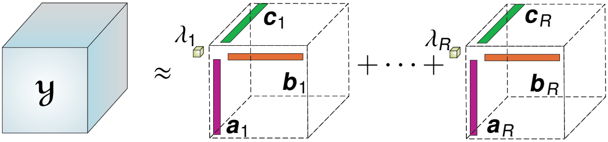
Canonical Polyadic Decomposition

Center for Computational and Data-Intensive Science and Engineering (CDISE)
Skolkowo Institute of Science and Technology (SKOLTECH), Moscow, Russia

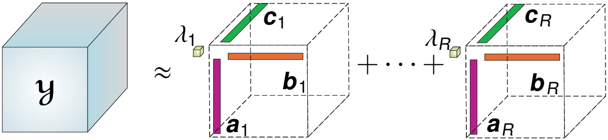
November 21, 2024

CANDECOMP/PARAFAC (CPD)



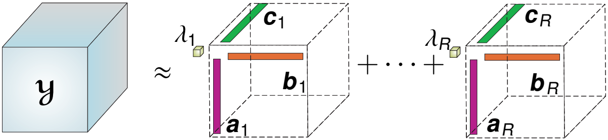
- Application of Tensors in a raw format may be intractable, as the required storage memory and a number of operations grow exponentially with the tensor order.
- To tackle this issue, represent higher-order tensors through multi-way operations over their latent components.
- Canonical Polyadic Decomposition (CPD): an extension of matrix factorization to extract rank-1 tensor patterns from multiway data.

CANDECOMP/PARAFAC (CPD)



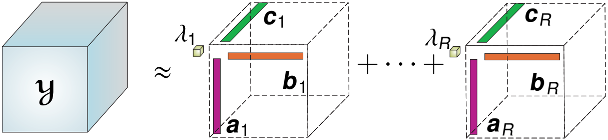
- 1927 first studied Hitchcock (1927)
- 1969 determined the complexity of matrix multiplication Strassen (1969).
- 1972 later known as parallel factor analysis (PARAFAC), a tool for chemometric analysis popularized by Harshman (1972),Carroll and Chang (1970) and KruskalKruskal (1977)
- Since the 1990's, the CPD has attracted attention in signal processing Sidiropoulos et al. (2000); Yeredor (2000); Comon and Rajih (2006).

CANDECOMP/PARAFAC (CPD)



- 2000s New applications of CPD in feature extraction, data reconstruction, image completion and various tracking scenarios Nion and Sidiropoulos (2009); Phan et al. (2015).
- 2014-CPD of kernel tensors in convolutional can accelerate the inference of convolutional neural networks Jaderberg et al. (2014).
- In cyber security, the tensor can represent network traffic volume: time $\times$ senderIP $\times$ receiverIP $\times$ port.

CANDECOMP/PARAFAC (CPD)



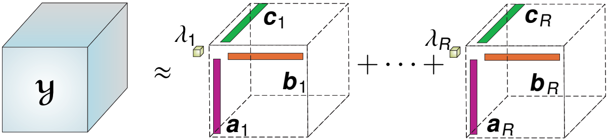
- Model parameters e.g., in multivariate polynomial approximation, Volterra series representation

$$h(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3) = w_0 + \sum_k \boldsymbol{w}_k^T \boldsymbol{x}_k + \sum_{k,l} \boldsymbol{x}_k^T \mathbf{W}_{k,l} \boldsymbol{x}_l + \sum_{k,l,m} \mathcal{W}_{k,l,m} \bar{\times}_1 \boldsymbol{x}_k \bar{\times}_2 \boldsymbol{x}_l \bar{\times}_3 \boldsymbol{x}_m$$

- Multivariate Gaussian distribution is defined by the tensor mean, $\mathcal{M} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, and mode-$n$ covariance, $\mathbf{R}^{(n)} \in \mathbb{R}^{I_n \times I_n}$

$$\boldsymbol{x} \sim \mathcal{N}\left(\boldsymbol{m}, \bigotimes_{n=N}^{1} \mathbf{R}^{(n)}\right)$$

CANDECOMP/PARAFAC (CPD)



- Over its long history, the model has been extended with various constraints
  with many efficient algorithms
- Despite the great successes, there are still many challenging problems in the CP tensor representation/decomposition.
- Computation for large volume and high order tensors, e.g., those of order $N = 10$ or 20
- Degeneracy

Part I

Basic Multilinear Algebra

- **Tensor**: a multi-way array of data $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$
  Matrices by bold capital letters, e.g.
  $\mathbf{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_R] \in \mathbb{R}^{I \times R}$,
  Vectors by bold italic letters, e.g. $\boldsymbol{a}_j$ or $\boldsymbol{I} = [I_1, I_2, \ldots, I_N]$.
- **Slice**: For an order-3 tensor $\mathcal{Y}$, $\mathbf{Y}_k = \mathbf{Y}_{::k}$ denote frontal slice, $\mathbf{Y}_{:j:}$, lateral slice, and $\mathbf{Y}_{i::}$ horizontal slice.
- **Tube**(fiber, vector) at a position (i, j) along the mode-3 is denoted by $\boldsymbol{y}_{ij:}$,
- **Tube** at a position $(i_1, \ldots, i_{n-1}, :, i_{n+1}, \ldots, i_N)$ along mode-n is an I_n vector

$$\mathcal{A}(i_1, \ldots, i_{n-1}, :, i_{n+1}, \ldots, i_N) = \begin{bmatrix} \mathcal{A}(i_1, \ldots, i_{n-1}, 1, i_{n+1}, \ldots, i_N) \\ \vdots \\ \mathcal{A}(i_1, \ldots, i_{n-1}, I_n, i_{n+1}, \ldots, i_N) \end{bmatrix}.$$

Definition (vectorization)

Vectorization of an order-N tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$: maps $\mathcal{A}$ to a column vector $\boldsymbol{a}$

$$\boldsymbol{a} = \mathrm{vec}(\mathcal{A}) = \begin{bmatrix} \mathrm{vec}\left(\mathcal{A}^{(1)}\right) \\ \vdots \\ \mathrm{vec}\left(\mathcal{A}^{(I_N)}\right) \end{bmatrix} \tag{1}$$

where $\mathcal{A}^{(i_N)}$ is an $(N-1)$-order subtensor: $\mathcal{A}^{(i_N)}(i_1, i_2, \ldots, i_{N-1}) = \mathcal{A}(i_1, i_2, \ldots, i_{N-1}, i_N)$.

Definition (**Unfolding or matricization of tensor)**

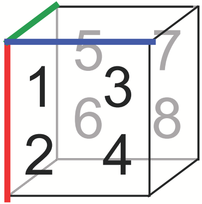
Mode-n unfolding of $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is to horizontally concatenate all tubes of $\mathcal{A}$ along mode-n to yield an $I_n \times (\prod_{m \neq n} I_m)$ matrix

$$\mathbf{A}_{(n)} = \begin{bmatrix} \text{vec}\Big(\mathcal{A}^{(1,n)}\Big) & \cdots & \text{vec}\Big(\mathcal{A}^{(I_n,n)}\Big) \end{bmatrix}^T$$

where $\mathcal{A}^{(i_n,n)}$ is an $(N-1)$-order subtensor of $\mathcal{A}$ whose the n-th index is fixed to i_n

$$\mathcal{A}^{(i_n,n)}(i_1, \ldots, i_{n-1}, i_{n+1}, \ldots, i_N) = \mathcal{A}(i_1, \ldots, i_n, \ldots, i_N).$$

Matricization



$$\mathbf{X}_{(1)} = \begin{bmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{bmatrix}$$

$$\mathbf{X}_{(2)} = \begin{bmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \end{bmatrix}$$

$$\mathbf{X}_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$$

$$\mathrm{vec}(\mathcal{X}) = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \end{bmatrix}$$

Definition (**Outer product)**

The outer product of the tensors $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and $\mathcal{X} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_M}$ is given by

$$\mathcal{Z} = \mathcal{Y} \circ \mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N \times J_1 \times J_2 \times \cdots \times J_M}, \tag{2}$$

where

$$z_{i_1, i_2, \ldots, i_N, j_1, j_2, \ldots, j_M} = y_{i_1, i_2, \ldots, i_N} \, x_{j_1, j_2, \ldots, j_M}. \tag{3}$$

The tensor $\mathcal{Z}$ contains all the possible combinations of pair-wise products between the elements of $\mathcal{Y}$ and $\mathcal{X}$.

