Improving on the stability Jennrich's algorithm

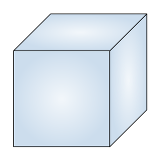
Eric Evert Michiel Vandecappelle Lieven De Lathauwer

KU Leuven kulak

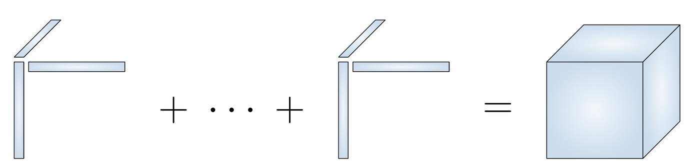
I. Setup: CPD and Jennrich's Algorithm

I.1 Decompose signal into canonical components.

A **tensor** $\mathcal{T}$ is a multiindexed array.

$\mathcal{T} =$  $\in \mathbb{R}^{N_1 \times N_2 \times N_3}$

Canonical **P**olyadic **D**ecomp. expresses $\mathcal{T}$ as minimal sum of rank 1 terms.

$\mathcal{T} = \sum_{r=1}^{R} \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r =$ 

R is the **rank** of $\mathcal{T}$.

Notation: $\mathbf{a}_r, \mathbf{b}_r, \mathbf{c}_r$ are vectors of length N_1, N_2, N_3, respectively.
Simplifying assumption: $N_1 = N_2 = N_3 = R$. I.e. assume $\mathcal{T}$ has low rank.

I.2 Jennrich: Eigenvector decomposition gives CPD.

Key idea: Columns of

$$\begin{pmatrix} \uparrow & & \uparrow \\ \mathbf{b}_1 & \cdots & \mathbf{b}_R \\ \downarrow & & \downarrow \end{pmatrix}^{-\top}$$

are equal to eigenvectors of $\mathbf{T}_k^{-1}\mathbf{T}_\ell$ which in turn are equal to generalized eigenvectors of the **matrix pencil** $(\mathbf{T}_k, \mathbf{T}_\ell)$.

$\Longrightarrow$ Generalized eigenvector decomp. of $(\mathbf{T}_k, \mathbf{T}_\ell)$ leads to CPD of $\mathcal{T}$.

Notation: $\mathbf{T}_k$ is the $R \times R$ matrix $(t_{ijk})_{i,j=1,\dots,R}$.

I.3 Small eigenvalue gaps leads to instability.

Gen. eigenvalues of $(\mathbf{T}_k, \mathbf{T}_\ell)$ are interpreted as points on the unit circle. The pencil $(\mathbf{T}_k, \mathbf{T}_\ell)$ has R generalized eigenvalues.

Small gaps between gen. eigenvalues causes instability in computing gen. eigenvectors. $\Longrightarrow$ Instability of Jennrich's algorithm as R grows.

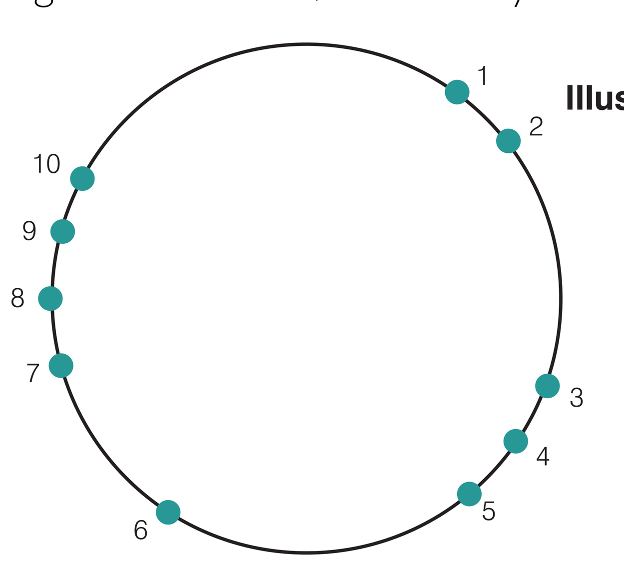


Illustration of generalized eigenvalues of $(\mathbf{T}_k, \mathbf{T}_\ell)$

● = generalized eigenvalue of $(\mathbf{T}_k, \mathbf{T}_\ell)$.

