Algoritmy pro rozsáhlá data

L09: rozklad na singulární hodnoty (SVD)

Jan Konecny

10. prosince 2021

O čem to je?

Rozklad singulární hodnoty (SVD) je faktorizací reálné (nebo komplexní) matice.

Věta 1

Jakákoli $m \times n$ matice A s $m \geq n$ může být rozložena na

$$A = U \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} V^{\mathrm{T}}$$

kde

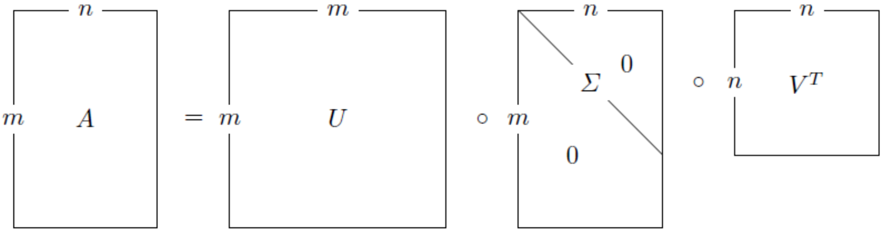
- $U \in \mathbb{R}^{m \times m}$ *a* $V \in \mathbb{R}^{n \times n}$ *jsou ortogonální*
- $\Sigma \in \mathbb{R}^{n \times n}$ *je diagonální* $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_n)$ *a*

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n \geq 0.$$

- Předpoklad $m \geq n$ není nijak omezující: v opačném případě můžeme větu aplikovat na A^{T}.

- Sloupce U a V se nazývají singulární vektory,
- Diagonální prvky σ_i matice Σ se nazývají singulární hodnoty.

SVD symbolicky:



Vlastnosti SVD

Pro důkaz budeme potřebovat:

Věta 2

Pro matici $Q_1 \in \mathbb{R}^{m\times k}$ s ortonormálními sloupci existuje matice $Q_2 \in \mathbb{R}^{m\times(m-k)}$ tak že $Q = (Q_1\ Q_2)$ je ortogonální matice.

Slovy:

- každá ortonormální báze podprostoru prostoru R^m může být rozšířena na ortonormální bázi celého prostoru.

2-norma matice:

$$\begin{aligned}\|A\|_2 &= \max_{\vec{x}} \frac{\|A\vec{x}\|_2}{\|\vec{x}\|_2} \\ &= \sqrt{\max_{1\le i\le n} \lambda_i(A^{\mathrm{T}}A)} =: \sigma_{\max}(A)\end{aligned}$$

Důkaz.

- Uvažujme vektor $\vec{x}$, pro který je $\|A\vec{x}\|_2$ maximální – vždy existuje (hledáme supremum spojité funkce nad uzavřenou množinou).
- Položme $A\vec{x} = \sigma_1\vec{y}$, kde $\|\vec{y}\|_2 = 1$ a $\sigma_1 = \|A\|_2$.
- Podle věty 1 můžeme zkonstruovat ortogonální matice

$$Z_1 = (\vec{y} \;\; \overline{Z}_2) \in \mathbb{R}^{m\times m}$$
$$W_1 = (\vec{x} \;\; \overline{W}_2) \in \mathbb{R}^{m\times m}$$

- Pak

$$Z_1^{\mathrm{T}} A W_1 = \begin{pmatrix} \sigma_1 & \vec{y}^{\mathrm{T}} A \overline{W}_2 \\ \vec{0} & \overline{Z}_2^{\mathrm{T}} A \overline{W}_2 \end{pmatrix},$$

protože $\vec{y}^{\mathrm{T}} A\vec{x} = \sigma_1$ a $\overline{Z}_2^{\mathrm{T}} A\vec{x} = \sigma_1 \overline{Z}_2^{\mathrm{T}} \vec{y} = \vec{0}$.

Důkaz.

