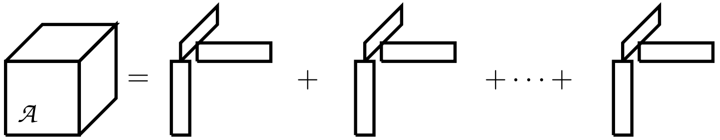
Pencil-based algorithms for the tensor rank decomposition are not stable

Paul Breiding (MPI MiS Leipzig)
Carlos Beltrán (Universidad de Cantabria)
Nick Vannieuwenhoven (KU Lueven)

paulbreiding.org
juliahomotopycontinuation.org

CPD for tensors $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$:

$$\mathcal{A} = \sum_{i=1}^{r} \mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i$$



minimal number of terms needed $= \operatorname{rank}(\mathcal{A})$

Notation

$\mathcal{S} := \{\mathcal{A} \mid \operatorname{rank}(\mathcal{A}) = 1\}$ *are the rank-one tensors.*
$\sigma_r := \{\mathcal{A} \mid \operatorname{rank}(\mathcal{A}) \leq r\}$ *are the tensors of rank at most* r*.*

We assume that w. probability=1 $\mathcal{A} \in \sigma_r$ has a unique decomposition (σ_r is "generically identifiable").

A direct algorithm for order-3 tensors

In some cases, the CPD of third-order tensors can be computed directly via a **generalized eigendecomposition**.

This is also called **Jenrich's algorithm**.

For simplicity, assume that $\mathcal{A} \in \mathbb{R}^{n\times n\times n}$ is of rank n. Say

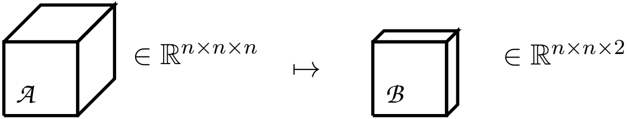
$$\mathcal{A} = \sum_{i=1}^{n} \mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i.$$

The steps are as follows.

1. Choose a matrix $Q \in \mathbb{R}^{n\times 2}$ with orthonormal columns $\mathbf{q}_1, \mathbf{q}_2$.

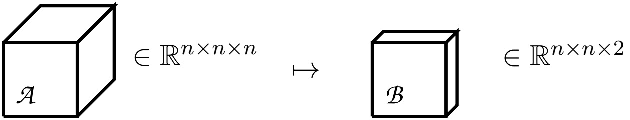
1. Choose a matrix $Q \in \mathbb{R}^{n\times 2}$ with orthonormal columns $\mathbf{q}_1, \mathbf{q}_2$.
2. Compute the **multilinear multiplication**

$$\mathcal{B} = (I, I, Q^T) \cdot \mathcal{A} := \sum_{i=1}^{n} \mathbf{a}_i \otimes \mathbf{b}_i \otimes (Q^T \mathbf{c}_i).$$



1. Choose a matrix $Q \in \mathbb{R}^{n\times 2}$ with orthonormal columns $\mathbf{q}_1, \mathbf{q}_2$.
2. Compute the **multilinear multiplication**

$$\mathcal{B} = (I, I, Q^T) \cdot \mathcal{A} := \sum_{i=1}^{n} \mathbf{a}_i \otimes \mathbf{b}_i \otimes (Q^T \mathbf{c}_i).$$



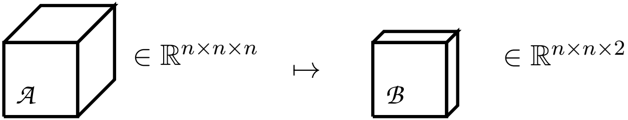
3. The two **slices** X_1 and X_2 of $\mathcal{B}$ are

$$X_j = \sum_{i=1}^{n} \langle \mathbf{q}_j, \mathbf{c}_i \rangle \mathbf{a}_i \otimes \mathbf{b}_i = A \operatorname{diag}(\mathbf{q}_j^T C) B^T$$

where $A = [\mathbf{a}_i]$ and $B = [\mathbf{b}_i]$ and $C = [\mathbf{C}_i]$.