The small gap between generalized eigenvalues 1 and 2 leads to instability in computing the generalized eigenvectors $\mathbf{v}_1$ and $\mathbf{v}_2$.

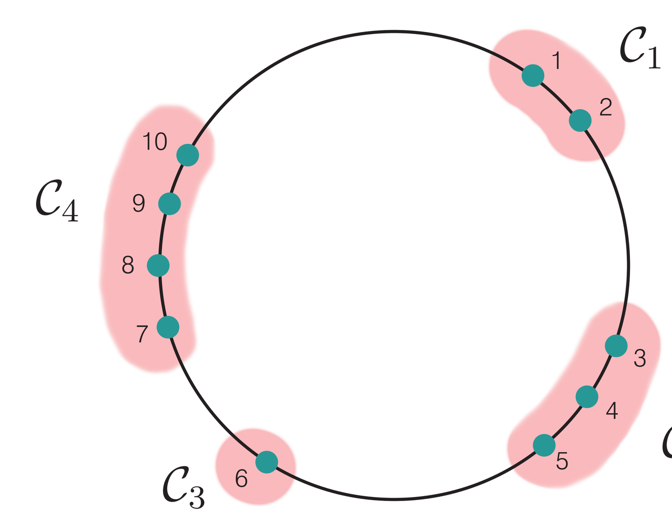
Similar issues occur in the other clusters of generalized eigenvalues.

In fact, using a single pencil to compute a CPD is a fundamental source of instability in Jennrich's algorithm. This effect is quantified by Beltrán, Breiding, and Vannieuwenhoven. GESD combats this effect by using multiple pencils for CPD computation.

II. Generalize EigenSpace Decomp.

II.1 Improve stability by computing eigenspaces corresponding to well separated eigenvalue clusters

Consider following clusters of generalized eigenvalues of $(\mathbf{T}_k, \mathbf{T}_\ell)$.

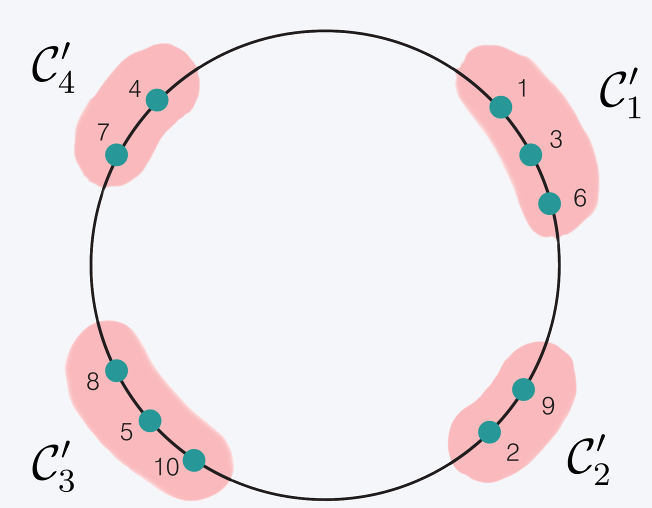


Clusters $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathcal{C}_4$ are well separated so can improve stability by only computing the corresponding eigenspaces $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4$.

Next step: Recover vectors $\mathbf{v}_1, \mathbf{v}_2$ from eigenspace $\mathcal{E}_1 = \text{span}\{\mathbf{v}_1, \mathbf{v}_2\}$.

II.2 Use a new pencil to split eigenspaces!

Consider a new subpencil $(\mathbf{T}_m, \mathbf{T}_n)$ of $\mathcal{T}$. The eigenvectors of this pencil are the same as those of $(\mathbf{T}_k, \mathbf{T}_\ell)$, but the corresponding eigenvalues will lie in new positions on the unit circle.



Now the clusters $\mathcal{C}_1', \mathcal{C}_2', \mathcal{C}_3', \mathcal{C}_4'$ are well separated so compute the corresponding eigenspaces $\mathcal{E}_1', \mathcal{E}_2', \mathcal{E}_3', \mathcal{E}_4'$.