- Položme

$$A_1 = Z_1^{\mathrm{T}} A W_1 = \begin{pmatrix} \sigma_1 & \vec{w}^T \\ \vec{0} & B \end{pmatrix}$$

- Pak

$$\left(\frac{\left\| A_1 \begin{pmatrix} \sigma_1 \\ \vec{w} \end{pmatrix} \right\|_2}{\left\| \begin{pmatrix} \sigma_1 \\ \vec{w} \end{pmatrix} \right\|_2} \right)^2 = \frac{1}{\sigma_1^2 + \vec{w}^{\mathrm{T}} \vec{w}} \left\| A_1 \begin{pmatrix} \sigma_1 \\ \vec{w} \end{pmatrix} \right\|_2^2$$

$$= \frac{1}{\sigma_1^2 + \vec{w}^{\mathrm{T}} \vec{w}} \left\| \begin{pmatrix} \sigma_1^2 + \vec{w}^{\mathrm{T}} \vec{w} \\ B\vec{w} \end{pmatrix} \right\|_2^2$$

$$\geq \sigma_1^2 + \vec{w}^{\mathrm{T}} \vec{w}$$

- Ale $\|A_1\|_2^2 = \|Z_1^T A W_1\|_2^2 = \sigma_1^2$. Tedy $\vec{w}$ musí být nulový.

Důkaz.

To znamená

$$A_1 = Z_1^{\mathrm{T}} A W_1 = \begin{pmatrix} \sigma & \vec{0}^{\mathrm{T}} \\ \vec{0} & B \end{pmatrix}.$$

Důkaz je pak dokončen indukcí:

- Předpokládejme, že

$$B = Z_2 \begin{pmatrix} \Sigma \\ 0 \end{pmatrix} W_2 \qquad \text{kde } \Sigma_2 = \operatorname{diag}(\sigma_2, \ldots, \sigma_n).$$

- Pak máme

$$\begin{aligned} A &= Z_1 \begin{pmatrix} \sigma_1 & \vec{0}^{\mathrm{T}} \\ \vec{0} & B \end{pmatrix} W_1^{\mathrm{T}} \\ &= Z_1 \begin{pmatrix} 1 & 0 \\ 0 & Z_2 \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & W_2^{\mathrm{T}} \end{pmatrix} W_1^{\mathrm{T}}. \end{aligned}$$

Důkaz.

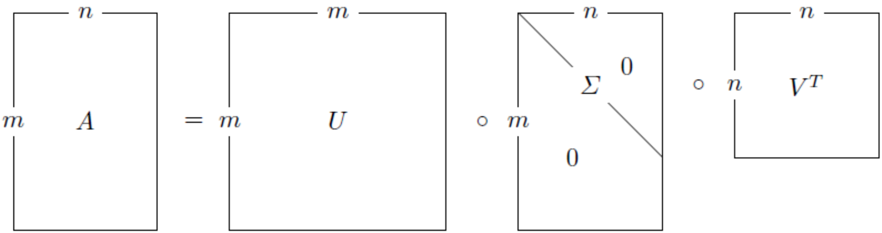
$$A = Z_1 \begin{pmatrix} 1 & 0 \\ 0 & Z_2 \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & W_2^{\mathrm{T}} \end{pmatrix} W_1^{\mathrm{T}}.$$

Definujeme U, Σ, V jako

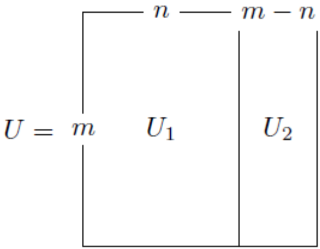
$$U = Z_1 \begin{pmatrix} 1 & 0 \\ 0 & Z_2 \end{pmatrix} \qquad \Sigma = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix} \qquad V = W_1 \begin{pmatrix} 1 & 0 \\ 0 & W_2^{\mathrm{T}} \end{pmatrix}.$$

□

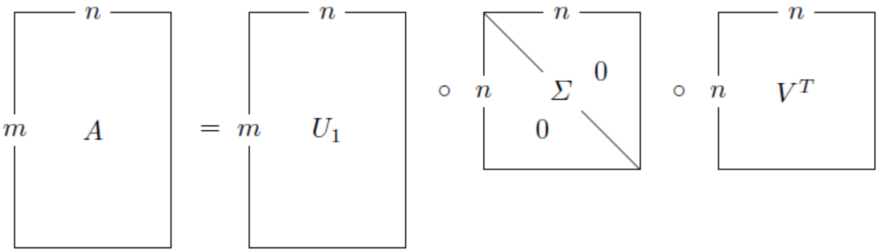
SVD symbolicky:



Matici U můžeme vertikálně rozdělit na $m \times n$ matici U_1 a $m \times (m-n)$ matici U_2:



Při násobení $U\Sigma$ se hodnoty z U_2 násobí pouze nulami.
Můžeme tedy psát:

