \mathbf{v_3} + \alpha_{31}\mathbf{q_1} + \alpha_{32}\mathbf{q_2} \Rightarrow \mathbf{v_3} = \mathbf{a_3} - \alpha_{31}\mathbf{q_1} - \alpha_{32}\mathbf{q_2}$$

se va reduce la:

$$\boxed{\mathbf{v_3} = \mathbf{a_3} - \langle \mathbf{a_3}, \mathbf{q_1} \rangle \cdot \mathbf{q_1} - \langle \mathbf{a_3}, \mathbf{q_2} \rangle \cdot \mathbf{q_2}}$$

Algoritmul Gram-Schmidt - exemplu (5)

Prin calcul, ajungem de la baza $B = \left\{ \begin{bmatrix} 3 \\ 4 \\ 0 \end{bmatrix}, \begin{bmatrix} 6 \\ 0 \\ 8 \end{bmatrix}, \begin{bmatrix} 0 \\ 7 \\ 0 \end{bmatrix} \right\}$ la baza

$Q = \left\{ \begin{bmatrix} 0.6 \\ 0.8 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.4116 \\ -0.3078 \\ 0.8757 \end{bmatrix}, \begin{bmatrix} -0.6865 \\ 0.5154 \\ 0.5129 \end{bmatrix} \right\}$.

Algoritmul Gram-Schmidt - generalizare

În baza exemplului anterior, concluzionăm că:

$$\boxed{\mathbf{v_i} = \mathbf{a_i} - \sum_{k=1}^{i-1} \langle \mathbf{q_k}, \mathbf{q_j} \rangle \cdot \mathbf{q_k}}$$

Evident, $\mathbf{q_i}$ rămâne $\mathbf{q_i} = \frac{\mathbf{v_i}}{||\mathbf{v_i}||}$

Factorizarea Gram-Schmidt

Vrem $A = QR$, unde Q este **ortogonală** și R **superior triunghiulară**.

Aplicăm algoritmul GS considerând baza inițială $B = \{\mathbf{a_1}, \dots, \mathbf{a_n}\}$, adică vectorii coloană componenți ai matricei A.

Obținem baza ortonormată Q, adică **matricea ortogonală** Q.

Matricea $R = (r_{ij})$ devine $\begin{cases} 0, & i > j \\ ||v_i||, & i = j \text{ (ca temă, puteți verifica).} \\ \langle a_j, q_i \rangle, & i < j \end{cases}$

Factorizarea Gram-Schmidt - concluzii

Complexitate? $O(n^3)$

O folosim în practică? **NU! E instabilă numeric!**

Factorizarea Gram-Schmidt modificată

Pentru a fi folosită factorizarea GS, se modifică ordinea operațiilor.

Știm $r_{ij} = \langle \mathbf{a_j}, \mathbf{q_k} \rangle$.

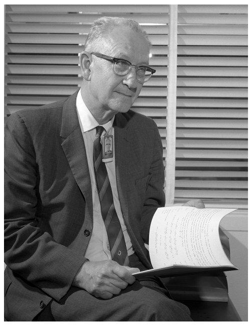
Mai știm $\mathbf{q_k} \perp \mathbf{q_j} \Rightarrow \langle \mathbf{q_k}, \mathbf{q_j} \rangle = 0 \Rightarrow \langle r_{kj}\mathbf{q_k}, \mathbf{q_j} \rangle = 0$.

Așadar, r_{ij} se rescrie ca $\boxed{r_{ij} = \left\langle \mathbf{a_j} - \sum_{k=1}^{i-1} r_{kj}\mathbf{q_k}, \mathbf{q_i} \right\rangle}$.

Factorizarea Householder

Schimbăm abordarea și căutăm o alternativă pentru GSM, întrucât acesta este în continuare instabil numeric.

Alston Scott Householder



Alston Scott Householder (1904-1993)

Reflector Householder

Definiție (reflector Householder)

Fie un vector $\mathbf{u} \in \mathbb{R}^n$, $n \in \mathbb{N}^*$, cu norma sa euclidiană $||\mathbf{u}||$. Atunci, se definește matricea $H \in \mathbb{R}^{n \times n}$ astfel încât:

$$\boxed{H = I_n - 2 \cdot \frac{\mathbf{u}\mathbf{u}^T}{||\mathbf{u}||^2} = I_n - 2 \cdot \frac{\mathbf{u}\mathbf{u}^T}{\mathbf{u}^T\mathbf{u}}}$$

Matricea H se numește **reflector Householder**.