- Rank-one matrix: $\mathbf{A} = \boldsymbol{a} \circ \boldsymbol{b} = \boldsymbol{a}\boldsymbol{b}^T \in \mathbb{R}^{I \times J}$
- Rank-one tensor $\mathcal{Z} = \boldsymbol{a} \circ \boldsymbol{b} \circ \boldsymbol{c} \in \mathbb{R}^{I \times J \times Q}$, where $z_{ijq} = a_i \, b_j \, c_q$.

Definition (**Kronecker product)**

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\ \mathbf{B} & a_{12}\ \mathbf{B} & \cdots & a_{1J}\ \mathbf{B} \\ a_{21}\ \mathbf{B} & a_{22}\ \mathbf{B} & \cdots & a_{2J}\ \mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\ \mathbf{B} & a_{I2}\ \mathbf{B} & \cdots & a_{IJ}\ \mathbf{B} \end{bmatrix} \tag{4}$$

Definition (**Khatri-Rao product)**

For $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{T \times J}$, their Khatri-Rao product performs:

$$\begin{aligned} \mathbf{A} \odot \mathbf{B} &= [\boldsymbol{a}_1 \otimes \boldsymbol{b}_1 \;\; \boldsymbol{a}_2 \otimes \boldsymbol{b}_2 \;\; \cdots \;\; \boldsymbol{a}_J \otimes \boldsymbol{b}_J] \\ &= \left[\operatorname{vec}(\boldsymbol{b}_1 \boldsymbol{a}_1^T) \;\; \operatorname{vec}(\boldsymbol{b}_2 \boldsymbol{a}_2^T) \;\; \cdots \;\; \operatorname{vec}(\boldsymbol{b}_J \boldsymbol{a}_J^T)\right] \in \mathbb{R}^{IT \times J}. \end{aligned}$$

Definition (**Hadamard product of two equal-size matrices)**

$$\mathbf{A} \circledast \mathbf{B} = \begin{bmatrix} a_{11}\, b_{11} & a_{12}\, b_{12} & \cdots & a_{1J}\, b_{1J} \\ a_{21}\, b_{21} & a_{22}\, b_{22} & \cdots & a_{2J}\, b_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\, b_{I1} & a_{I2}\, b_{I2} & \cdots & a_{IJ}\, b_{IJ} \end{bmatrix}. \tag{5}$$

Properties: Vectorization I

- $\mathrm{vec}\left(\boldsymbol{a}\boldsymbol{b}^T\right) = \boldsymbol{b} \otimes \boldsymbol{a}$
- $\mathrm{vec}(\boldsymbol{a} \circ \boldsymbol{b} \circ \boldsymbol{c}) = \boldsymbol{c} \otimes \boldsymbol{b} \otimes \boldsymbol{a}$
- $\mathrm{vec}\left(\mathbf{A}\mathbf{B}^T\right) = (\mathbf{B} \odot \mathbf{A})\mathbf{1}_R$
- $\mathrm{vec}\left(\mathbf{A}\,\mathrm{diag}(\boldsymbol{s})\mathbf{B}^T\right) = (\mathbf{B} \odot \mathbf{A})\boldsymbol{s}$
- $\mathrm{vec}\left(\mathbf{A}\mathbf{G}\mathbf{B}^T\right) = (\mathbf{B} \otimes \mathbf{A})\,\mathrm{vec}(\mathbf{G})$
- $\mathrm{vec}(\mathbf{A} \circledast \mathbf{B}) = \mathrm{vec}(\mathbf{A}) \circledast \mathrm{vec}(\mathbf{B})$
- $\mathrm{vec}(\mathbf{A}\mathbf{B} - \mathbf{B}\mathbf{A}) = (\mathbf{I} \otimes \mathbf{A} - \mathbf{A}^T \otimes \mathbf{I})\,\mathrm{vec}(\mathbf{B})$

Properties I

$(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T$
Very important

$$
\begin{aligned}
(\mathbf{A} \otimes \mathbf{B})^T(\mathbf{C} \otimes \mathbf{D}) &= (\mathbf{A}^T\mathbf{C}) \otimes (\mathbf{B}^T\mathbf{D}) \\
(\mathbf{A} \otimes \mathbf{B})(\mathbf{E} \odot \mathbf{F}) &= (\mathbf{A}\mathbf{E}) \odot (\mathbf{B}\mathbf{F}) \\
(\mathbf{A} \odot \mathbf{B})^T \, (\mathbf{A} \odot \mathbf{B}) &= \mathbf{A}^T\mathbf{A} \, \circledast \, \mathbf{B}^T\mathbf{B}, \\
(\mathbf{A} \odot \mathbf{B})^\dagger &= [(\mathbf{A}^T\mathbf{A}) \, \circledast (\mathbf{B}^T\mathbf{B})]^{-1}(\mathbf{A} \odot \mathbf{B})^T
\end{aligned}
$$

- Orthogonal matrices **U** and **V**
  $(\mathbf{U} \otimes \mathbf{V})^T(\mathbf{U} \otimes \mathbf{V}) = (\mathbf{U}^T\mathbf{U}) \otimes (\mathbf{V}^T\mathbf{V}) = \mathbf{I}$
  $(\mathbf{U} \odot \mathbf{V})^T(\mathbf{U} \odot \mathbf{V}) = (\mathbf{U}^T\mathbf{U}) \circledast (\mathbf{V}^T\mathbf{V}) = \mathbf{I}$

If **A** and **B** are orthogonal matrices, then $\mathbf{F} = \mathbf{A} \otimes \mathbf{B}$ is also orthogonal

Tensor-matrix product I

Definition (mode-n tensor-matrix product)

$$\mathcal{Y} = \mathcal{G} \times_n \mathbf{A}\,, \quad \text{or} \quad \mathbf{Y}_{(n)} = \mathbf{A}\,\mathbf{G}_{(n)}.$$

For order-3 tensor $\mathcal{G}$ of size $I \times J \times K$, $\mathbf{A}$ ($L \times I$), $\mathbf{B}$ ($M \times J$), and $\mathbf{C}$ ($N \times K$)

$\mathcal{Y} = \mathcal{G} \times_1 \mathbf{A} \rightarrow \mathbf{Y}(:,:,k) = \mathbf{A}\mathbf{G}(:,:,k)$

$\mathcal{Y} = \mathcal{G} \times_2 \mathbf{B} \rightarrow \mathbf{Y}(:,:,k) = \mathbf{G}(:,:,k)\mathbf{B}^T$

$\mathcal{Y} = \mathcal{G} \times_3 \mathbf{C} \rightarrow \mathbf{Y}(:,j,:) = \mathbf{G}(:,j,:)\mathbf{C}^T$

Given $\mathcal{Y} = \mathcal{G} \times_1 \mathbf{A}$, then

$$\begin{aligned}
\mathbf{Y}_{(1)} &= \mathbf{A}\,\mathbf{G}_{(1)} \\
\mathbf{Y}_{(2)} &= \mathbf{G}_{(2)}(\mathbf{I}_{K\times K} \otimes \mathbf{A}^T) \\
\mathbf{Y}_{(3)} &= \mathbf{G}_{(3)}(\mathbf{I}_{J\times J} \otimes \mathbf{A}^T)
\end{aligned}$$

Tensor-matrix multiplication

$\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, $\mathbf{B} \in \mathbb{R}^{M \times J}$, $\boldsymbol{c} \in \mathbb{R}^{K}$

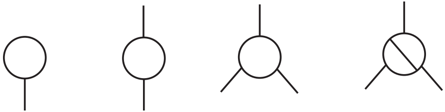
$$\mathcal{Y} = \mathcal{X} \times_2 \mathbf{B} \quad \in \mathbb{R}^{I \times M \times K}$$

$$\begin{aligned} y_{i,m,k} &= \sum_{j=1}^{J} x_{i,j,k}\, b_{m,j} \\ \mathbf{Y}_{:,:,k} &= \mathbf{X}_{:,:,k}\, \mathbf{B}^T \\ \mathbf{Y}_{i,:,:} &= \mathbf{B}\, \mathbf{X}_{i,:,:} \end{aligned}$$