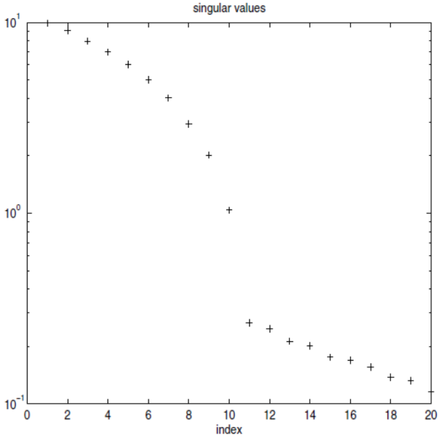


Aplikace SVD

Uvažujme, že matici A s nízkou hodností a přidaným šumem

$$A = A_0 + N,$$

kde šum (N) je malý v porovnání A_0.
Singulární hodnoty se pak typicky chovají takto:



Pokud je šum dostatečně malý, můžeme

- uvažovat pouze velké vlastní hodnoty a rekonstruovat matici A pouze pomocí nich (a odpovídajících singulárních vektorů);
- aproximovat tak matici A maticí s nižší hodnosti.

Předpokládejme, že počet velkých vlastních hodnot je k.

Pak aproximujeme

$$A \approx \sum_{i=1}^{n} \sigma_i u_i v_i^{\mathrm{T}} := A_k.$$

Tomuto se říká ořezaná SVD.

Ořezaná SVD je řešením aproximačního problému, kde se chce aproximovat danou matici maticí nižší hodnosti.

Věta 3

Předpokládejme, že matice $A \in \mathrm{R}^{m\times n}$ má hodnost $r > k$.
Problém aproximace matice A

$$\min_{h(Z)=k} \|A - Z\|_2$$

má řešení

$$Z = A_k = U_k \Sigma_k V_k^{\mathrm{T}}$$

kde $U_k = (u_1, \ldots, u_k)$ $V_k = (v_1, \ldots, v_k)$ a $\Sigma_k = \mathrm{diag}(\sigma_1, \ldots, \sigma_k)$.
To minimum je

$$\|A - A_k\|_2 = \sigma_{k+1}.$$

Toto s dá ukázat i pro *Frobeniovu normu*

$$\|A\|_F = \sqrt{\sum_{i,j} a_{ij}^2}$$

Věta 4

Předpokládejme, že matice $A \in \mathrm{R}^{m \times n}$ má hodnost $r > k$.
Problém aproximace matice A

$$\min_{h(Z)=k} \|A - Z\|_F$$

má řešení

$$Z = A_k = U_k \Sigma_k V_k^{\mathrm{T}}$$

kde $U_k = (u_1, \ldots, u_k)$ $V_k = (v_1, \ldots, v_k)$ a $\Sigma_k = \mathrm{diag}(\sigma_1, \ldots, \sigma_k)$.
To minimum je

$$\|A - A_k\|_F = \left(\sum_{i=k+1}^{n} \sigma_i^2 \right)^{\frac{1}{2}}$$

Analýza hlavních komponent (principal component analysis (PCA))

Aproximační vlastnosti SVD mohou být použity k vysvětlení ekvivalence SVD a PCA.

- Uvažujme, že $X \in \mathbb{R}^{m \times n}$ je datová matice, kde každý sloupec je pozorování náhodného vektoru z $\mathbb{R}^n$ s nulovým průměrem.
- Nechť SVD matice X je $X = U\Sigma V^{\mathrm{T}}$.
- Pravé singulární vektory $\vec{v}_i$ se nazývají směry hlavních komponent v X.
- Vektor

  $$\vec{z}_1 = X\vec{v}_1 = \sigma_1 \vec{u}_1$$

  má největší varianci ve vzorku mezi všemi normalizovanými lineárními kombinacemi sloupců z X:

  $$\mathrm{Var}(\vec{z}_1) = \mathrm{Var}(X\vec{v}_1) = \frac{\sigma_1^2}{m}.$$

- Nalezení vektoru s maximální variancí je ekvivalentní maximalizaci tzv. Rayleighova kvocientu

$$\sigma_1^2 = \max_{\vec{v} \neq \vec{0}} \frac{\vec{v}^{\mathrm{T}} X^{\mathrm{T}} X \vec{v}}{\vec{v}^{\mathrm{T}} \vec{v}}, \qquad \operatorname{argmax}_{\vec{v} \neq \vec{0}} \frac{\vec{v}^{\mathrm{T}} X^{\mathrm{T}} X \vec{v}}{\vec{v}^{\mathrm{T}} \vec{v}}$$

- Normalizovaná proměnná $\vec{u}_1 = \frac{1}{\sigma_1} X \vec{v}$ se nazývá normalizovaná hlavní komponenta v X.
- Pokud máme nalezen vektor největší variance ve vzorku, obvykle chceme pokračovat a najít vektor druhé největší variance ve vzorku, který je kolmý k tomu prvnímu.
- Toto je provedeno výpočtem vektoru redukované matice $X - \sigma_1 \vec{u}_1 \vec{v}_1^{\mathrm{T}}$.
- Pokračováním tohoto procesu můžeme najít všechny hlavní komponenty, tedy vypočítáme singulární vektory.