Reflector Householder - proprietăți

Câteva proprietăți ai reflectorului Householder cuprind:

1. **Simetria**: $H = H^T$
2. **Ortogonalitatea și involuția**: $H^T H = H H^T = I_n$

Demonstrațiile rămân ca temă!

Reflector Householder - geometric

Produsul $H\mathbf{x}$ (cu H **reflector** și $\mathbf{x}$ **vector**) va **reflecta** vectorul $\mathbf{x}$ relativ față de planul perpendicular pe $\mathbf{u}$.

Să demonstrăm cazul 2D...

Demonstrație Householder geometric (1)

Fie $\mathbf{u}, \mathbf{v} \in \mathbb{R}^2$ aleși astfel încât $\mathbf{u} \perp \mathbf{v}$ și $||\mathbf{u}|| = 1$.

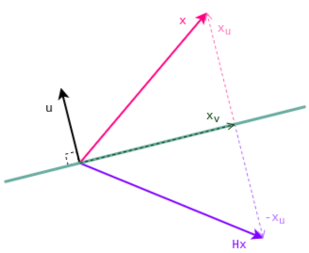
Orice vector $\mathbf{x} \in \mathbb{R}^2$ poate fi descompus ca $\mathbf{x} = \mathbf{x_v} + \mathbf{x_u}$, unde $\mathbf{x_v} \parallel \mathbf{v}$ și $\mathbf{x_u} \parallel \mathbf{u}$. Deoarece $\mathbf{x_u}$ este proiecția lui $\mathbf{x}$ pe $\mathbf{u}$ avem:

$$\mathbf{x_u} = \frac{\mathbf{u}\mathbf{u}^T}{||\mathbf{u}||^2} \cdot \mathbf{x} = \mathbf{u}\mathbf{u}^T\mathbf{x}$$

Aplicăm acum transformarea $H\mathbf{x}$:

$$H\mathbf{x} = (I - 2\mathbf{u}\mathbf{u}^T)(\mathbf{x_v} + \mathbf{u}\mathbf{u}^T x) = \mathbf{x_v} - \mathbf{u}\mathbf{u}^T\mathbf{x} = \mathbf{x_v} - \mathbf{x_u}$$

Demonstrație Householder geometric (2)



Interpretarea geometrică a reflectorului Householder. Cu turcoaz este desenat "planul" (dreapta) perpendicular(ă) vectorului **u**.

Aflarea relfectorului

Fie $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ ($n \in \mathbb{N}^*$) astfel încât $H\mathbf{x} = \mathbf{y}$, unde H este un reflector Householder. Atunci:

$$\boxed{H = I_n - 2\mathbf{u}\mathbf{u}^T, \text{ unde } \mathbf{u} = \frac{\mathbf{x} - \mathbf{y}}{||\mathbf{x} - \mathbf{y}||}}$$

Demonstrația se face prin simpla înlocuire.

Fundamentul factorizării (1)

Fie $A = \begin{bmatrix} \mathbf{a_1} & \mathbf{a_2} & \dots & \mathbf{a_n} \end{bmatrix} \in \mathbb{R}^{n \times n}$ și vectorul $\mathbf{e_1}$ din baza subspațiului aritmetic $\mathbb{R}^n$ ($n \in \mathbb{N}^*$). Vrem $\mathbf{a_1} \to ||\mathbf{a_1}||\mathbf{e_1}$. Cum?

$$\boxed{H_1 = I_n - 2\mathbf{u_1}\mathbf{u_1^T}}, \text{ unde } \mathbf{u_1} = \frac{\mathbf{a_1} - \rho_1 \mathbf{e_1}}{||\mathbf{a_1} - \rho_1 \mathbf{e_1}||} \text{ și } \rho_1 = ||\mathbf{a_1}||$$

Aplicând operația H_1A, obținem:

$$\boxed{H_1 A = \begin{bmatrix} H_1\mathbf{a_1} & H_1\mathbf{a_2} & \dots & H_1\mathbf{a_n} \end{bmatrix} = \begin{bmatrix} \rho_1 & R_{1,2:n} \in \mathbb{R}^{1\times(n-1)} \\ 0 & B \in \mathbb{R}^{(n-1)\times(n-1)} \end{bmatrix}}$$

Fundamentul factorizării (2)

Continuând recursiv pe matricea B (cu vectorul $\mathbf{e_1}$ din baza subspațiului aritmetic $\mathbb{R}^{n-1}$), obținem:

$$\boxed{\hat{H}_2 = I_{n-1} - 2\mathbf{u_2}\mathbf{u_2^T}}, \ \mathbf{u_2} = \frac{\mathbf{b_1} - \rho_2 \mathbf{e_1}}{||\mathbf{b_1} - \rho_2 \mathbf{e_1}||} \text{ și } \rho_2 = ||\mathbf{b_1}||$$

