MANCHESTER 1824

The University of Manchester

NLA GROUP MANCHESTER

Computing the Square Root of a Low-Rank Perturbation of the Scaled Identity Matrix

Massimiliano Fasi[1], Nicholas J. Higham[2], **Xiaobo Liu**[2]

[1]Department of Computer Science, Durham University, UK
[2]Department of Mathematics, The University of Manchester, UK

7th IMA Conference on Numerical Linear Algebra and Optimization, June 30, 2022

Matrix Square Root

- X is a square root of $A \in \mathbb{C}^{n\times n} \iff X^2 = A$.
- Number of square roots may be zero, finite or infinite.

$$\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \pm 1 & 0 \\ 0 & \pm 1 \end{bmatrix}^2 = \begin{bmatrix} a & b \\ c & -a \end{bmatrix}^2, \quad a^2 + bc = 1.$$

Definition

For A with no eigenvalues on $\mathbb{R}^- = \{x \in \mathbb{R} \colon x \le 0\}$ the principal square root $A^{1/2}$ is unique square root X with spectrum in the region

$$-\pi/2 < \arg(\lambda(X)) < \pi/2.$$

Matrix Square Roots in Machine Learning

Shampoo (Gupta, Koren, & Singer, 2018): preconditioned stochastic gradient method for second-order optimization. Needs

$$L_t^{-1/2p} G_t R_t^{-1/2q}, \quad t = 1, \dots, \ell,$$

where

$$L_t = \alpha I_n + \sum_{s=1}^{t} G_s G_s^*, \quad R_t = \alpha I_k + \sum_{s=1}^{t} G_s^* G_s, \quad \alpha > 0,$$

and $G_1, \dots, G_t \in \mathbb{R}^{n \times k}$ are of rank at most r.

Visual recognition (Lin, 2020): features aggregated via bilinear pooling, using the square root of

$$A = \alpha I_n + \frac{1}{k} \sum_{i=1}^{k} x_i x_i^*, \quad \alpha > 0.$$

Computing the (principal) square root

$$A = \alpha I_n + UV^*, \quad \alpha \in \mathbb{C},\ U, V \in \mathbb{C}^{n\times k},\ k \le n,\ \Lambda(A) \cap \mathbb{R}^- = \emptyset.$$

Theorem (Harris, 1993; Higham, 2008)

Let $U, V \in \mathbb{C}^{n\times k}$ with $k \le n$ and assume that V^*U is nonsingular. Then

$$f(A) = f(\alpha) I_n + U(V^*U)^{-1}\Big(f(\alpha I_k + V^*U) - f(\alpha) I_k\Big) V^*.$$

Explicit Formula for General Functions

Theorem (Harris, 1993; Higham, 2008)

Let $U, V \in \mathbb{C}^{n\times k}$ with $k \le n$ and assume that V^*U is nonsingular. Then

$$f(A) = f(\alpha)I_n + U(V^*U)^{-1}\Big(f(\alpha I_k + V^*U) - f(\alpha)I_k\Big)V^*.$$

- $f(A)$ is a perturbation of rank at most k of the scaled identity matrix.
- $f(A)$ can be computed by evaluating f and the inverse at two $k \times k$ matrices.

Problem: requires V^*U nonsingular, which is not required for f to exist!

Sherman–Morrison–Woodbury Formula

If $U, V \in \mathbb{C}^{n\times k}$ and $I_k + V^*A^{-1}U$ is nonsingular then

$$(A + UV^*)^{-1} = A^{-1} - A^{-1}U(I_k + V^*A^{-1}U)^{-1}V^*A^{-1}.$$

- Does not require V^*U nonsingular!

Sherman–Morrison–Woodbury Formula

If $U, V \in \mathbb{C}^{n\times k}$ and $I_k + V^*A^{-1}U$ is nonsingular then

$$(A + UV^*)^{-1} = A^{-1} - A^{-1}U(I_k + V^*A^{-1}U)^{-1}V^*A^{-1}.$$

- Does not require V^*U nonsingular!

Derivation

Obtained using $A + UV^* = A(I_n + A^{-1}U \cdot V^*)$ and

$$(I_n + BC)^{-1} = I_n - B(I_k + CB)^{-1}C, \quad B \in \mathbb{C}^{n\times k}, C \in \mathbb{C}^{k\times n}$$

$$\begin{aligned} I &= I + BC - (I + BC)B(I + CB)^{-1}C \\ &= I + BC - B(I + CB)(I + CB)^{-1}C \\ &= I + BC - BC \end{aligned}$$

Square Root Update Formula

Theorem

Let $U, V \in \mathbb{C}^{n\times k}$ with $k \leq n$ have full rank and let the matrix $A = \alpha I_n + UV^*$ have no eigenvalues on $\mathbb{R}^-$. Then

$$A^{1/2} = \alpha^{1/2} I_n + U\Big((\alpha I_k + V^* U)^{1/2} + \alpha^{1/2} I_k\Big)^{-1} V^*.$$

- Does not contain the factor $(V^*U)^{-1}$!