Výpočet SVD

Vstup: matice A je $m \times n$ a $m \geq n$.
První krok: **Householderova bidiagonalizace** – zredukujeme A na horní bidiagonální matici pomocí Householderových transformací zleva a zprava:

$$A = H \begin{pmatrix} B \\ 0 \end{pmatrix} W^{\mathrm{T}} \qquad B = \begin{pmatrix} \alpha_1 & \beta_1 & & & & \\ & \alpha_2 & \beta_2 & & & \\ & & \ddots & \ddots & & \\ & & & & \alpha_{n-1} & \beta_{n-1} \\ & & & & & \alpha_n \end{pmatrix}.$$

Toto se dělá, protože

- singulární hodnoty matice A odpovídají odmocninám vlastních hodnot matice $A^{\mathrm{T}}A$.
- B sestrojíme pomocí podobnostních transformací, takže $B^{\mathrm{T}}B$ bude mít stejné vlastní hodnoty $A^{\mathrm{T}}A$.
- $B^{\mathrm{T}}B$ je tridiagonální, takže je snadnější najít její vlastní hodnoty.

Definice 5

Householderova transformace (Householderův reflektor): Matice ve tvaru

$$H = I - 2\vec{w}\vec{w}^{\mathrm{T}}, \qquad \text{kde } \|\vec{w}\|_2 = 1 \; .$$

- Vektor $H\vec{x}$ reprezentuje zrcadlový odraz vektoru $\vec{x}$ vzhledem k nadrovině kolmé k vektoru $\vec{w}$.
- Householderova transformace je symetrická a ortogonální. Takže máme

$$H = H^{\mathrm{T}} = H^{-1}$$

Nechť $\vec{x}$ a $\vec{y}$ jsou stejně velké vektory.
Můžeme vytvořit HT t.ž. $\vec{=}H\vec{x}$:

$$\vec{w} = \frac{\vec{x} - \vec{y}}{\|\vec{x} - \vec{y}\|_2}$$

Sestavíme dvě konečné posloupnosti Householderových transformací:

$$H^{(k)} = I - 2\vec{x}^{(k)}\vec{x}^{(k)\mathrm{T}} \qquad k = 1, 2, \ldots, n$$
$$W^{(k)} = I - 2\vec{y}^{(k)}\vec{y}^{(k)\mathrm{T}} \qquad k = 1, 2, \ldots, n-2,$$

kde $\vec{x}^{(k)\mathrm{T}}\vec{x}^{(k)} = \vec{y}^{(k)\mathrm{T}}\vec{y}^{(k)} = 1$ tak, aby

$$\underbrace{H^{(n)} \cdots H^{(1)}}_{H} A \underbrace{W^{(1)} \cdots W^{(n-2)}}_{W^{\mathrm{T}}} = \left(\begin{array}{ccccc} \alpha_1 & \beta_1 & & & \\ & \alpha_2 & \beta_2 & & \\ & & \ddots & \ddots & \\ & & & \alpha_{n-1} & \beta_{n-1} \\ & & & & \alpha_n \\ \hline 0 & & \cdots & & 0 \\ \vdots & & & & \vdots \\ 0 & & \cdots & & 0 \end{array}\right)$$

Díky vlastnostem Householderových transformací máme

$$A = H \begin{pmatrix} B \\ 0 \end{pmatrix} W^{\mathrm{T}} \qquad \text{a} \qquad \begin{pmatrix} B \\ 0 \end{pmatrix} = HAW^{\mathrm{T}}.$$

Pokud položíme $A^{(1)} = A$ a definujeme

$$\begin{aligned} A^{(k+\frac{1}{2})} &= H^{(k)} A^{(k)} & \qquad (k = 1, 2, \ldots, n) \\ A^{(k+1)} &= A^{(k+\frac{1}{2})} W^{(k)} & \qquad (k = 1, 2, \ldots, n) \end{aligned}$$

Protože během této redukce používáme pouze ortogonální transformace, matice B (resp. $\begin{pmatrix} B \\ 0 \end{pmatrix}$) tytéž singulární hodnoty jako A.