și, dacă $H_2 = \begin{bmatrix} 1 & 0 \\ 0 & \hat{H}_2 \end{bmatrix}$, obținem:

$$\boxed{H_2 H_1 A = \begin{bmatrix} H_2H_1\mathbf{a_1} & H_2H_1\mathbf{a_2} & \dots & H_2H_1\mathbf{a_n} \end{bmatrix} = \begin{bmatrix} \rho_1 & R_{12} & R_{1,3:n} \\ 0 & \rho_2 & R_{2,3:n} \\ 0 & 0 & C \end{bmatrix}}$$

Fundamentul factorizării (3)

De exemplu, pentru o matrice 4×4, procesul va arăta cam așa:

$$A = \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \to H_1 A = \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \end{bmatrix}$$

$$\to H_2 H_1 A = \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \end{bmatrix} \to \dots$$

Factorizarea Householder

Vrem $A = QR$, unde Q este **ortogonală** și R **superior triunghiulară**.

Devine evident că $R = H_{n-1}H_{n-2}\dots H_1 A$.

Știm că:

1. Produsul de matrice ortogonale este o matrice ortogonală *(demonstrația la tablă)*;
2. $H^2 = HH^T = H^TH = I_n$ (unde H este reflector Householder).

Obținem $Q = H_1 H_2 \dots H_{n-1}$ (demonstrația se face prin calcul).

Factorizarea Householder - exemplu (1)

Fie $A = \begin{bmatrix} 2 & 4 & 5 \\ 1 & -1 & 1 \\ 2 & 1 & -1 \end{bmatrix}$. Vrem să găsim $Q, R \in \mathbb{R}^3$ astfel încât $A = QR$.

Pasul 1. Calculăm H_1 și B.

Știm $\mathbf{a_1} = \begin{bmatrix} 2 & 1 & 2 \end{bmatrix}^T$, deci $\rho_1 = ||\mathbf{a_1}|| = 3$, așadar:

$$\mathbf{u_1} = \frac{\mathbf{a_1} - \rho_1 \mathbf{e_1}}{||\mathbf{a_1} - \rho_1 \mathbf{e_1}||} = \frac{1}{\sqrt{6}} \begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix}$$

Factorizarea Householder - exemplu (2)

Fie $A = \begin{bmatrix} 2 & 4 & 5 \\ 1 & -1 & 1 \\ 2 & 1 & -1 \end{bmatrix}$. Vrem să găsim $Q, R \in \mathbb{R}^3$ astfel încât $A = QR$.

Pasul 1. Calculăm H_1 și B.

Trecând la $H_1 = I_3 - 2\mathbf{u_1}\mathbf{u_1}^T$ obținem:

$$H_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{1}{3}\begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix} \begin{bmatrix} -1 & 1 & 2 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 & -1 & -2 \\ -1 & 1 & 2 \\ -2 & 2 & 4 \end{bmatrix}$$

Calculăm H_1A pentru a afla B:

$$H_1 A = \begin{bmatrix} 3 & 3 & 3 \\ 0 & 0 & 3 \\ 0 & 3 & 3 \end{bmatrix} \Rightarrow B = \begin{bmatrix} 0 & 3 \\ 3 & 3 \end{bmatrix}$$

Factorizarea Householder - exemplu (3)

Fie $A = \begin{bmatrix} 2 & 4 & 5 \\ 1 & -1 & 1 \\ 2 & 1 & -1 \end{bmatrix}$. Vrem să găsim $Q, R \in \mathbb{R}^3$ astfel încât $A = QR$.

Pasul 2. Calculăm H_2 și R.

Știm $\mathbf{b_1} = \begin{bmatrix} 0 & 3 \end{bmatrix}^T$, deci $\rho_2 = ||\mathbf{b_1}|| = 3$, așadar:

$$\mathbf{u_2} = \frac{\mathbf{b_2} - \rho_2 \mathbf{e_1}}{||\mathbf{b_2} - \rho_2 \mathbf{e_1}||} = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 \\ 1 \end{bmatrix} \text{ (amintim că } \mathbf{e_1} \in \mathbb{R}^2\text{)}$$

Factorizarea Householder - exemplu (4)

Fie $A = \begin{bmatrix} 2 & 4 & 5 \\ 1 & -1 & 1 \\ 2 & 1 & -1 \end{bmatrix}$. Vrem să găsim $Q, R \in \mathbb{R}^3$ astfel încât $A = QR$.