1. Choose a matrix $Q \in \mathbb{R}^{n \times 2}$ with orthonormal columns $\mathbf{q}_1, \mathbf{q}_2$.
2. Compute the **multilinear multiplication**

$$\mathcal{B} = (I, I, Q^T) \cdot \mathcal{A} := \sum_{i=1}^{n} \mathbf{a}_i \otimes \mathbf{b}_i \otimes (Q^T \mathbf{c}_i).$$



3. The two **slices** X_1 and X_2 of $\mathcal{B}$ are

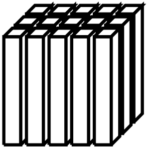
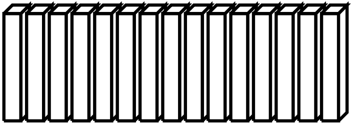
$$X_j = \sum_{i=1}^{n} \langle \mathbf{q}_j, \mathbf{c}_i \rangle \mathbf{a}_i \otimes \mathbf{b}_i = A \operatorname{diag}(\mathbf{q}_j^T C) B^T$$

where $A = [\mathbf{a}_i]$ and $B = [\mathbf{b}_i]$ and $C = [\mathbf{C}_i]$.
Hence, $X_1 X_2^{-1}$ has the following eigenvalue decomposition:

$$X_1 X_2^{-1} = A \operatorname{diag}(\mathbf{q}_1^T C) \operatorname{diag}(\mathbf{q}_2^T C)^{-1} A^{-1}$$

from which A can be found as the matrix of eigenvectors.

4. By a **1-flattening**

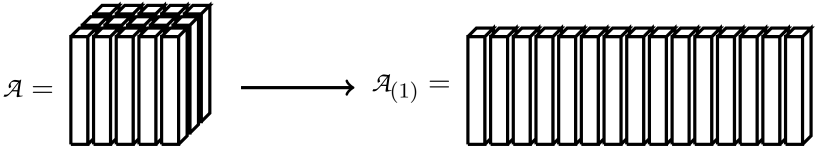
$\mathcal{A} =$  $\longrightarrow$ $\mathcal{A}_{(1)} =$ 

we find

$$\mathcal{A}_{(1)} := \sum_{i=1}^{n} \mathbf{a}_i (\mathbf{b}_i \otimes \mathbf{c}_i)^T = A(B \odot C)^T,$$

where $B \odot C := [\mathbf{b}_i \otimes \mathbf{c}_i]_i \in \mathbb{R}^{n^2 \times n}$.

4. By a **1-flattening**



we find

$$\mathcal{A}_{(1)} := \sum_{i=1}^{n} \mathbf{a}_i (\mathbf{b}_i \otimes \mathbf{c}_i)^T = A(B \odot C)^T,$$

where $B \odot C := [\mathbf{b}_i \otimes \mathbf{c}_i]_i \in \mathbb{R}^{n^2 \times n}$.

5. Computing

$$A \odot (A^{-1} \mathcal{A}_{(1)})^T = A \odot (B \odot C) = [\mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i]_i,$$

solves the tensor decomposition problem.

Let's perform an experiment in Tensorlab v3.0:

1. Create a rank-25 random tensor of size $25 \times 25 \times 25$:

```
>> FactorMatrices{1} = randn(25,25);
>> FactorMatrices{2} = randn(25,25);
>> FactorMatrices{3} = randn(25,25);
% generate the full tensor
>> A = cpdgen(FactorMatrices);
```

Let's perform an experiment in Tensorlab v3.0:

1. Create a rank-25 random tensor of size $25 \times 25 \times 25$:

```
>> FactorMatrices{1} = randn(25,25);
>> FactorMatrices{2} = randn(25,25);
>> FactorMatrices{3} = randn(25,25);
% generate the full tensor
>> A = cpdgen(FactorMatrices);
```

2. Compute $\mathcal{A}$'s decomposition U and compare the outputs relative to the machine precision $\epsilon \approx 2 \cdot 10^{-16}$:

```
>> U = cpd_gevd(A, 25);
>> E = A - cpdgen(U);
>> norm( E(:), 2 ) / eps
ans =
     8.6249e+04
```