Nechť σ je singulární hodnota A se singulárními vektory $\vec{u}$ a $\vec{v}$.
Pak $A\vec{v} = \sigma\vec{u}$ je ekvivalentní

$$\begin{pmatrix} B \\ 0 \end{pmatrix} \underline{\vec{v}} = \sigma \underline{\vec{v}} \qquad \text{kde} \quad \underline{\vec{v}} = W^{\mathrm{T}}\vec{v}, \underline{\vec{u}} = H^{\mathrm{T}}\vec{u}$$

Takže pokud máme SVD

$$\begin{pmatrix} B \\ 0 \end{pmatrix} = \hat{U}\Sigma\hat{V}^{\mathrm{T}}$$

pak

$$A = H\hat{U}\Sigma\hat{V}^{\mathrm{T}}W^{\mathrm{T}},$$

tj. $U = H\hat{U}$ a $V = \hat{W}V$.

Ilustrace Householderovy bidiagonalizační procedury na malé 6×5 matici: Násobením zleva chceme dostat

$$A^{(1+\frac{1}{2})} = H^{\mathrm{T}(1)} \begin{pmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{pmatrix} = \begin{pmatrix} * & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{pmatrix}$$

- chceme, aby ta HT způsobila vynulování hodnot v prvním sloupci, až na první.
- tj. aby v $\vec{y} = H\vec{x}$ byl $\vec{y}$ první sloupec A a $\vec{x}$ byl vektor $(\|y\|_2, 0, \ldots, 0)^{\mathrm{T}}$.
- a podle toho určíme HT: $\vec{w} = \frac{\vec{x}-\vec{y}}{\|\vec{x}-\vec{y}\|_2}$

Následně další transformací zprava chceme dostat nulové prvky v prvním řádku(od sloupce 3 po n).
Abychom toho dosáhli, zvolíme:

$$\mathbb{R}^{5\times 5} \ni W^{(1)} = \begin{pmatrix} 1 & 0 \\ 0 & Z_1 \end{pmatrix}$$

kde Z_1 je HT.
Zjevně taková transformace nemění prvky v prvním sloupci, takže nuly, které jsme v předchozím kroku dostali do prvního sloupce, tam zůstanou:

$$H^{\mathrm{T}(1)} A W^{(1)} = \begin{pmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{pmatrix} W^{(1)} = \begin{pmatrix} \times & * & 0 & 0 & 0 \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \\ 0 & * & * & * & * \end{pmatrix} =: A^{(2)}.$$

Analogicky pokračujeme dál.

QR-algoritmus pro symetrickou tridiagonální matici

QR dekompozice matice A je rozklad matice A na

$$A = QR,$$

kde

- Q je ortogonální matice;
- R je horní trojúhelníková matice.

Všimněme si, že

$$A' = RQ = Q^{-1}QRQ = Q^{-1}AQ.$$

Tj. A' a A jsou podobné matice – a tedy mají stejné vlastní hodnoty.

- V našem případě to budeme aplikovat na symetrickou tridiagonální matici T a T' bude opět symetrická tridiagonální matice – postupně budeme konvergovat k diagonální matici.

Redukujeme matici T na diagonální matici

$$T \mapsto \Lambda Q^{\mathrm{T}} T Q, \quad Q = Q_1 Q_2 \ldots,$$

kde

- $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n)$.
- Matice Q_i budou ortogonální; budou konstruovány pomocí rovinných rotací.

Neexistuje ale konečný algoritmus pro výpočet Λ.
Počítáme sekvenci matic

$$T_0 := T, \quad T_i = Q_i^{\mathrm{T}} T_{i-1} Q_i, \quad i = 1, 2, \ldots$$

tak, že konverguje na diagonální matici

$$\lim_{i \to \infty} T_i = \Lambda.$$

První verze QR-algoritmu pro symetrickou tridiagonální matici $T \in \mathbb{R}^{n \times n}$

```
for i=1:maxit
  [Q,R]=qr(T)
  T=R*Q
```

- v dalším využijeme tzv. posunů pro urychlení konvergence

Wilkinsonův posun

```
for i=1:maxit
  mu=wilkshift(T(n-1:n,n-1:n));
  [Q,R]=qr(T-mu*I)
  T=R*Q+mu*I

function mu=wilkshift(T);
  l = eig(T);
  if abs(l(1)-T(2,2))<abs(l(2)-T(2,2))
    mu=l(1);
  else
    mu=1(2);
```

- Vidíme, že je nejdříve vypočítána QR dekompozice posunuté matice $QR = T - \mu I$ a pak je posun přidán zpět $T = RQ + \mu I$.
- Posun je ta vlastní hodnota 2×2 submatice v pravém dolním rohu, která je blíže t_{nn}. Tomuto se říká Wilkinsonův posun.