Pasul 2. Calculăm H_2 și R.

Calculăm rapid $\hat{H}_2$ și H_2:

$$\hat{H}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} -1 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \Rightarrow H_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Calculăm și matricea R:

$$R = H_2 H_1 A = 3 \begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Factorizarea Householder - exemplu (5)

Am obținut deci că matricea $A = \begin{bmatrix} 2 & 4 & 5 \\ 1 & -1 & 1 \\ 2 & 1 & -1 \end{bmatrix}$ se descompune QR astfel: $Q = H_1 H_2 = \frac{1}{3}\begin{bmatrix} 2 & 2 & 1 \\ 1 & -2 & 2 \\ 2 & -1 & -2 \end{bmatrix}$ și $R = H_2 H_1 A = 3\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$.

Factorizarea Householder - concluzii

Complexitate? $O(n^3)$

O folosim în practică? DA! Este foarte stabilă și rapidă!

...și atunci de ce nu se încheie prezentarea?

Wallace Givens

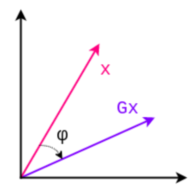
James Wallace Givens, Jr (1910-1993)

Matrice Givens 2D (1)

Ideea principală este de a roti un vector în plan folosind o matrice.

Pentru a roti un vector $\mathbf{x} \in \mathbb{R}^2$ cu $\varphi \in \mathbb{R}$ radiani, folosim:
$G_\varphi = \begin{bmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{bmatrix}$ *(demonstrația rămâne ca temă).*

Matrice Givens 2D (2)



Rotirea lui $\mathbf{x}$ în sensul acelor de ceasornic cu φ radiani, folosind matricea G, adică $G\mathbf{x}$

Eliminarea unei coordonate în 2D

Fie $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ și $\mathbf{y} = \begin{bmatrix} ||\mathbf{x}|| \\ 0 \end{bmatrix}$ doi vectori aleși astfel încât $\mathbf{y} = G\mathbf{x}$.

$$\boxed{\cos\varphi = \frac{x_1}{||\mathbf{x}||} = \frac{x_1}{\sqrt{x_1^2 + x_2^2}}} \text{ și } \boxed{\sin\varphi = \frac{x_2}{||\mathbf{x}||} = \frac{x_2}{\sqrt{x_1^2 + x_2^2}}}$$

Aceste formule se dovedesc rapid:

$$\mathbf{y} = G\mathbf{x} = \begin{bmatrix} \cos\varphi & \sin\varphi \\ -\sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} ||\mathbf{x}|| \\ 0 \end{bmatrix}$$

Matrice Givens (1)

Matricea de rotație 2D poate fi generalizată pentru mai multe dimensiuni, rotind un vector în planul definit de coordonatele i și j:

$$\boxed{G_{ij} = \begin{bmatrix} 1 & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & \cos\varphi & \dots & \sin\varphi & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & -\sin\varphi & \dots & \cos\varphi & \dots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & 1 \end{bmatrix}}$$

Matrice Givens (2)

$$g_{ab} = \begin{cases} \cos\varphi, & a = b = i \text{ sau } a = b = j \\ \sin\varphi, & a = i,\ b = j \\ -\sin\varphi, & a = j,\ b = i \\ 1, & a = b,\ a \neq i,\ a \neq j \\ 0, & \text{în rest} \end{cases}$$

Eliminarea unei coordonate (1)

Simpla operație conduce către: $G_{ij}\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_i \cos\varphi + x_j \sin\varphi \\ \vdots \\ \color{red}{-x_i \sin\varphi + x_j \cos\varphi} \\ \vdots \\ x_n \end{bmatrix}$

Vrem să eliminăm partea cu roșu.