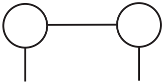
$$\mathcal{Z} = \mathcal{X}\, \bar{\times}_3\, \boldsymbol{c} \quad \in \mathbb{R}^{I \times J}$$

$$\begin{aligned} z_{i,j} &= \sum_{k=1}^{K} x_{i,j,k}\, c_k \\ &= \mathbf{X}_{i,j,:}^T\, \boldsymbol{c} \end{aligned}$$

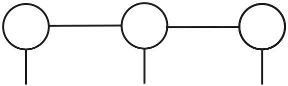
Graph representation of Tensor Network I



(a) Vector, matrix, order-3 tensor and diagonal tensor



(b) Matrix multiplication
$y_{i,j} = \sum_r a_{i,r} b_{r,j}$



(c) TN of three tensors
$y_{i,j,k} = \sum_{r,s} a_{i,r} b_{r,j,s} c_{s,k}$

Low-rank Approximation I

Problem

Find the best representation of a data (matrix) by another matrix of low-rank

$$\mathbf{Y} \approx \mathbf{U}\mathbf{V}^T$$

where $\mathbf{Y}$ *is (often) big, incomplete (sparse).*

Theorem (Eckart-Young-Mirsky)

The low-rank matrix approximation problem

$$\min_{\mathbf{X}} \quad \|\mathbf{Y} - \mathbf{X}\|_F^2 \quad s.t. \quad rank(\mathbf{X}) \leq R$$

has analytical solution $\mathbf{X} = \mathbf{U}\,\mathrm{diag}(\sigma_1, \ldots, \sigma_R)\mathbf{V}^T$*, which is rank-R truncated singular value decomposition of the data matrix* $\mathbf{Y}$*.*

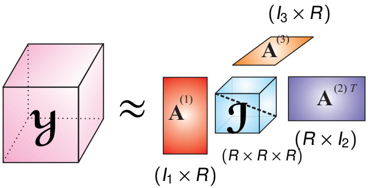
Low-rank Approximation II

- Low-rank decomposition of $\mathbf{X}$ is not unique
  Since for arbitrary non-singular $\mathbf{Q}$

$$\mathbf{A}\mathbf{B}^T = \mathbf{A}\mathbf{Q}\mathbf{Q}^{-1}\mathbf{B}^T = (\mathbf{A}\mathbf{Q})(\mathbf{B}\mathbf{Q}^{-1T})^T$$

- **Uniqueness**: if $\mathbf{A}$ is a mixing matrix, and $\mathbf{B}$ comprises source signals, separation of sources is not possible without incorporating additional information
  - Orthogonality is not sufficient
  - Nonnegativity is not sufficient.
    Nonnegative matrix factorization is not unique.
  - Statistical independence in Independent Component Analysis
  - Sparsity

Canonical Polyadic Decomposition - PARAFAC



- $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is explained by N factor matrices $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R}$

$$
\begin{aligned}
\mathcal{Y} &\approx \sum_{r=1}^{R} \boldsymbol{a}_r^{(1)} \circ \boldsymbol{a}_r^{(2)} \circ \cdots \circ \boldsymbol{a}_r^{(N)} \qquad \|\boldsymbol{a}_r^{(n)}\|_2 = 1, \forall n \neq N \\
&= \mathcal{I} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} = \hat{\mathcal{Y}}
\end{aligned}
$$

Rank and Border Rank I

- **Tensor Rank** the minimal rank-one tensor terms to fully represent the tensor
- **Border Rank**
  If there exists a sequence of tensors of rank at most $r < s$ whose limit is $\mathcal{A}$, the least value of s is the border rank of $\mathcal{A}$.
  The following tensor is of rank-3 but its border rank 2

  $$\mathcal{A} = \boldsymbol{u} \circ \boldsymbol{u} \circ \boldsymbol{v} + \boldsymbol{u} \circ \boldsymbol{v} \circ \boldsymbol{u} + \boldsymbol{v} \circ \boldsymbol{u} \circ \boldsymbol{u}$$

  with $\|\boldsymbol{u}\| = \|\boldsymbol{v}\| = 1$ and $\boldsymbol{u}^T \boldsymbol{v} \neq 1$,
  can be approximated with an arbitrary precision by the following sequence of rank-2 tensors $\mathcal{A}_n$ as $n \to \infty$

Rank and Border Rank II

$$\begin{aligned}\mathcal{A}_n &= n(\boldsymbol{u} + \frac{1}{n}\boldsymbol{v}) \circ (\boldsymbol{u} + \frac{1}{n}\boldsymbol{v}) \circ (\boldsymbol{u} + \frac{1}{n}\boldsymbol{v}) - n\boldsymbol{u} \circ \boldsymbol{u} \circ \boldsymbol{u} \\ &= \boldsymbol{u} \circ \boldsymbol{u} \circ \boldsymbol{v} + \boldsymbol{u} \circ \boldsymbol{v} \circ \boldsymbol{u} + \boldsymbol{v} \circ \boldsymbol{u} \circ \boldsymbol{u} \\ &\quad + \frac{1}{n}(\boldsymbol{u} \circ \boldsymbol{v} \circ \boldsymbol{v} + \boldsymbol{v} \circ \boldsymbol{u} \circ \boldsymbol{v} + \boldsymbol{v} \circ \boldsymbol{v} \circ \boldsymbol{u}) + \frac{1}{n^2}\boldsymbol{v} \circ \boldsymbol{v} \circ \boldsymbol{v}\end{aligned}$$

- **Diverging component - Degeneracy**
  Whenever $\mathcal{A}_n \to \mathcal{A}$ as $n \to \infty$, there should exist at least $1 \leq i \neq j \leq r$ such that

$$\|\mathbf{a}^1_{i,n} \circ \mathbf{a}^2_{i,n} \circ \cdots \circ \mathbf{a}^d_{i,n}\|_F \to \infty$$

  Degeneracy is often encountered when attempting to approximate a tensor using numerical optimization algorithms

Kruskal Rank and Uniqueness of CPD I

Definition (**Kruskal rank)**

A matrix $\mathbf{A}$ has k-rank k_A if and only if every subset of k_A columns of $\mathbf{A}$ is full column rank, and this does not hold true for $k_A + 1$.

- **K-rank and the rank of a matrix**
  A matrix of rank-R, there is at least one subset of R linearly independent columns. In a matrix of k-rank k_A, every subset of k_A columns is of rank k_A.
- **Kruskal's sufficient condition for the essential uniqueness of CPD**
  A d-th order tensor admits an essentially unique CPD if

$$\sum_{k=1}^{d} \text{k-rank}\{\mathbf{A}^{(k)}\} \geq 2R + d - 1$$

Kruskal Rank and Uniqueness of CPD II

- For generic matrices of size $I_n \times R$, the k-rank equals $\min(I_n, R)$. If matrices all have more columns than rows, the sufficient condition can be simplified to

$$\sum_{k=1}^{d} I_n \geq 2R + d - 1$$

Kruskal Rank and Uniqueness of CPD III

- For a CP tensor $\mathcal{X} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$, suppose that $(\mathbf{B} \odot \mathbf{A})$ is not of full column rank. Then there exists a nonzero vector $\boldsymbol{u}$ such that $(\mathbf{B} \odot \mathbf{A})\boldsymbol{u} = 0$.
- From mode-3 unfolding of $\mathcal{X}$, for any vector $\boldsymbol{z}$, we have

  $$\mathbf{X}_{(3)}^T = (\mathbf{B} \odot \mathbf{A})\mathbf{C}^T = (\mathbf{B} \odot \mathbf{A})(\mathbf{C} + \boldsymbol{z}\boldsymbol{u}^T)^T$$

  i.e., adding $\boldsymbol{u}$ to column $\mathbf{C}$ preserves the decomposition,

  $$\mathcal{X} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C} + \boldsymbol{z}\boldsymbol{u}^T]\!]$$

  but produces a different solution for $\mathbf{C}$.
- We can select $\boldsymbol{z}$ such that $(\mathbf{C} + \boldsymbol{z}\boldsymbol{u}^T)$ has zero column. Implying that $\mathcal{X}$ has a CPD with $(R - 1)$ components.
- **Necessary for essential uniqueness:** $(\mathbf{B} \odot \mathbf{A})$ and $(\mathbf{C} \odot \mathbf{A})$ and $(\mathbf{C} \odot \mathbf{B})$ have full column rank

Kruskal Rank and Uniqueness of CPD IV

K-rank of Khatri-Rao product (Stegeman and Sidiropoulos (2007)) Consider matrices $\mathbf{A}(I \times R)$ and $\mathbf{B}(J \times R)$:

- If $k_{\mathbf{A}} = 0$ or $k_{\mathbf{B}} = 0$, then $k_{\mathbf{A} \odot \mathbf{B}} = 0$.
- If $k_{\mathbf{A}} \geq 1$ and $k_{\mathbf{B}} \geq 1$, then $k_{\mathbf{A} \odot \mathbf{B}} \geq \min(k_{\mathbf{A}} + k_{\mathbf{B}} - 1, R)$.