Observe $\mathcal{E}_1 = \text{span}\{\mathbf{v}_1, \mathbf{v}_2\}$ and $\mathcal{E}_1' = \text{span}\{\mathbf{v}_1, \mathbf{v}_3, \mathbf{v}_6\}$. Thus $\mathbf{v}_1 = \mathcal{E}_1 \cap \mathcal{E}_1'$.

II.3 GESD recursively deflates tensor rank.

In practice, GESD recursively writes $\mathcal{T}$ as a sum of tensors of reduced rank.

In our example, GESD uses $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4$ to write the rank 10 tensor $\mathcal{T}$ as

$$\mathcal{T} = \mathcal{T}^1 + \mathcal{T}^2 + \mathcal{T}^3 + \mathcal{T}^4$$

where $\mathcal{T}^1, \mathcal{T}^2, \mathcal{T}^3$ and $\mathcal{T}^4$ have ranks 2, 3, 1 and 4, respectively. $\mathcal{T}^1$ can then be decomposed into a sum of rank 1 tensors using the pencil $(\mathcal{T}^1_m, \mathcal{T}^1_n)$.

Variations in GESD are possible. E.g. one could compute intersections of eigenspaces as described above rather than working recursively.

III. QZ CPD method: Avoiding inverses

III.1 Jennrich's algorithm computes an unnecessary inverse.

Jennrich's algorithm computes the inverse of the matrix of eigenvectors of a pencil $(\mathbf{T}_k, \mathbf{T}_\ell)$.

Inverse computation can be avoided by considering "joint generalized eigenvalues" instead of eigenvectors.

III.2 QZ decomposition basics.

QZ decomposition is generalization of the Schur decomposition to matrix pencils.

Given a matrix pencil $(\mathbf{T}_k, \mathbf{T}_\ell)$, QZ computes orthogonal $\mathbf{Q}$ and $\mathbf{Z}$ such that

$$\mathbf{Q}\mathbf{T}_k\mathbf{Z}^\top \quad \text{and} \quad \mathbf{Q}\mathbf{T}_\ell\mathbf{Z}^\top$$

are both upper triangular matrices.

Similar to the matrix setting, generalized eigenvalues of $(\mathbf{T}_k, \mathbf{T}_\ell)$ are given by the diagonal entries of $(\mathbf{Q}\mathbf{T}_k\mathbf{Z}^\top, \mathbf{Q}\mathbf{T}_\ell\mathbf{Z}^\top)$

Computing a QZ decomposition is a standard step in a generalized eigenvalue decomposition. E.g. Matlab's eig routine applied to matrix pencils starts with a QZ decomposition.

III.2 A single QZ decomposition recovers a factor matrix!

For generic low rank tensors $\mathcal{T}$, a QZ decomposition of a subpencil can be used to simultaneously upper triangularize all frontal slices of $\mathcal{T}$.

If $\mathbf{Q}, \mathbf{Z}$ are orthogonal matrices such that

$$\mathbf{Q}\mathbf{T}_k\mathbf{Z}^\top \quad \text{and} \quad \mathbf{Q}\mathbf{T}_\ell\mathbf{Z}^\top$$

are both upper triangular matrices, then

$$\mathbf{Q}\mathbf{T}_r\mathbf{Z}^\top$$

is upper triangular for all $r = 1, \dots, R$.

In this case, the jth entry of $\mathbf{c}_r$ is the jth diagonal entry of $\mathbf{Q}\mathbf{T}_r\mathbf{Z}^\top$.

In fact, a second QZ can be used to reveal a second factor matrix.

Extending the matrix pencil case, vectors on the diagonal of the upper triangular tensor $\mathcal{T} \cdot_1 \mathbf{Q} \cdot_2 \mathbf{Z}^\top$ can be naturally interpreted as "joint generalized eigenvalues" of $\mathcal{T}$. In this framework, the joint generalized eigenvectors of $\mathcal{T}$ are equal to the vectors $\mathbf{c}_1, \dots, \mathbf{c}_R$.