Eliminarea unei coordonate (2)

Așadar, pentru a elimina coordonata j și a o păstra pe i:

$$\boxed{G_{ij}\mathbf{x} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}, \text{ unde } y_k = \begin{cases} \sqrt{x_i^2 + x_j^2}, & k = i \\ 0, & k = j \\ x_k, & k \neq i,\ k \neq j \end{cases}}$$

Eliminarea unei coordonate (3)

La modul general, operația $\mathbf{x} \to ||\mathbf{x}||\mathbf{e}_1$ devine:

$$G_{12}\mathbf{x} = \begin{bmatrix} \sqrt{x_1^2 + x_2^2} \\ 0 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} \to (G_{13} \cdot G_{12})\,\mathbf{x} = \begin{bmatrix} \sqrt{x_1^2 + x_2^2 + x_3^2} \\ 0 \\ 0 \\ \vdots \\ x_n \end{bmatrix} \to \dots$$

Eliminarea unei coordonate (4)

La modul general, operația $\mathbf{x} \to ||\mathbf{x}||\mathbf{e}_1$ devine:

$$(G_{1n} \dots G_{13} \cdot G_{12})\,\mathbf{x} = \begin{bmatrix} \sqrt{x_1^2 + x_2^2 + \dots + x_n^2} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} ||\mathbf{x}|| \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Factorizarea Givens (1)

Vrem $A = QR$, unde Q este **ortogonală** și R **superior triunghiulară**.

Aplicăm algoritmul Givens considerând $A = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \dots & \mathbf{a}_n \end{bmatrix} \in \mathbb{R}^{n \times n}$.

Definim matricele $G_1, \dots, G_{n-1}$:

$$G_k = G_{kn} G_{k(n-1)} \dots G_{k(k+1)}$$

Prin $G_1 A$ ajungem la forma:

$G_1 A = \begin{bmatrix} G_1\mathbf{a}_1 & G_1\mathbf{a}_2 & \dots & G_1\mathbf{a}_n \end{bmatrix} = \begin{bmatrix} ||\mathbf{a}_1|| & R_{1,2:n} \\ 0 & B \end{bmatrix}$.

Factorizarea Givens (2)

Vrem $A = QR$, unde Q este **ortogonală** și R **superior triunghiulară**.

Continuând recursiv, $G_2 \begin{bmatrix} R_{1,2:n} \\ B_2 \end{bmatrix} = \begin{bmatrix} R_{12} & R_{1,3:n} \\ ||\mathbf{b}_2|| & R_{2,3:n} \\ 0 & C \end{bmatrix}$.

Așadar, $\boxed{R = G_{n-1} \dots G_1 A}$.

Se poate demonstra că:

$$\boxed{Q = G_1^T \cdot G_2^T \dots G_n^T = \prod_{i=1}^{n} G_i^T}$$

Puteți demonstra rezultatul ca temă!

Fundamentul factorizării Givens (1)

De exemplu, pentru o matrice 4×4, procesul va arăta cam așa:

$$A = \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \to G_1 A = \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \end{bmatrix}$$

$$\to G_2 G_1 A = \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \\ 0 & 0 & \boxtimes & \boxtimes \end{bmatrix} \to \dots$$

...deci la fel ca la Householder!

Fundamentul factorizării Givens (2)

Putem însă să vedem lucrurile mai în profunzime:

$$A = \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \to G_{12} A = \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix}$$

$$\to G_{13} G_{12} A = \begin{bmatrix} \boxtimes & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ 0 & \boxtimes & \boxtimes & \boxtimes \\ \boxtimes & \boxtimes & \boxtimes & \boxtimes \end{bmatrix} \to \dots$$

Avem deci acces la granularitate!

Factorizarea Givens - exemplu

Ca temă, puteți să factorizați Givens următoarea matrice:

$$A = \begin{bmatrix} 0 & -1 & 1 \\ 4 & 2 & 0 \\ 3 & 4 & 0 \end{bmatrix}$$

Veți obține: $A \approx \begin{bmatrix} 0 & -0.447 & 0.894 \\ 0.8 & -0.537 & -0.268 \\ 0.6 & 0.716 & 0.358 \end{bmatrix} \cdot \begin{bmatrix} 5 & 4 & 0 \\ 0 & 2.236 & -0.447 \\ 0 & 0 & 0.897 \end{bmatrix}.$

Factorizarea Givens - concluzii

Complexitate? $O(n^3)$

O folosim în practică? DA!

Când o preferăm în detrimentul Householder? (dau bonus!)

MATRICE SPARSE

Pentru cine și-a notat...

Am lăsat în această prezentare 5 exerciții și/sau demonstrații ca temă.

Cei care le lucrează **individual** până data viitoare și le aduc **scrise pe o hârtie semnată**, pot primi până la 15% din bonus!

Bibliografie

Pentru aceste prezentări, am utilizat:

1. Cărțile *Matrix Decomposition and Applications*, respectiv *Numerical Matrix Decomposition and its Modern Applications: A Rigorous First Course* ale lui **Jun Lu**.

Sfârșit

Mulțumesc frumos pentru atenție!

Vă rog frumos să completați formularul de feedback!