(brief) Proof: Assume that $\mathbf{A}$ and $\mathbf{B}$ have at least one row with all elements nonzero. Since $(\mathbf{A} \odot \mathbf{B})$ has R columns, $k_{(\mathbf{A} \odot \mathbf{B})} \leq R$.

- Assume $k_{(\mathbf{A} \odot \mathbf{B})} < R$. Let n be the smallest number of linearly dependent columns of $(\mathbf{A} \odot \mathbf{B})$, i.e., $k_{(\mathbf{A} \odot \mathbf{B})} = n - 1$
- Collect n linearly dependent columns of $(\mathbf{A} \odot \mathbf{B})$ in the matrix $(\mathbf{A}_n \odot \mathbf{B}_n)$. There exists a nonzero vector, $\boldsymbol{d}$ such that $(\mathbf{A}_n \odot \mathbf{B}_n)\boldsymbol{d} = 0$.
- $\text{vec}\left(\mathbf{B}_n \,\text{diag}(\boldsymbol{d})\mathbf{A}^T\right) = (\mathbf{A}_n \odot \mathbf{B}_n)\boldsymbol{d} = 0$ implies

$$\mathbf{B}_n \,\text{diag}(\boldsymbol{d})\mathbf{A}_n^T = 0$$

Kruskal Rank and Uniqueness of CPD V

- Sylvester's inequality

$$
\begin{aligned}
0 = \operatorname{rank}(\mathbf{B}_n \operatorname{diag}(\boldsymbol{d})\mathbf{A}_n^T) &\geq \operatorname{rank}(\mathbf{A}_n) + \operatorname{rank}(\mathbf{B}_n \operatorname{diag}(\boldsymbol{d})) - n \\
&= \operatorname{rank}(\mathbf{A}_n) + \operatorname{rank}(\mathbf{B}_n) - n
\end{aligned}
$$

  where the last equality is due to the fact that $\operatorname{diag}(\boldsymbol{d})$ is nonsingular.

- Since $n = k_{\mathbf{A}\odot\mathbf{B}} + 1$

$$k_{\mathbf{A}\odot\mathbf{B}} \geq \operatorname{rank}(\mathbf{A}_n) + \operatorname{rank}(\mathbf{B}_n) - 1 \geq k_{\mathbf{A}_n} + k_{\mathbf{B}_n} - 1$$

- Since **B** has at least one row with all elements nonzero, and $(\mathbf{A}_n \odot \mathbf{B}_n)\boldsymbol{d} = 0$, we come to that columns of $\mathbf{A}_n$ are linearly dependent, i.e., $k_{\mathbf{A}_n} \geq k_{\mathbf{A}}$.
- Similarly, $k_{\mathbf{B}_n} \geq k_{\mathbf{B}}$. This gives

$$k_{\mathbf{A}\odot\mathbf{B}} \geq k_{\mathbf{A}} + k_{\mathbf{B}} - 1$$

Kruskal Rank and Uniqueness of CPD VI

If Kruskal's condition $2R + 2 \leq k_{\mathbf{A}} + k_{\mathbf{B}} + k_{\mathbf{C}}$ holds, then

- $k_{\mathbf{A}} \geq 2$, $k_{\mathbf{B}} \geq 2$ and $k_{\mathbf{C}} \geq 2$
- $r_{\mathbf{B} \odot \mathbf{A}} = r_{\mathbf{C} \odot \mathbf{B}} = r_{\mathbf{C} \odot \mathbf{A}} = R$.

Since $k_{\mathbf{C}} \leq R$, it follows that $R \leq k_{\mathbf{A}} + k_{\mathbf{B}} - 2$.

- If $k_{\mathbf{A}} \leq 1$, then $k_{\mathbf{B}} \geq R + 1$, which is impossible.
- Hence, $k_{\mathbf{A}} \geq 2$.

We have

$$r_{\mathbf{B} \odot \mathbf{A}} \geq k_{\mathbf{B} \odot \mathbf{A}} \geq \min(k_{\mathbf{A}} + k_{\mathbf{B}} - 1, R) = R$$

implying that $r_{\mathbf{B} \odot \mathbf{A}} = R$.

Matrix Multiplication Algorithms and CPD I

Matrix multiplication $\mathbf{C} = \mathbf{AB}$

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}, \quad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix}, \quad C = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix},$$

where $\mathbf{A}$ and $\mathbf{B}$ of the sizes 2×2 and 2×2, respectively.

Naive algorithm

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} A_{11} {\color{red}\times} B_{11} + A_{12} {\color{red}\times} B_{21} & A_{11} {\color{red}\times} B_{12} + A_{12} {\color{red}\times} B_{22} \\ A_{21} {\color{red}\times} B_{11} + A_{22} {\color{red}\times} B_{21} & A_{21} {\color{red}\times} B_{12} + A_{22} {\color{red}\times} B_{22} \end{bmatrix}.$$

Matrix Multiplication Algorithms and CPD II

The Strassen algorithm defines

$$
\begin{aligned}
M_1 &= (A_{11} + A_{22})\times(B_{11} + B_{22});\\
M_2 &= (A_{21} + A_{22})\times B_{11};\\
M_3 &= A_{11}\times(B_{12} - B_{22});\\
M_4 &= A_{22}\times(B_{21} - B_{11});\\
M_5 &= (A_{11} + A_{12})\times B_{22};\\
M_6 &= (A_{21} - A_{11})\times(B_{11} + B_{12});\\
M_7 &= (A_{12} - A_{22})\times(B_{21} + B_{22}),
\end{aligned}
$$

using only 7 multiplications

$$
\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} M_1 + M_4 - M_5 + M_7 & M_3 + M_5 \\ M_2 + M_4 & M_1 - M_2 + M_3 + M_6 \end{bmatrix}
$$

Matrix Multiplication Algorithms and CPD III

Consider two matrices **A** and **B** of the sizes $P \times Q$ and $Q \times S$, respectively

Their matrix product $\mathbf{G} = \mathbf{EF}$ of the size $P \times S$

The matrix multiplication can be represented by a tensor $\mathcal{T}$ of the size $PQ \times QS \times PS$ such that

$$\text{vec}(\mathbf{G}) = \mathcal{T} \times_1 \text{vec}\left(\mathbf{E}^T\right)^T \times_2 \text{vec}\left(\mathbf{F}^T\right)^T$$

$$\mathbf{T}_{(1)} = \left[\begin{array}{cccc|cccc|cccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

The tensor $\mathcal{T}$ has a CP representation of rank-7

$$\mathcal{T} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$$

where

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & -1 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} 1 & 1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 & 1 & 0 & 1 \end{bmatrix},$$

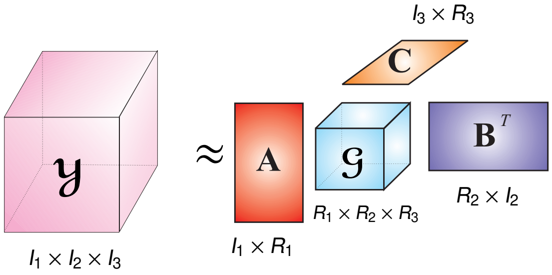
$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 1 & -1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0, \\ 1 & -1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Then