Let's perform an experiment in Tensorlab v3.0:

1. Create a rank-25 random tensor of size $25 \times 25 \times 25$:

```
>> FactorMatrices{1} = randn(25,25);
>> FactorMatrices{2} = randn(25,25);
>> FactorMatrices{3} = randn(25,25);
% generate the full tensor
>> A = cpdgen(FactorMatrices);
```

2. Compute $\mathscr{A}$'s decomposition U and compare the outputs relative to the machine precision $\epsilon \approx 2 \cdot 10^{-16}$:

```
>> U = cpd_gevd(A, 25);
>> E = A - cpdgen(U);
>> norm( E(:), 2 ) / eps
ans =
     8.6249e+04
```

What happened?

Let us look more closely at the computational problem:

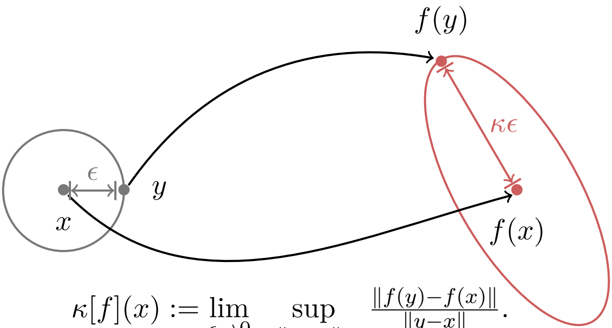
- The input is a tensor $\mathcal{A} \in \sigma_{25} \subset \mathbb{R}^{25\times 25\times 25}$ of rank 25.
- The output is the tuple $(\mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i)_{i=1}^{25} \in \mathcal{S}^{\times 25}$.

Let $f : \sigma_{25} \to \mathcal{S}^{\times 25}$ be the function that maps a tensor to its decomposition. Then, what we observed was

$$\frac{\|f(\mathcal{A}) - f(\mathcal{A}')\|}{\|\mathcal{A} - \mathcal{A}'\|} \approx 8 \cdot 10^4$$

with $\|\mathcal{A} - \mathcal{A}'\| \approx 2 \cdot 10^{-16}$.

The **condition number** quantifies the **worst-case sensitivity** of a (local) function f to perturbations of the input.



$$\kappa[f](x) := \lim_{\epsilon \to 0} \sup_{y:\|x-y\|\leq\epsilon} \frac{\|f(y)-f(x)\|}{\|y-x\|}.$$

Here: $f : \sigma_r \to \mathcal{S}^{\times r}$ is a local inverse of the addition map:

$$\begin{aligned} \Phi_r : \mathcal{S} \times \cdots \times \mathcal{S} &\to \mathbb{R}^{n_1 \times n_2 \times n_3} \\ (\mathcal{A}_1, \ldots, \mathcal{A}_r) &\mapsto \mathcal{A}_1 + \cdots + \mathcal{A}_r \end{aligned}$$

Proposition (Beltrán, Breiding, Vannieuwenhoven)

If σ_r is generically identifiable, there is an open dense submanifold $\mathcal{M}_r \subset \sigma_r$ such that:

1. *For all $\mathcal{A} \in \mathcal{M}_r$ the condition number is the same for all local inverses. We denote it by $\kappa(\mathcal{A})$.*
2. *$\kappa(\mathcal{A}) < \infty$ for all $\mathcal{A} \in \mathcal{M}_r$.*

The interpretation of the condition number is: if $\mathcal{A} = \mathcal{A}_1 + \cdots + \mathcal{A}_r$ and $\mathcal{A}' = \mathcal{A}_1' + \cdots + \mathcal{A}_r'$, then for $\|\mathcal{A} - \mathcal{A}'\|_F \approx 0$ we have the **asymptotically sharp bound**

$$\underbrace{\min_{\pi \in \mathfrak{S}_r} \sqrt{\sum_{i=1}^{r} \|\mathcal{A}_i - \mathcal{A}_{\pi_i}'\|_F^2}}_{\text{forward error}} \lesssim \underbrace{\kappa(\mathcal{A})}_{\text{condition number}} \cdot \underbrace{\|\mathcal{A} - \mathcal{A}'\|_F}_{\text{backward error}}$$