III.3 Upper triangular slices leads to triangular factors.

Let $\mathbf{A}$ be a matrix with columns $\mathbf{a}_1, \dots, \mathbf{a}_R$ and similarly define $\mathbf{B}$ and $\mathbf{C}$. If $\mathcal{T}_r$ is upper triangular for each r, then (in an appropriate ordering of columns) $\mathbf{A}$ and $\mathbf{B}^\top$ are both upper triangular matrices. The QZ CPD algorithm then easily follows from

$$\mathbf{T}_r = \mathbf{A}D_r(\mathbf{C})\mathbf{B}^\top \quad \text{for all} \quad r = 1, \dots, R.$$

Here $D_r(\mathbf{C})$ is a diagonal matrix with entries given by the rth row of $\mathbf{C}$.

IV. Numerical results

IV.1 Performance of methods for various tensor ranks.

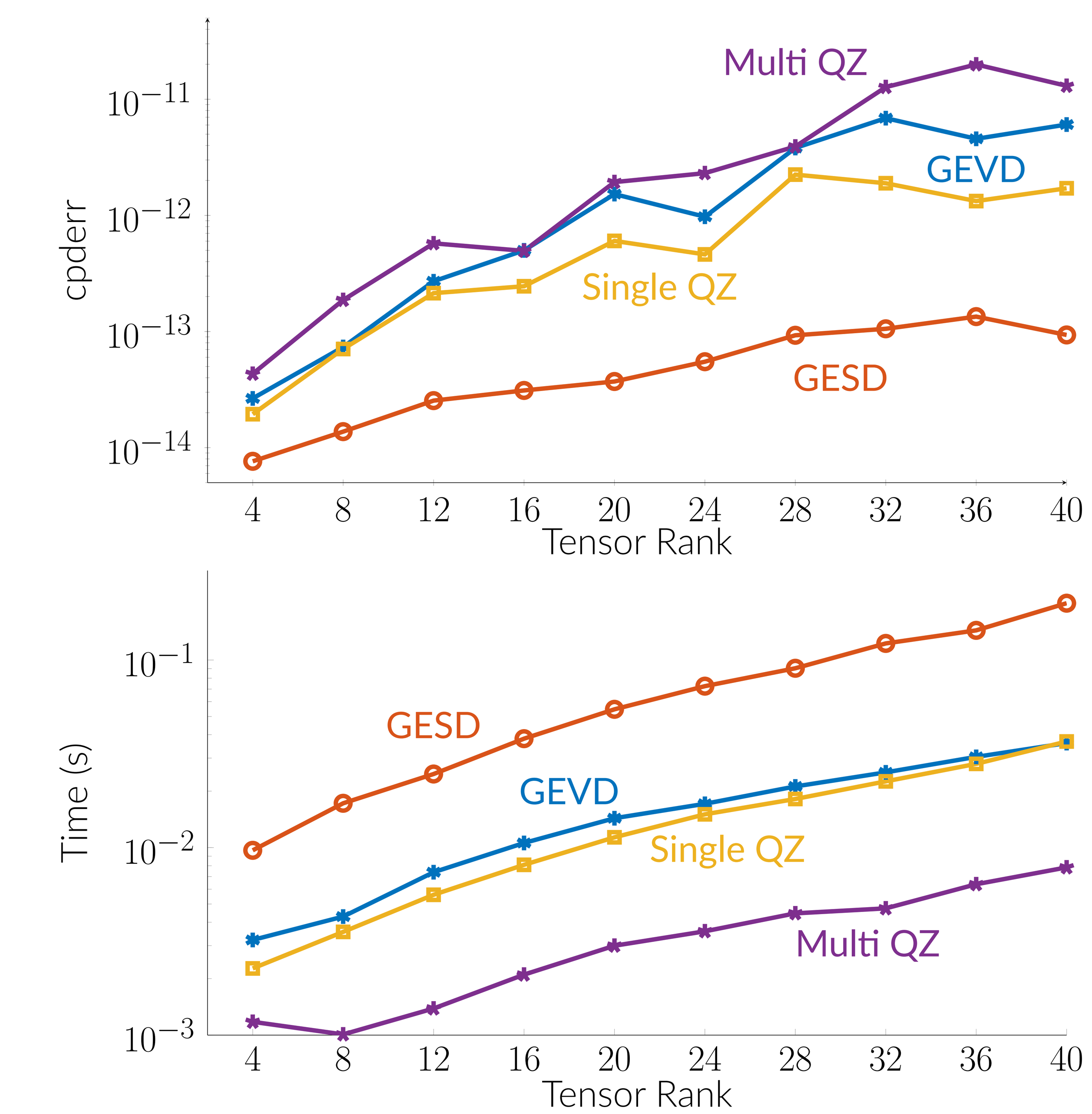


Figure 1. Single QZ is a direct improvement on Jennrich's algorithm (as implemented in Tensorlab's cpd_gevd). GESD is the most accurate but slowest method. Multi QZ is the fastest but least accurate method.

IV.2 Performance against fixed tensor rank.

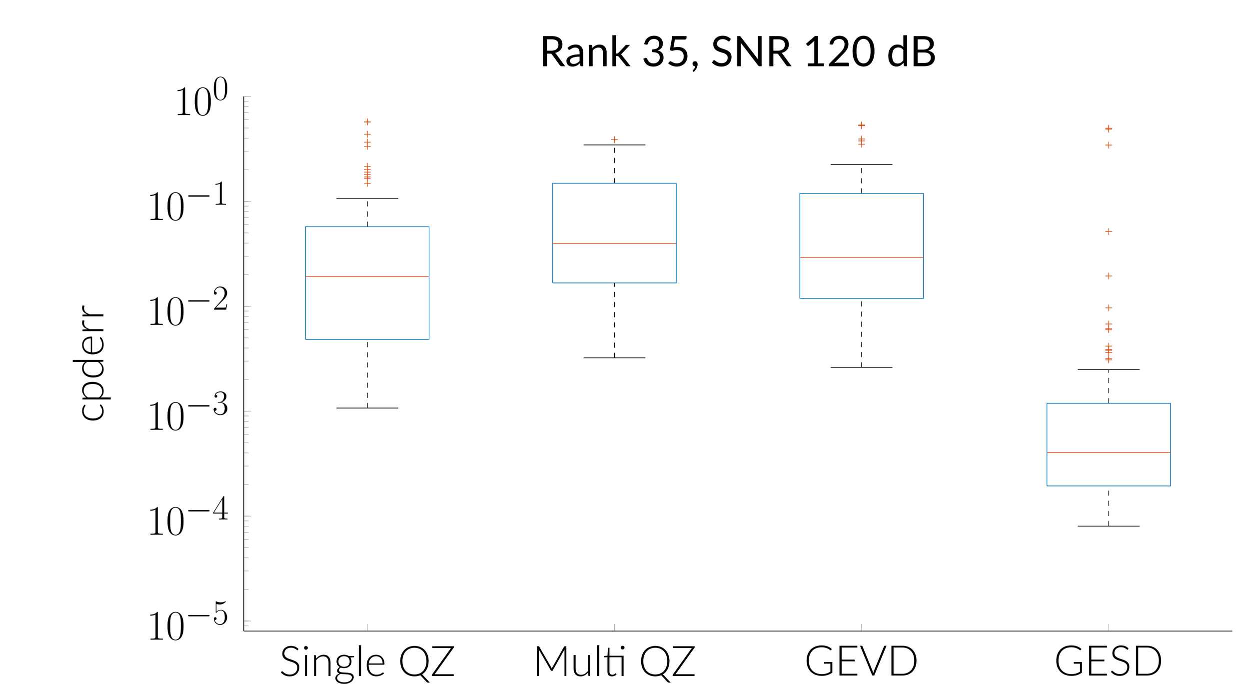


Figure 2. Accuracy against Rank 35 tensors with 120 dB SNR.