SVD rozklad

Vypracovali:

Adam Podlas (pod0107)
Jan Vargovský (var0065)
Libor Polehňa (pol0306)

Na vstup dostaneme čtvercovou bidiagonální matici B. Iterativně aplikujeme Golub-Kahan algoritmus, jehož výstupem jsou matice U, B', V, takové, že platí $U^T B' V = B$. U a V jsou ortogonální transformační matice. Iterace provádíme, dokud nejsou na horní bidiagonále čísla, jejichž hodnota je zanedbatelná, a dají se brát jako nula - menší než ε (v našem případě 10^{-12}).

Golub-Kahan algoritmus:

1. Na diagonále matice B jsou hodnoty d_1-d_n, na horní bidiagonále f_1-f_{n-1}.
2. Z matice B vytvoříme matici T 2x2 tak, že:

$$T = \begin{pmatrix} d_{n-1}^2 + f_{n-2}^2 & d_{n-1} f_{n-1} \\ d_{n-1} f_{n-1} & d_n^2 + f_{n-1}^2 \end{pmatrix}$$

3. Z matice T vypočítáme vlastní čísla λ_1 a λ_2.
4. Vybereme vlastní číslo λ_i, které je nejblíž k hodnotě $d_{n-1}^2 + f_{n-2}^2$
5. Vypočítáme c = cos(θ) a s = cos(θ) tak, že:

$$\begin{pmatrix} c & s \\ -s & c \end{pmatrix}^T \begin{pmatrix} d_1^2 - \lambda_i \\ d_1 f_1 \end{pmatrix} = \begin{pmatrix} x \\ 0 \end{pmatrix}$$

6. Vytvoříme Givensovu rotaci G1 = G(1, 2, θ) a aplikujeme ji na matici B.

$$G_1 = \begin{pmatrix} c & -s & 0 & \cdots & & 0 \\ s & c & & & & \\ 0 & & 1 & & & \\ \vdots & & & & & \vdots \\ & & & & \ddots & 0 \\ 0 & & & \cdots & 0 & 1 \end{pmatrix}, B \leftarrow BG_1$$

7. Počítáme Givensovy rotaci U a V a střídavě je aplikujeme zleva a zprava, abychom posunovali nenulové číslo na konec matice B. Pro větší efektivitu při rotování matice B násobíme pouze 2 změněné řádky/sloupce, nikoliv celé matice.

$$U_1 = \begin{pmatrix} c & s & 0 & \cdots & & 0 \\ -s & c & & & & \\ 0 & & 1 & & & \\ \vdots & & & & & \vdots \\ & & & & \ddots & 0 \\ 0 & & & \cdots & 0 & 1 \end{pmatrix}, B \leftarrow U_1 B$$

$$V_2 = \begin{pmatrix} 1 & 0 & & \cdots & & 0 \\ 0 & c & -s & & & \\ & s & c & & & \\ & & & 1 & & \vdots \\ & & & & \ddots & 0 \\ 0 & & & \cdots & 0 & 1 \end{pmatrix}, B \leftarrow BV_2$$

$$U_2 = \begin{pmatrix} 1 & 0 & & \cdots & & 0 \\ 0 & c & s & & & \\ & -s & c & & & \\ \vdots & & & 1 & & \vdots \\ & & & & \ddots & 0 \\ 0 & & & \cdots & 0 & 1 \end{pmatrix}, B \leftarrow U_2 B$$

Tak až do U_{n-1} a V_{n-1}.

8. Kroky 2-7 opakujeme, dokud matice B neobsahuje na horní bidiagonále pouze nuly (čísla menší, než ε.
9. Na konci algoritmus vrací jak upravenou matici B, tak transformační matice U a V (U = $U_1U_2 \ldots U_{n-1}$; V = $V_1V_2 \ldots V_{n-1}$), tak že B = $U^T B' V$.