Back to our example

```
>> FactorMatrices{1} = randn(25,25);
>> FactorMatrices{2} = randn(25,25);
>> FactorMatrices{3} = randn(25,25);
>> A = cpdgen(FactorMatrices);
>> U = cpd_gevd(A, 25);
>> E = A - cpdgen(U);
>> norm( E(:), 2 ) / eps
ans =
     8.6249e+04
```

We understand now that this can happen, because of a high condition number. However,

```
>> kappa = condition_number(U)
ans =
     2.134
```

The only explanation is that **there is something wrong with the algorithm**.

We show that algorithms based on a reduction to tensors in $\mathbb{R}^{n_1 \times n_2 \times 2}$ are **numerically unstable**.

The forward error produced by the algorithm divided by the backward error is “much” larger than the condition number, for some inputs.

Pencil-based algorithms

A **pencil-based algorithm** (PBA) is an algorithm that computes the CPD of

$$\mathcal{A} = \sum_{i=1}^{r} \mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i \in \sigma_r \subset \mathbb{R}^{n_1 \times n_2 \times n_3}$$

in the following way:

S1. Choose a fixed $Q \in \mathbb{R}^{n_3 \times 2}$ with orthonormal columns.

S2. $\mathcal{B} \leftarrow (I, I, Q^T) \cdot \mathcal{A}$;

S3. $\{\mathbf{a}_1, \ldots, \mathbf{a}_r\} \leftarrow$ decompose $\mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times 2}$;

S4. Choose an order $A := (\mathbf{a}_1, \ldots, \mathbf{a}_r)$;

S5. $(\mathbf{b}_1 \otimes \mathbf{c}_1, \ldots, \mathbf{b}_r \otimes \mathbf{c}_r) \leftarrow (A^\dagger \mathcal{A}_{(1)})^T$;

S6. $\texttt{output} \leftarrow (\mathbf{a}_1 \otimes \mathbf{b}_1 \otimes \mathbf{c}_1, \ldots, \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r)$.

Pencil-based algorithms

A **pencil-based algorithm** (PBA) is an algorithm that computes the CPD of

$$\mathcal{A} = \sum_{i=1}^{r} \mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i \in \sigma_r \subset \mathbb{R}^{n_1 \times n_2 \times n_3}$$

in the following way:

OK Choose a fixed $Q \in \mathbb{R}^{n_3 \times 2}$ with orthonormal columns.

OK $\mathcal{B} \leftarrow (I, I, Q^T) \cdot \mathcal{A}$;

BAD $\{\mathbf{a}_1, \ldots, \mathbf{a}_r\} \leftarrow$ decompose $\mathcal{B} \in \mathbb{R}^{n_1 \times n_2 \times 2}$;

OK Choose an order $A := (\mathbf{a}_1, \ldots, \mathbf{a}_r)$;

OK $(\mathbf{b}_1 \otimes \mathbf{c}_1, \ldots, \mathbf{b}_r \otimes \mathbf{c}_r) \leftarrow (A^\dagger \mathcal{A}_{(1)})^T$;

OK $\texttt{output} \leftarrow (\mathbf{a}_1 \otimes \mathbf{b}_1 \otimes \mathbf{c}_1, \ldots, \mathbf{a}_r \otimes \mathbf{b}_r \otimes \mathbf{c}_r)$.