Jakmile mimodiagonální prvky v 2×2 submatici v pravém dolním rohu budou blízko nule, algoritmus pokračuje vypuštěním této vlastní hodnoty hodnoty a sníží dimenzi zpracovávané matice o 1.

```
function [D,it] = qrtrid(T)
  n=size(T,1)
  it=0;
  for i=n:-1:3
    while abs(T(i-1,i)) > (abs(T(i,i))+abs(T(i-1,i-1))) *eps
      it=it+1
      mu=wilkshift(T(i-1:i,i-1:i))
      [Q,R]=qr(T(i:i,i:i))-mu*eye(i)
      T=R*Q+mu*eye(i)
    D(i)=T(i,i)
  D(1:2)=eig(T(1:2,1:2))'
```

Jak počítat QR dekompozici:

- Householderovy transformace
- (efektivněji) rotace G

$$G(i,j,\vartheta) := \begin{array}{cc} \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} & \begin{array}{l} \\ \\ \leftarrow i \\ \\ \leftarrow j \\ \\ \\ \end{array} \\ \begin{array}{ccccccc} & & \uparrow & & \uparrow & & \\ & & i & & j & & \end{array} & \end{array}$$

kde $s = \sin(\vartheta)$ a $c = \cos(\vartheta)$.

Pro $\vec{y} = G(i, j, \vartheta)\vec{x}$ máme

$$y_k = \begin{cases} cx_i - sx_j & \text{pro } k = i \\ sx_i + cx_j & \text{pro } k = j \\ x_k & \text{jinak} \end{cases}$$

Můžeme vynutit vynulování y_j nastavením

$$c = \frac{x_i}{\sqrt{|x_i|^2 + |x_j|^2}} \quad s = \frac{-x_j}{\sqrt{|x_i|^2 + |x_j|^2}}$$

$$H = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix} \xrightarrow{G(1,2,\vartheta_1)^*\cdot} \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix}$$

$$\xrightarrow{G(2,3,\vartheta_2)^*\cdot} \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix} \xrightarrow{G(3,4,\vartheta_3)^*\cdot} \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix} = R$$

Rotace G jsou ortogonální matice, jejich součin pak tvoří matici Q.

Pro ten tridiagonální případ (s posunem)

$$R = G_{n-1}^{\mathrm{T}} \cdots G_1^{\mathrm{T}} (T^{(0)} - \mu I) = \begin{pmatrix} \times & \times & + & & & \\ 0 & \times & \times & + & & \\ & 0 & \times & \times & + & \\ & & 0 & \times & \times & + \\ & & & 0 & \times & \times \\ & & & & 0 & \times \end{pmatrix}$$

Pak ty rotace aplikujeme zase zprava: $RG_1 \cdots G_{n-1}$.
Výsledek po dvou krocích:

$$\begin{pmatrix} \times & \times & \times & & & \\ + & \times & \times & \times & & \\ & + & \times & \times & \times & \\ & & & \times & \times & \times \\ & & & & \times & \times \\ & & & & & \times \end{pmatrix}$$

Předtím vynulované prvky se zase zaplňují.

Po $n-1$ krocích dostáváme

$$T^{(1)} = RG_1 \cdots G_{n-1} + \mu I \begin{pmatrix} \times & \times & \times & & & \\ + & \times & \times & \times & & \\ & + & \times & \times & \times & \\ & & + & \times & \times & \times \\ & & & + & \times & \times \\ & & & & + & \times \end{pmatrix}$$

Udělali jsme vlastně podobností transformaci $Q = G_1 G_2 \cdots G_{n-1}$ a s použitím $R = Q^{\mathrm{T}}(T^{(0)} - \mu I)$ můžeme psát

$$T^{(1)} = RQ + \mu I = Q^T (T^{(0)} - \mu I) Q + \mu I = Q^{\mathrm{T}} T^{(0)} Q,$$

takže $T^{(1)}$ je také symetrická.