▶ Structure exploiting Lanczos method for Hadamard product of low-rank matrices

Prof. D. Kressner
A. Cortinovis

The Hadamard product, or component-wise product, is defined as

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} * \begin{bmatrix} b_{11} & \dots & b_{1n} \\ \vdots & \vdots & \vdots \\ b_{n1} & \dots & b_{nn} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & \dots & a_{1n}b_{1n} \\ \vdots & \vdots & \vdots \\ a_{n1}b_{n1} & \dots & a_{nn}b_{nn} \end{bmatrix}$$

and is a fundamental building block of many algorithms in scientific computing and data analysis. This is partly because Hadamard products correspond to products of multivariate functions. For example if $a_{ij} = f(x_i, y_j)$ and $b_{ij} = g(x_i, y_j)$, it is known that smoothness of the functions f, g implies that these matrices can be well approximated by low-rank matrices, and their Hadamard product also admits a good low-rank approximation. This property is not fully reflected on the algebraic side; the Hadamard product generically multiplies (and thus often drastically increases) ranks. However, exploiting the structure of the Hadamard product to create fast matrix-vector multiplication, which can be used inside the Lanczos algorithm [1] we can design computationally efficient algorithms for Hadamard products.

Let A and B be $m \times n$ matrices with low-rank structures

$$A = U_A \Sigma_A V_A^T, \quad B = U_B \Sigma_B V_B^T,$$

with $\Sigma_A \in \mathbb{R}^{k_A \times k_A}, \Sigma_B \in \mathbb{R}^{k_B \times k_B}$ and $k_A, k_B \ll \min\{m, n\}$.

a) Read Section 2.1 of [2] in order to understand that the Hadamard product of A and B admits the following representation

$$A * B = \left(U_A^T \odot U_B^T\right)^T (\Sigma_A \otimes \Sigma_B) \left(V_A^T \odot V_B^T\right), \tag{1}$$

where $\odot$ represents the Khatri-Rao product, and $\otimes$ the Kronecker product (all definitions and properties in [2, Section 2.1]).

b) Given $x \in \mathbb{R}^m$ implement — by exploiting (1) — the matrix-vector multiplication $(A * B)x$ without forming the matrix $A * B$ explicitly.

c) Apply the Lanczos algorithm to matrix $(A * B)(A * B)^T$ to get the (reduced) SVD of $(A * B)$; see [2, Section 4.1]. Use the fast matrix-vector multiplication from the previous point.

d) For $M = 1$ and $N = 2$, create matrices A and B by evaluating functions

$$f(x, y) = \frac{1}{x + y} \text{ and } g(x, y) = \frac{1}{\sqrt{x^2 + y^2}},$$

respectively, on the grid $\{0.1, 0.2, \dots, M\} \times \{0.1, 0.2, \dots, N\}$, and matrix $C = A * B$. Get the low-rank approximation of A, B and C by truncating their singular values lower than $\varepsilon = 10^{-4}$. What is the rank of this low-rank approximation of C? Compare to the low-rank representation (1).

e) Apply the method from c) to A and B from task d), for $M = N = 50, 100, 150, 200, 250, 300$ and $\varepsilon = 10^{-8}$. Plot the times needed to get the low-rank approximation of C this way, and directly (run `svd` on C). Calculate the errors $\|A * B - C\|_F$.

▶ References

[1] Horst D. Simon, Hongyuan Zha. Low-Rank Matrix Approximation Using the Lanczos Bidiagonalization Process with Applications. *SIAM Journal on Scientific Computing*, 21(6), 2257–2274.

[2] Daniel Kressner, Lana Periša. Recompression of Hadamard Products of Tensors in Tucker Format. *SIAM Journal on Scientific Computing*, 39(5), A1879–A1902.