The BAD step transforms the numerically “easy” problem

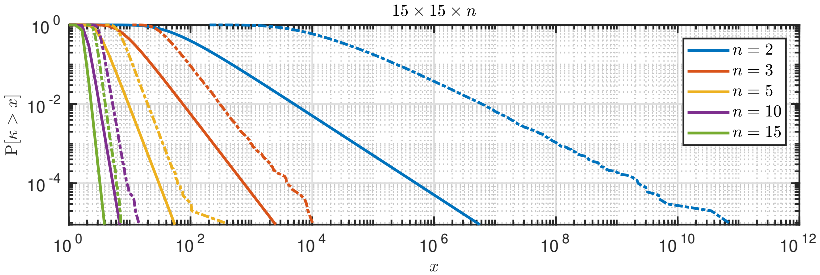
$$\text{compute the CPD of } \mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$$

into the numerically hard problem

$$\text{compute the CPD of } \mathcal{B} = (I, I, Q^T)\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times 2}.$$

The reason for this is that we can have:

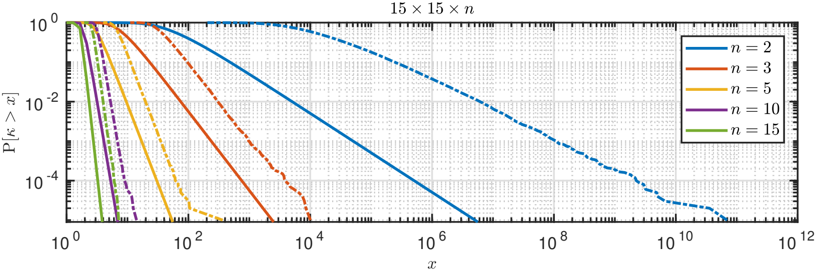
$$\kappa(\mathcal{A}) \approx 1, \quad \text{while} \quad \kappa(\mathcal{B}) \gg 1.$$



Dashed lines = empirical distribution of $\kappa(A)$ for

$$\mathcal{A} = \sum_{i=1}^{15} \mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i \in \mathbb{R}^{15\times 15\times n},$$

where the $\mathbf{a}_i, \mathbf{b}_i, \mathbf{c}_i$ are independent Gaussian vectors.



Dashed lines = empirical distribution of $\kappa(A)$ for

$$\mathcal{A} = \sum_{i=1}^{15} \mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i \in \mathbb{R}^{15\times 15\times n},$$

where the $\mathbf{a}_i, \mathbf{b}_i, \mathbf{c}_i$ are independent Gaussian vectors.

Theorem (Breiding, Vannieuwenhoven (2019))

Let $\mathcal{A} = \sum_{i=1}^{r} \mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i \in \mathbb{R}^{n_1\times n_2\times 2}$ *have Gaussian factors.*
Then: $\mathbb{E}\, \kappa(\mathcal{A}) = \infty$.

Let $\{\widetilde{\mathcal{A}}_1, \ldots, \widetilde{\mathcal{A}}_r\}$ be the CPD of $\mathcal{A}$ (in floating-point representation) returned by the PBA.

We show that for every $\epsilon > 0$ there exists an open neighborhood $\mathcal{O}_\epsilon \subset \sigma_r$ such that the **excess factor**

$$\begin{aligned}\omega(\mathcal{A}) &= \frac{\text{observed forward error due to algorithm}}{\text{maximum forward error due to problem}} \\ &:= \frac{\min_{\pi \in \mathfrak{S}_r} \sqrt{\sum_{i=1}^{r} \|\mathcal{A}_i - \widetilde{\mathcal{A}}_i\|^2}}{\kappa(\mathcal{A}) \cdot \|\mathcal{A} - \mathrm{fl}(\mathcal{A})\|_F}\end{aligned}$$

behaves like a constant times ϵ^{-1}.

For PBAs this ratio is essentially $= \dfrac{\kappa(\mathcal{B})}{\kappa(\mathcal{A})}$.