Matrix Multiplication Algorithms and CPD V

$$
\begin{aligned}
\mathrm{vec}(\mathbf{G}) &= \mathfrak{T} \times_1 \mathrm{vec}\left(\mathbf{E}^T\right)^T \times_2 \mathrm{vec}\left(\mathbf{F}^T\right)^T \\
&= [\![\mathrm{vec}\left(\mathbf{E}^T\right)^T \mathbf{A}, \mathrm{vec}\left(\mathbf{F}^T\right)^T \mathbf{B}, \mathbf{C}]\!] \\
&= \mathbf{C}((\mathrm{vec}\left(\mathbf{E}^T\right)^T \mathbf{A}) \circledast (\mathrm{vec}\left(\mathbf{F}^T\right)^T \mathbf{B}))
\end{aligned}
$$

TUCKER Decomposition



$$
\begin{aligned}
\mathcal{Y} &\approx \sum_{j_1=1}^{R_1} \sum_{j_2=1}^{R_2} \cdots \sum_{j_N=1}^{R_N} g_{j_1 j_2 \cdots j_N} \left(\boldsymbol{a}_{j_1}^{(1)} \circ \boldsymbol{a}_{j_2}^{(2)} \circ \cdots \circ \boldsymbol{a}_{j_N}^{(N)} \right) \\
&= \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} = \mathcal{G} \times \{\mathbf{A}\}
\end{aligned}
$$

Part II

Algorithms for CANDECOMP/-PARAFAC

Trilinear and Direct Trilinear Decompositions I

Generalized Rank Annihilation Method (GRAM) Booksh and Kowalski (1994); Booksh et al. (1994); Faber et al. (1994).

- Given $\mathcal{Y}$ a tensor of size $J \times J \times 2$ with only two frontal slices: $\mathbf{Y}_1$ and $\mathbf{Y}_2$.
  CPD of $\mathcal{Y}$ by $\mathbf{A} \in \mathbb{R}^{J \times J}$, $\mathbf{B} \in \mathbb{R}^{J \times J}$ and $\mathbf{C} = [\underline{\boldsymbol{c}}_1^T, \underline{\boldsymbol{c}}_2^T]^T \in \mathbb{R}^{2 \times J}$

$$\mathbf{Y}_1 = \mathbf{A}\,\mathbf{D}_1\,\mathbf{B}^T, \qquad \mathbf{Y}_2 = \mathbf{A}\,\mathbf{D}_2\,\mathbf{B}^T, \tag{6}$$

where $\mathbf{D}_1 = \operatorname{diag}\left\{\underline{\boldsymbol{c}}_1\right\}$, $\mathbf{D}_2 = \operatorname{diag}\left\{\underline{\boldsymbol{c}}_2\right\}$.

Trilinear and Direct Trilinear Decompositions II

- Generalized Eigenvalue problem of two square matrices $\mathbf{Y}_1$ and $\mathbf{Y}_2$

$$\begin{aligned}\mathbf{Y}_1\,(\mathbf{B}^T)^{-1} &= \mathbf{A}\,\mathbf{D}_1\,\mathbf{B}^T\,(\mathbf{B}^T)^{-1} = \mathbf{A}\,\mathbf{D}_2\,\mathbf{D}_2^{-1}\mathbf{D}_1 = \mathbf{A}\,\mathbf{D}_2\mathbf{B}^T(\mathbf{B}^T)^{-1}\mathbf{D} \\ &= \mathbf{Y}_2\,(\mathbf{B}^T)^{-1}\mathbf{D}, \qquad (7)\end{aligned}$$

where $\mathbf{D} = \mathbf{D}_2^{-1}\mathbf{D}_1 = \mathrm{diag}\{\underline{\boldsymbol{c}}_1 \oslash \underline{\boldsymbol{c}}_2\}$ is a diagonal matrix of generalized eigenvalues, and columns of $(\mathbf{B}^T)^{-1}$ are corresponding eigenvectors.

- Due to the scaling ambiguity, we can set the second row vector $\underline{\boldsymbol{c}}_2 = \mathbf{1}^T$, hence, the first row vector $\underline{\boldsymbol{c}}_1$ represents generalized eigenvalues:

$$\underline{\boldsymbol{c}}_1 = \mathrm{diag}\{\mathbf{D}\}^T. \qquad (8)$$

Trilinear and Direct Trilinear Decompositions III

Algorithm 1: GRAM

Input: $\underline{\mathbf{Y}}$: input data of size $J \times J \times 2$, J: number of basis components

Output: $\mathbf{A} \in \mathbb{R}^{J \times J}$, $\mathbf{B} \in \mathbb{R}^{J \times J}$ and $\mathbf{C} \in \mathbb{R}^{2 \times J}$ such that $\underline{\mathbf{Y}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$

begin

$[\Phi, \mathbf{D}] = \mathtt{eig}(\mathbf{Y}_1, \mathbf{Y}_2)$;

$\mathbf{B} = (\Phi^T)^{-1}$;

$\mathbf{A} = \mathbf{Y}_2 \Phi$;

$\mathbf{C} = [\mathrm{diag}\{\mathbf{D}\}, \mathbf{1}]^T$;

end

Trilinear and Direct Trilinear Decompositions IV

Direct Tri-Linear Decomposition (DTLD) (Sanchez and Kowalski)

Nonsquare matrices, $\mathbf{Y}_1$ and $\mathbf{Y}_2$, can be transformed to matrices $\mathbf{T}_q$ of size $J \times J$ by projecting on the two orthonormal matrices $\mathbf{U} \in \mathbb{R}^{I \times J}$ and $\mathbf{V} \in \mathbb{R}^{T \times J}$, usually via a singular value decomposition

$$\begin{aligned} [\mathbf{U}, \mathbf{S}, \mathbf{V}] &= \mathtt{svd}(\mathbf{Y}_1 + \mathbf{Y}_2, J) \\ \mathbf{T}_q &= \mathbf{U}^T \mathbf{Y}_q \mathbf{V}, \qquad (q = 1, 2). \end{aligned}$$

Applying the GRAM algorithm to the tensor $\mathcal{T}$, factors **A**, **B** and **C** can be found as

$$\begin{aligned} [\Phi, \mathbf{D}] &= \mathtt{eig}(\mathbf{T}_1, \mathbf{T}_2) && (9) \\ \mathbf{A} &= \mathbf{U}\,\mathbf{T}_2 \Phi && (10) \\ \mathbf{B} &= \mathbf{V}(\Phi^T)^{-1} && (11) \\ \mathbf{C} &= [\mathrm{diag}\{\mathbf{D}\}, \mathbf{1}]^T. && (12) \end{aligned}$$

Trilinear and Direct Trilinear Decompositions V

The GRAM method is restricted to the tensor having only two slices.
For tensor with more than 2 slides, $\underline{\mathbf{Y}} \in \mathbb{R}^{I \times T \times Q}$ using the Tucker model with a core tensor $\underline{\mathbf{G}} \in \mathbb{R}^{J \times J \times 2}$

$$\underline{\mathbf{Y}} = [\![\underline{\mathbf{G}}; \mathbf{A}_t, \mathbf{B}_t, \mathbf{C}_t]\!], \tag{13}$$

where $\mathbf{A}_t \in \mathbb{R}^{I \times J}$, $\mathbf{B}_t \in \mathbb{R}^{T \times J}$, and $\mathbf{C}_t \in \mathbb{R}^{Q \times 2}$ are factors obtained by using the ALS Tucker algorithm.
The core tensor $\underline{\mathbf{G}}$ is then factorized using the GRAM algorithm

$$\underline{\mathbf{G}} = [\![\mathbf{A}_f, \mathbf{B}_f, \mathbf{C}_f]\!]. \tag{14}$$

The final model is given as $\underline{\mathbf{Y}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ where

$$\mathbf{A} = \mathbf{A}_t \, \mathbf{A}_f \tag{15}$$
$$\mathbf{B} = \mathbf{B}_t \, \mathbf{B}_f \tag{16}$$
$$\mathbf{C} = \mathbf{C}_t \, \mathbf{C}_f. \tag{17}$$