Square Root Update Formula

Theorem

Let $U, V \in \mathbb{C}^{n\times k}$ with $k \le n$ have full rank and let the matrix $A = \alpha I_n + UV^*$ have no eigenvalues on $\mathbb{R}^-$. Then

$$A^{1/2} = \alpha^{1/2} I_n + U\Big((\alpha I_k + V^*U)^{1/2} + \alpha^{1/2} I_k\Big)^{-1} V^*.$$

- Does not contain the factor $(V^*U)^{-1}$!

Key idea: Taking for f the square root in the $f(A)$ formula:

$$A^{1/2} = \alpha^{1/2} I_n + U(V^*U)^{-1}\Big((\alpha I_k + V^*U)^{1/2} - \alpha^{1/2} I_k\Big) V^*$$

and use

$$(\alpha I_k + V^*U)^{1/2} - \alpha^{1/2} I_k = V^*U\Big((\alpha I_k + V^*U)^{1/2} + \alpha^{1/2} I_k\Big)^{-1},$$

which is essentially $(\sqrt{1+x} - 1)/x = 1/(\sqrt{1+x} + 1)$!

Schur Method for the Square Root

- Compute Schur decomp. $A = QTQ^*$.

- Expand $U^2 = T$, where U is upper triangular, for primary square root:

$$u_{ii}^2 = t_{ii},$$

$$(u_{ii} + u_{jj})u_{ij} = t_{ij} - \sum_{k=i+1}^{j-1} u_{ik}u_{kj}.$$

U is found either a column or a superdiagonal at a time.

- $\sqrt{A} = QUQ^*$.

- **Schur decomp. may not be available**
  - In low precisions or on custom hardware. No nonsymmetric dense eigensolver in NVIDIA cuSOLVER library.
  - In multiprecision environments. E.g., Julia (Version 1.7.1) and MATLAB Symbolic Math Toolbox in VPA arithmetic.
- **Matrix Iterations** are attractive. Newton–Schulz iteration is being used in deep learning.

Newton methods

The (scaled) DB iteration is (Denman & Beavers, 1976)

$$X_{i+1} = \frac{1}{2}\left(\mu_i X_i + \mu_i^{-1} Y_i^{-1}\right), \quad X_0 = A,$$
$$Y_{i+1} = \frac{1}{2}\left(\mu_i Y_i + \mu_i^{-1} X_i^{-1}\right), \quad Y_0 = I,$$

for some scaling parameter $\mu_i \in \mathbb{R}$.

Product form DB (Cheng, Higham, Kenney, & Laub, 2001)

$$M_{i+1} = \frac{1}{2}\left(I + \frac{\mu_i^2 M_i + \mu_i^{-2} M_i^{-1}}{2}\right), \quad M_0 = A,$$
$$X_{i+1} = \frac{1}{2}\mu_i X_i\left(I + \mu_i^{-2} M_i^{-1}\right), \quad X_0 = A.$$
$$Y_{i+1} = \frac{1}{2}\mu_i Y_i\left(I + \mu_i^{-2} M_i^{-1}\right), \quad Y_0 = I.$$

The Structured DB Iteration

Facts: $A^{1/2}$ is a rank-k prturbation to $\alpha^{1/2} I_n$, and

$$A^{1/2} = \alpha^{1/2} I_n + U\big((\alpha I_k + V^* U)^{1/2} + \alpha^{1/2} I_k\big)^{-1} V^*.$$

is in the form $A^{1/2} = \alpha^{1/2} I_n + UZV^*$ for some $Z \in \mathbb{C}^{k\times k}$.

Key idea: For $A = \alpha I_n + UV^*$, write the DB iterates in the form

$$\begin{aligned} X_i &= \beta_i I_n + UB_i V^*, \quad \beta_i \in \mathbb{C}, \quad B_i \in \mathbb{C}^{k\times k}, \\ Y_i &= \gamma_i I_n + UC_i V^*, \quad \gamma_i \in \mathbb{C}, \quad C_i \in \mathbb{C}^{k\times k}, \end{aligned}$$

and use SMW

$$(A + UV^*)^{-1} = A^{-1} - A^{-1} U (I_k + V^* A^{-1} U)^{-1} V^* A^{-1}.$$

for X_i^{-1} and Y_i^{-1}.