Formally, we showed the following result:

Theorem (Beltrán, Breiding, Vannieuwenhoven (2019))

There exist a constant $k > 0$ and a tensor

$$\mathcal{O} \in \sigma_r \subset \mathbb{R}^{n_1 \times n_2 \times n_3}$$

with the following properties: for all sufficiently small $\epsilon > 0$, there exists an open neighborhood $\mathcal{O}_\epsilon$ of $\mathcal{O}$, such that for all tensors $\mathcal{A} \in \mathcal{O}_\epsilon$ we have

$$\omega(\mathcal{A}) = \frac{\textit{observed forward error due to algorithm}}{\textit{maximum forward error due to problem}} \geq k\epsilon^{-1}.$$

Formally, we showed the following result:

Theorem (Beltrán, Breiding, Vannieuwenhoven (2019))

There exist a constant $k > 0$ and a tensor

$$\mathcal{O} \in \sigma_r \subset \mathbb{R}^{n_1 \times n_2 \times n_3}$$

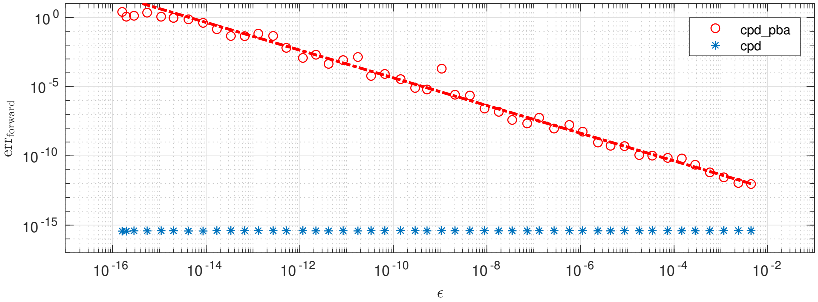
with the following properties: for all sufficiently small $\epsilon > 0$, there exists an open neighborhood $\mathcal{O}_\epsilon$ of $\mathcal{O}$, such that for all tensors $\mathcal{A} \in \mathcal{O}_\epsilon$ we have

$$\omega(\mathcal{A}) = \frac{\textit{observed forward error due to algorithm}}{\textit{maximum forward error due to problem}} \geq k\epsilon^{-1}.$$

The reason for this is

$$\kappa(\mathcal{A}) \approx 1, \quad \text{while} \quad \kappa(\mathcal{B}) \gg 1.$$

Distribution of the forward error



Forward error $\mathrm{err}_{\mathrm{forward}}$ for random tensors in $\mathcal{O}_\epsilon$:

cpd-pba $=$ Pencil-based algorithm

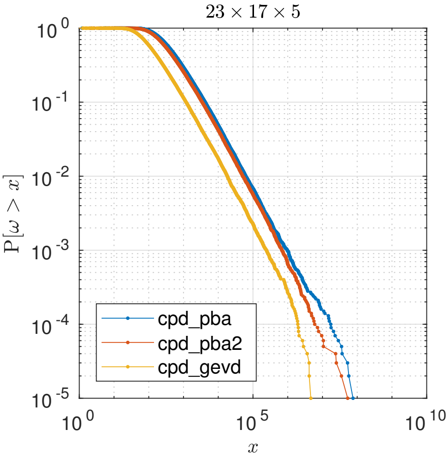
cpd $=$ Pencil-based algorithm $+$ iterative refinement.

In our formal statement ...

1. we show that the excess factor is unbounded in a small neighborhood $\mathcal{O}_\epsilon$;
2. the projection matrix Q is chosen independently from $\mathcal{A}$.

Experiments indicate that a high excess factor is a problem in general.

Empirical distribution of the excess factor



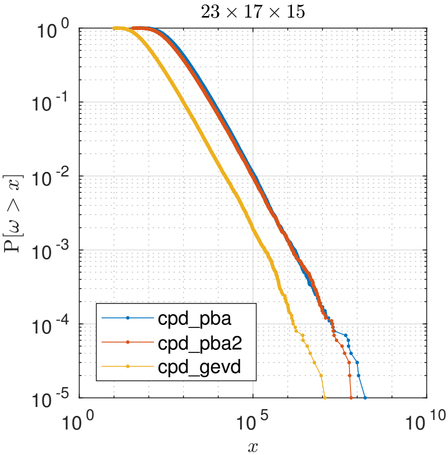
10^5 random $23 \times 17 \times 5$ tensors $\mathcal{A} = \sum_{i=1}^{17} \mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i$ of rank 17 with Gaussian factors.

`cpd-pba`:
use GEVD for $\mathcal{B} = (I, I, Q^T)\mathcal{A}$; random Q.