Trilinear and Direct Trilinear Decompositions VI

Algorithm 2: DTLD

Input: $\underline{\mathbf{Y}}$: input data of size $I \times T \times Q$, J: number of basis components

Output: $\mathbf{A} \in \mathbb{R}^{I \times J}$, $\mathbf{B} \in \mathbb{R}^{T \times J}$ and $\mathbf{C} \in \mathbb{R}^{Q \times J}$ such that $\underline{\mathbf{Y}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$

begin

$\underline{\mathbf{Y}} = [\![\underline{\mathbf{G}}; \mathbf{A}_t, \mathbf{B}_t, \mathbf{C}_t]\!]$ /* Tucker ALS */;

$[\Phi, \mathbf{D}] = \mathtt{eig}(\mathbf{G}_1, \mathbf{G}_2)$;

$\mathbf{A} = \mathbf{A}_t \, \mathbf{G}_2 \, \Phi$;

$\mathbf{B} = \mathbf{B}_t \, (\Phi^T)^{-1}$;

$\mathbf{C} = \mathbf{C}_t \, [\mathrm{diag}\{\mathbf{D}\}, \mathbf{1}]^T$ /* or $\mathbf{C} = \mathbf{Y}_{(3)} \, (\mathbf{B} \odot \mathbf{A})$ */;

end

Trilinear and Direct Trilinear Decompositions VII

DTLD

- Efficient initialization for inexact case or contrained CPD
- Work only for order-3 tensor
- rank-R must not exceed tensor dimensions

Generalized rank annihilation method for Higher Order CPD

Is there a closed form solution for CPD of tensors of higher order?

How???

The higher order tensor is first unfolded to be an order-3 tensor of size $I \times J \times K$ so that the two largest dimensions are greater than or equal to rank R, then factorized using DTLD.

Generalized rank annihilation method for Higher Order CPD

Is there a closed form solution for CPD of tensors of higher order?

How???

The higher order tensor is first unfolded to be an order-3 tensor of size $I \times J \times K$ so that the two largest dimensions are greater than or equal to rank R, then factorized using DTLD.

Generalized DTLD method for Higher Order CPD

Example

Given a rank-3 tensor, $\mathcal{Y}$, of size $4 \times 4 \times 5 \times 6$. Find its CPD $\mathcal{Y} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}]\!]$ using DTLD.

- Reshaping $\mathcal{Y}$ to an order-3 tensor, $\mathcal{Y}_{(3,4)}$, of size $4 \times 4 \times 30$ which is still of rank-3.
- Apply DTLD to $\mathcal{Y}_{(3,4)}$ and obtain a rank-3 CPD $\mathcal{Y}_{(3,4)} = [\![\mathbf{A}, \mathbf{B}, \mathbf{F}]\!]$
- Since $\mathbf{F} = \mathbf{D} \odot \mathbf{C}$ has Khatri-Rao product, $\boldsymbol{c}_r \boldsymbol{d}_r^T$ is the best rank-1 of the folding matrix, $\mathbf{F}_r$ of size 5×6 from $\boldsymbol{f}_r = \operatorname{vec}(\mathbf{F}_r)$

$$\mathbf{F}_r \approx \boldsymbol{c}_r \boldsymbol{d}_r^T$$

  using truncated SVD

Generalized DTLD method for Higher Order CPD

Example

Given a rank-3 tensor, $\mathcal{Y}$, of size $4 \times 4 \times 5 \times 6 \times 7$. Find its CPD $\mathcal{Y} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}]\!]$ using DTLD.

- Reshaping $\mathcal{Y}$ to an order-3 tensor, $\mathcal{Y}_{(2,3),(4,5)}$, of size $4 \times 20 \times 42$ which is still of rank-3.
- Apply DTLD to $\mathcal{Y}_{(3,4)}$ and obtain a rank-3 CPD $\mathcal{Y}_{(2,3),(4,5)} = [\![\mathbf{A}, \mathbf{U}, \mathbf{V}]\!]$
- It can be shown that $\mathbf{U} = \mathbf{B} \odot \mathbf{C}$, $\mathbf{V} = \mathbf{E} \odot \mathbf{D}$, hence $\boldsymbol{b}_r \boldsymbol{c}_r^T = \texttt{reshape}(\boldsymbol{u}_\mathrm{r}, [45])$ is the best rank-1 of the matrix of size 4×5 folded from $\boldsymbol{u}_r$
  $\boldsymbol{d}_r \boldsymbol{e}_r^T = \texttt{reshape}(\boldsymbol{v}_\mathrm{r}, [67])$ is the best rank-1 of the matrix of size 6×7 folded from $\boldsymbol{v}_r$

Alternating Least Squares Algorithms I

Minimize Frobenius norm of the approximation error

$$D(\mathcal{Y}\|\hat{\mathcal{Y}}) = \frac{1}{2}\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2$$

where $\hat{\mathcal{Y}} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$

Update rule for A

In order to update **A**, we keep **B** and **C** fixed and rewrite the Frobenius norm for the mode-1 unfolding of the error tensor

$$D(\mathcal{Y}\|\hat{\mathcal{Y}}) = \frac{1}{2}\|\mathbf{Y}_{(1)} - \mathbf{A}(\mathbf{C} \odot \mathbf{B})^T\|_F^2.$$

Compute the gradient and set it to zero

$$\mathbf{A}(\mathbf{C} \odot \mathbf{B})^T(\mathbf{C} \odot \mathbf{B}) - \mathbf{Y}_{(1)}(\mathbf{C} \odot \mathbf{B}) = 0 \qquad (18)$$

we obtain the update for **A**

$$\mathbf{A} = \mathbf{Y}_{(1)}(\mathbf{C} \odot \mathbf{B})((\mathbf{C}^T\mathbf{C}) \circledast (\mathbf{B}^T\mathbf{B}))^{-1} \tag{19}$$

The multiplication $\mathbf{G} = \mathbf{Y}_{(1)}(\mathbf{C} \otimes \mathbf{B})$ is computed as R projections of the tensor $\mathcal{Y}$ by vectors $\boldsymbol{b}_r$ and $\boldsymbol{c}_r$

$$\boldsymbol{g}_r = \mathcal{Y} \times_2 \boldsymbol{b}_r \times_3 \boldsymbol{c}_r \tag{20}$$

Update for B

$$\mathbf{B} = \mathbf{Y}_{(2)}(\mathbf{C} \odot \mathbf{A})((\mathbf{C}^T\mathbf{C}) \circledast (\mathbf{A}^T\mathbf{A}))^{-1} \tag{21}$$

Update for C

$$\mathbf{C} = \mathbf{Y}_{(3)}(\mathbf{B} \odot \mathbf{A})((\mathbf{B}^T\mathbf{B}) \circledast (\mathbf{A}^T\mathbf{A}))^{-1} \tag{22}$$

ALS Algorithms for Higher order Tensors

Cost function

$$D(\mathcal{Y}\|\hat{\mathcal{Y}}) = \frac{1}{2}\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 = \frac{1}{2}\|\mathbf{Y}_{(n)} - \mathbf{A}^{(n)}\{\mathbf{A}\}^{\odot_{-n}^{T}}\|_F^2.$$

1. Initialize all $\mathbf{A}^{(n)}$ randomly or by using SVD Berry et al. (2007), or by selected fibers from the data $\mathcal{Y}$.
2. Estimate $\mathbf{A}^{(n)}$ from the approximation by solving

$$\min_{\mathbf{A}^{(n)}} D_F(\mathcal{Y}\|\hat{\mathcal{Y}}) = \frac{1}{2}\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2, \text{ with fixed } \mathbf{A}^{(m)},\ m \neq n.$$

3. For nonnegative components, set all negative elements of $\mathbf{A}^{(n)}$ to zero or a small positive value ε.
4. Estimate other factors until convergence.