The Structured DB Iteration

For $A = \alpha I_n + UV^*$, the DB iterates are

$$X_i = \beta_i I_n + UB_i V^*, \quad \beta_i \in \mathbb{C}, \quad B_i \in \mathbb{C}^{k\times k},$$
$$Y_i = \gamma_i I_n + UC_i V^*, \quad \gamma_i \in \mathbb{C}, \quad C_i \in \mathbb{C}^{k\times k},$$

where $\beta_0 = \alpha$, $B_0 = I_k$, $\gamma_0 = 1$, $C_0 = 0$, and

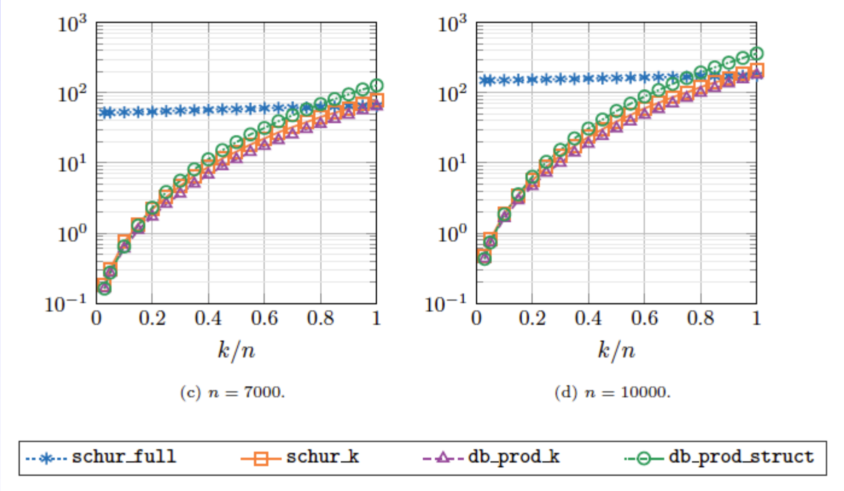
$$\beta_{i+1} = \frac{\mu_i\beta_i + (\mu_i\gamma_i)^{-1}}{2},$$
$$B_{i+1} = \frac{1}{2}\left(\mu_i B_i - (\mu_i\gamma_i)^{-1} C_i(\gamma_i I_k + V^* UC_i)^{-1}\right),$$
$$\gamma_{i+1} = \frac{\mu_i\gamma_i + (\mu_i\beta_i)^{-1}}{2},$$
$$C_{i+1} = \frac{1}{2}\left(\mu_i C_i - (\mu_i\beta_i)^{-1} B_i(\beta_i I_k + V^* UB_i)^{-1}\right).$$

Methods

$$A^{1/2} = \alpha^{1/2} I_n + U\Big((\alpha I_k + V^* U)^{1/2} + \alpha^{1/2} I_k\Big)^{-1} V^*$$

- Schur method on A.
- Schur method on $(\alpha I_k + V^* U)^{1/2}$.
- DB method on $(\alpha I_k + V^* U)^{1/2}$.
- Structured DB method on A.

Numerical Experiment



$A = \alpha I + UV^*$ with $\alpha = 0.1$ and U, V normal$(0, n^{-2})$.

Concluding Remarks

- $\sqrt{A}$ is enjoying new interest in machine learning applications. Exploiting the structure yields much faster algorithms than the standard Schur method!

- Schur decomposition is not always available, in which case matrix iterations are attractive!

M. Fasi, N. J. Higham and X. Liu. Computing the square root of a low-rank perturbation of the scaled identity matrix. MIMS EPrint 2022.1, Janurary 2022. Revised May 2022.

- Codes available at https://github.com/Xiaobo-Liu/sqrtm-lrpsi

References I

- S. H. Cheng, N. J. Higham, C. S. Kenney, and A. J. Laub.
  Approximating the logarithm of a matrix to specified accuracy.
  SIAM J. Matrix Anal. Appl., 22(4):1112–1125, 2001.

- E. D. Denman and A. N. Beavers, Jr.
  The matrix sign function and computations in systems.
  Appl. Math. Comput., 2:63–94, 1976.

References II

V. Gupta, T. Koren, K. Regan, and Y. Singer.
Shampoo: Preconditioned stochastic tensor optimization.
In Proceedings of the 35th International Conference on Machine Learning, Jennifer Dy and Andreas Krause, editors, volume 80 of Proceedings of Machine Learning Research, Stockholmsmässan, Stockholm Sweden,2018, pages 1842–1850.

L. A. Harris.
Computation of functions of certain operator matrices.
Linear Algebra Appl., 194:31–34, 1993.

N. J. Higham.
Functions of Matrices: Theory and Computation.
Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.

T.-Y. Lin.
Higher-Order Representations for Visual Recognition.
PhD thesis, College of Informationand Computer Sciences, University of Massachusetts Amherst, Massachusetts, USA, Feb. 2020.