`cpd-pba2`:
use iterative method for $\mathcal{B}$; random Q.

`cpd-gevd`:
use GEVD for $\mathcal{B}$; choose Q depending on $\mathcal{A}$.

Empirical distribution of the excess factor



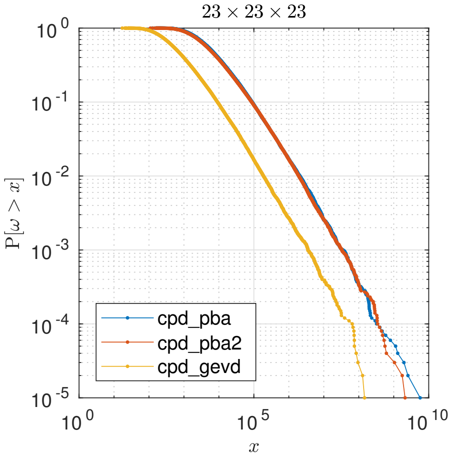
10^5 random $23 \times 17 \times 15$ tensors $\mathcal{A} = \sum_{i=1}^{17} \mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i$ of rank 17 with Gaussian factors.

`cpd-pba`:
use GEVD for $\mathcal{B} = (I, I, Q^T)\mathcal{A}$; random Q.

`cpd-pba2`:
use iterative method for $\mathcal{B}$; random Q.

`cpd-gevd`:
use GEVD for $\mathcal{B}$; choose Q depending on $\mathcal{A}$.

Empirical distribution of the excess factor



10^5 random $23 \times 23 \times 23$ tensors $\mathcal{A} = \sum_{i=1}^{23} \mathbf{a}_i \otimes \mathbf{b}_i \otimes \mathbf{c}_i$ of rank 23 with Gaussian factors.

`cpd-pba`:
use GEVD for $\mathcal{B} = (I, I, Q^T)\mathcal{A}$; random Q.

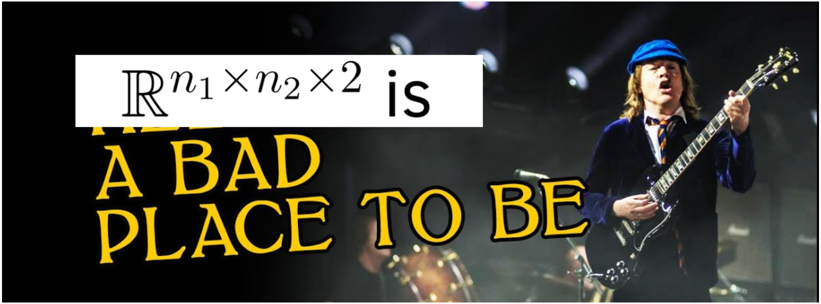
`cpd-pba2`:
use iterative method for $\mathcal{B}$; random Q.

`cpd-gevd`:
use GEVD for $\mathcal{B}$; choose Q depending on $\mathcal{A}$.

Conclusions

Take-away story:

1. Reduction to a matrix pencil yields numerically unstable algorithms for computing CPDs.
2. The reason is that the ratio of condition numbers $\frac{\kappa(\mathcal{B})}{\kappa(\mathcal{A})}$ for $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $\mathcal{B} = (I, I, Q^T)\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times 2}$ is unbounded.



Further reading

- Beltrán, Breiding, and Vannieuwenhoven, *Pencil-based algorithms for tensor rank decomposition are not stable*, SIAM J. Matrix Anal. and Appl., 2019.
- Beltrán, Breiding, and Vannieuwenhoven, *The average condition number of most tensor rank decomposition problems is infinite*, arXiv1903.05527.
- Breiding and Vannieuwenhoven, *The condition number of join decompositions*, SIAM J. Matrix Anal. and Appl., 2018.
- Breiding and Vannieuwenhoven, *On the average condition number of tensor rank decompositions*, IMA J. Num. Anal., 2019.
- Breiding and Vannieuwenhoven, *A Riemannian trust region method for the canonical tensor rank approximation problem*, SIAM J. Optim, 2018.