ALS Algorithms for Higher order Tensors

Cost function

$$D(\mathcal{Y}\|\hat{\mathcal{Y}}) = \frac{1}{2}\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 = \frac{1}{2}\|\mathbf{Y}_{(n)} - \mathbf{A}^{(n)}\{\mathbf{A}\}^{\odot_{-n}^T}\|_F^2.$$

1. Initialize all $\mathbf{A}^{(n)}$ randomly or by using SVD Berry et al. (2007), or by selected fibers from the data $\mathcal{Y}$.
2. Estimate $\mathbf{A}^{(n)}$ from the approximation by solving

$$\min_{\mathbf{A}^{(n)}} D_F(\mathcal{Y}\|\hat{\mathcal{Y}}) = \frac{1}{2}\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2, \text{with fixed } \mathbf{A}^{(m)}, m \neq n.$$

3. For nonnegative components, set all negative elements of $\mathbf{A}^{(n)}$ to zero or a small positive value ε.
4. Estimate other factors until convergence.

ALS Algorithms for Higher order Tensors

Cost function

$$D(\mathcal{Y}\|\hat{\mathcal{Y}}) = \frac{1}{2}\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 = \frac{1}{2}\|\mathbf{Y}_{(n)} - \mathbf{A}^{(n)}\{\mathbf{A}\}^{\odot_{-n}^{T}}\|_F^2.$$

1. Initialize all $\mathbf{A}^{(n)}$ randomly or by using SVD Berry et al. (2007), or by selected fibers from the data $\mathcal{Y}$.
2. Estimate $\mathbf{A}^{(n)}$ from the approximation by solving

$$\min_{\mathbf{A}^{(n)}} D_F(\mathcal{Y}\|\hat{\mathcal{Y}}) = \frac{1}{2}\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2, \text{with fixed } \mathbf{A}^{(m)}, m \neq n.$$

3. For nonnegative components, set all negative elements of $\mathbf{A}^{(n)}$ to zero or a small positive value ε.
4. Estimate other factors until convergence.

ALS Algorithms for Higher order Tensors

Cost function

$$D(\mathcal{Y}\|\hat{\mathcal{Y}}) = \frac{1}{2}\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 = \frac{1}{2}\|\mathbf{Y}_{(n)} - \mathbf{A}^{(n)}\{\mathbf{A}\}^{\odot^T_{-n}}\|_F^2.$$

1. Initialize all $\mathbf{A}^{(n)}$ randomly or by using SVD Berry et al. (2007), or by selected fibers from the data $\mathcal{Y}$.
2. Estimate $\mathbf{A}^{(n)}$ from the approximation by solving

$$\min_{\mathbf{A}^{(n)}} D_F(\mathcal{Y}\|\hat{\mathcal{Y}}) = \frac{1}{2}\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2, \text{with fixed } \mathbf{A}^{(m)}, m \neq n.$$

3. For nonnegative components, set all negative elements of $\mathbf{A}^{(n)}$ to zero or a small positive value ε.
4. Estimate other factors until convergence.

Gradient of the cost function w.r.t $\mathbf{A}^{(n)}$

$$\mathbf{G}^{(n)} = \mathbf{Y}_{(n)} \left(\bigodot_{k \neq n} \mathbf{A}^{(k)} \right) - \mathbf{A}^{(n)} \left(\circledast_{k \neq n} \mathbf{A}^{(k)T} \mathbf{A}^{(k)} \right) \in \mathbb{R}^{I_n \times R},$$

ALS Algorithm for CP

$$\mathbf{A}^{(n)} \leftarrow \mathbf{Y}_{(n)} \left(\bigodot_{k \neq n} \mathbf{A}^{(k)} \right) \left(\left\{ \mathbf{A}^T \mathbf{A} \right\}^{\circledast -n} \right)^{\dagger}$$

$\mathbf{A}^{\dagger}$ is the Moore-Penrose inverse of $\mathbf{A}$,

Pros: simple, work well for general case.

Cons: high cost due to computing $\{\mathbf{A}\}^{\odot -n}$, $\mathbf{Y}_{(n)} \{\mathbf{A}\}^{\odot -n}$
not guaranteed to converge to a global minimum or even a stationary point, but only to a solution where the cost functions cease to decrease Kolda and Bader (2009); Berry et al. (2007)
relatively slow for nearly collinear data Langville et al. (2006)

Gradient of the cost function w.r.t $\mathbf{A}^{(n)}$

$$\mathbf{G}^{(n)} = \mathbf{Y}_{(n)} \left(\bigodot_{k \neq n} \mathbf{A}^{(k)} \right) - \mathbf{A}^{(n)} \left(\circledast_{k \neq n} \mathbf{A}^{(k)T} \mathbf{A}^{(k)} \right) \in \mathbb{R}^{I_n \times R},$$

ALS Algorithm for CP

$$\mathbf{A}^{(n)} \leftarrow \mathbf{Y}_{(n)} \left(\bigodot_{k \neq n} \mathbf{A}^{(k)} \right) \left(\left\{ \mathbf{A}^T \mathbf{A} \right\}^{\circledast_{-n}} \right)^{\dagger}$$

$\mathbf{A}^{\dagger}$ is the Moore-Penrose inverse of $\mathbf{A}$,

Pros: simple, work well for general case.

Cons: high cost due to computing $\{\mathbf{A}\}^{\odot_{-n}}$, $\mathbf{Y}_{(n)} \{\mathbf{A}\}^{\odot_{-n}}$
not guaranteed to converge to a global minimum or even a stationary point, but only to a solution where the cost functions cease to decrease Kolda and Bader (2009); Berry et al. (2007).
relatively slow for nearly collinear data Langville et al. (2006).

Gradient of the cost function w.r.t $\mathbf{A}^{(n)}$

$$\mathbf{G}^{(n)} = \mathbf{Y}_{(n)} \left(\bigodot_{k \neq n} \mathbf{A}^{(k)} \right) - \mathbf{A}^{(n)} \left(\circledast_{k \neq n} \mathbf{A}^{(k)T} \mathbf{A}^{(k)} \right) \in \mathbb{R}^{I_n \times R},$$

ALS Algorithm for CP

$$\mathbf{A}^{(n)} \leftarrow \mathbf{Y}_{(n)} \left(\bigodot_{k \neq n} \mathbf{A}^{(k)} \right) \left(\left\{ \mathbf{A}^T \mathbf{A} \right\}^{\circledast_{-n}} \right)^{\dagger}$$

$\mathbf{A}^{\dagger}$ is the Moore-Penrose inverse of $\mathbf{A}$,

Pros: simple, work well for general case.

Cons: high cost due to computing $\{\mathbf{A}\}^{\odot_{-n}}$, $\mathbf{Y}_{(n)}\ \{\mathbf{A}\}^{\odot_{-n}}$
not guaranteed to converge to a global minimum or even a stationary point, but only to a solution where the cost functions cease to decrease Kolda and Bader (2009); Berry et al. (2007).
relatively slow for nearly collinear data Langville et al. (2006).

Gradient of the cost function w.r.t $\mathbf{A}^{(n)}$

$$\mathbf{G}^{(n)} = \mathbf{Y}_{(n)} \left(\bigodot_{k \neq n} \mathbf{A}^{(k)} \right) - \mathbf{A}^{(n)} \left(\circledast_{k \neq n} \mathbf{A}^{(k)T} \mathbf{A}^{(k)} \right) \in \mathbb{R}^{I_n \times R},$$

ALS Algorithm for CP

$$\mathbf{A}^{(n)} \leftarrow \mathbf{Y}_{(n)} \left(\bigodot_{k \neq n} \mathbf{A}^{(k)} \right) \left(\left\{ \mathbf{A}^T \mathbf{A} \right\}^{\circledast_{-n}} \right)^{\dagger}$$

$\mathbf{A}^{\dagger}$ is the Moore-Penrose inverse of $\mathbf{A}$,

Pros: simple, work well for general case.

Cons: high cost due to computing $\{\mathbf{A}\}^{\odot_{-n}}$, $\mathbf{Y}_{(n)}\ \{\mathbf{A}\}^{\odot_{-n}}$
not guaranteed to converge to a global minimum or even a stationary point, but only to a solution where the cost functions cease to decrease Kolda and Bader (2009); Berry et al. (2007).
relatively slow for nearly collinear data Langville et al. (2006).

Line search I

Bro's Line search (LS)

- Predict the factors $\mathbf{A}^{(n)}$ from their previous estimates

$$\mathbf{A}_t^{(n)} = \mathbf{A}_{t-1}^{(n)} + \eta \left(\mathbf{A}_{t-1}^{(n)} - \mathbf{A}_{t-2}^{(n)} \right), \quad n = 1, 2, \ldots, N, \qquad (23)$$

  where t denotes the iteration index, and η is stepsize.
- A heuristic stepsize η is iteratively searched to improve the fit

$$\eta \leftarrow \alpha \, \eta, \qquad (24)$$

  where α is a suitably chosen scalar constant.
- The LS method can find a suitable step value but not an optimal step.

Line Search for order-3 CPD

$$
\begin{aligned}
\hat{\mathcal{Y}} &= [\![\mathbf{A}_{t+1}, \mathbf{B}_{t+1}, \mathbf{C}_{t+1}]\!] \\
&= [\![\mathbf{A}_t + \eta\Delta\mathbf{A}_t, \mathbf{B}_t + \eta\Delta\mathbf{B}_t, \mathbf{C}_t + \eta\Delta\mathbf{C}_t]\!] \\
&= [\![\mathbf{A}_t, \mathbf{B}_t, \mathbf{C}_t]\!] \\
&\quad + \eta([\![\Delta\mathbf{A}_t, \mathbf{B}_t, \mathbf{C}_t]\!] + [\![\mathbf{A}_t, \Delta\mathbf{B}_t, \mathbf{C}_t]\!] + [\![\mathbf{A}_t, \mathbf{B}_t, \Delta\mathbf{C}_t]\!]) \\
&\quad + \eta^2([\![\Delta\mathbf{A}_t, \Delta\mathbf{B}_t, \mathbf{C}_t]\!] + [\![\Delta\mathbf{A}_t, \mathbf{B}_t, \Delta\mathbf{C}_t]\!] + [\![\mathbf{A}_t, \Delta\mathbf{B}_t, \Delta\mathbf{C}_t]\!]) \\
&\quad + \eta^3[\![\Delta\mathbf{A}_t, \Delta\mathbf{B}_t, \Delta\mathbf{C}_t]\!] \\
&= [\![\mathbf{A}_t, \mathbf{B}_t, \mathbf{C}_t]\!] \\
&\quad + \eta[\![[\Delta\mathbf{A}_t, \mathbf{A}_t, \mathbf{A}_t], [\mathbf{B}_t, \Delta\mathbf{B}_t, \mathbf{B}_t], [\mathbf{C}_t, \mathbf{C}_t, \Delta\mathbf{C}_t]]\!] \\
&\quad + \eta^2[\![[\Delta\mathbf{A}_t, \Delta\mathbf{A}_t, \mathbf{A}_t], [\Delta\mathbf{B}_t, \mathbf{B}_t, \Delta\mathbf{B}_t], [\mathbf{C}_t, \Delta\mathbf{C}_t, \Delta\mathbf{C}_t]]\!] \\
&\quad + \eta^3[\![\Delta\mathbf{A}_t, \Delta\mathbf{B}_t, \Delta\mathbf{C}_t]\!]
\end{aligned}
$$

Exact line search

Enhanced or Exact Line Search (ELS) Rajih et al. (2008) finds optimal step values at every even iteration by minimizing the modified cost function

$$D(\mathcal{Y}\|\hat{\mathcal{Y}}) = \frac{1}{2}\left\|\mathcal{Y} - [\![\mathbf{A}_t^{(1)} + \eta_1\Delta\mathbf{A}_t^{(1)}, \mathbf{A}_t^{(2)} + \eta_2\Delta\mathbf{A}_t^{(2)}, \mathbf{A}_t^{(3)} + \eta_2\Delta\mathbf{A}_t^{(3)}]\!]\right\|_F^2,$$

where η_1, η_2, η_3 are stepsizes, and $\Delta\mathbf{A}_t^{(n)} = \mathbf{A}_{t-1}^{(n)} - \mathbf{A}_{t-2}^{(n)}$.
When $\eta_1 = \eta_2 = \eta_3 = \eta$, ELS find roots of a polynomial $\boldsymbol{p}(\mu)$ of degree 5. An optimal stepsize is the root of that polynomial $\boldsymbol{p}(\mu)$ which yields the smallest cost function value.

- ELS alleviates bottlenecks more efficiently than LS.
- But less efficient if the directions provided by the algorithm are bad Comon et al. (2009a).
- High computational cost for higher order tensors, due to building up $(2N-1)$- order polynomials $\boldsymbol{p}(\eta)$.

Line search: Example

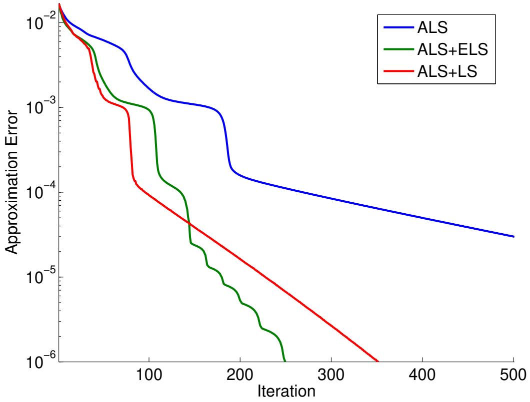


Figure: Approximation errors of the ALS with/without line searches for factorizations of tensors of size $30 \times 30 \times 30$ and rank $R = 10$ composed from highly correlated components which have $\boldsymbol{a}_r^{(n)T} \boldsymbol{a}_s^{(n)} \in [0.8, 0.999]$, $r \neq s$.

CPD with Orthogonality Constraints

CP decomposition of tensor $\mathcal{Y}$

$$\begin{aligned} \min \quad & \|\mathcal{Y} - [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]\|_F^2 \\ \text{s.t} \quad & \mathbf{A}^T\mathbf{A} = \mathbf{I}_R \end{aligned}$$

where the matrix $\mathbf{A}$ is orthogonal

Update rule for A

Update rule for B and C

Part III

Sensitivity

Convolutional Neural Networks I

Model	Params	SIZE(MB)
SimpleNet	6.4M	24.4
SqueezeNet	1.25M	4.7
Inception v4*	42.71M	163
Inception v3*	23.83M	91
Incep-Resv2*	55.97M	214
ResNet-152	60.19M	230
ResNet-50	25.56M	97.70
AlexNet	60.97M	217.00
GoogleNet	7M	40
VGG16	138.36M	512.2

- Deep neural networks show excellent performance in solving complex AI tasks in wide areas including computer vision, speech recognition and natural language processing,
- Most state-of-the-art NNs are Overparameterized, contain a huge number of parameters and thus computationally demanding

Convolutional Neural Networks II

- Common approaches to redundancy reduction in NNs include
  sparsification (prunning),
  low-rank approximation to the weights within the convolutional and fully connected layers,
  prunning layers,
  scaling
  building up a deep network from shallow networks with share parameters

Factorized Layers in Convolutional Neural Networks I

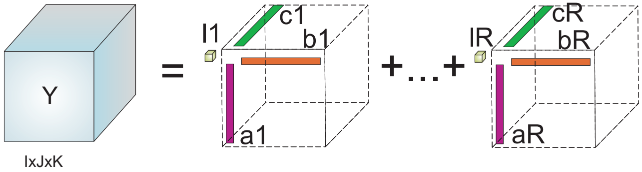


Figure: CANDECOMP/PARAFAC (CPD) as a tool to extract rank-1 patterns from multiway data or compression of convolutional tensors.

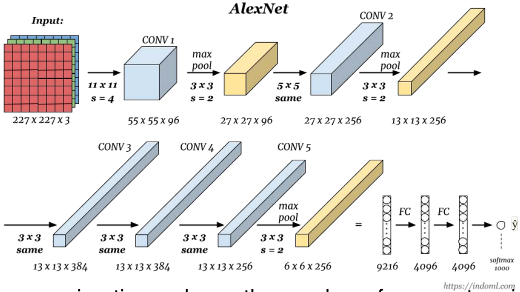
Low-rank tensor factorization

Factorization of weights in convolutional and fully connected layers can replace the dense and large kernels by network of small kernels.

This method reduces the number of parameters in CNNs.

Canonical polyadic tensor decomposition is one of TDs.

Factorized Layers in Convolutional Neural Networks II

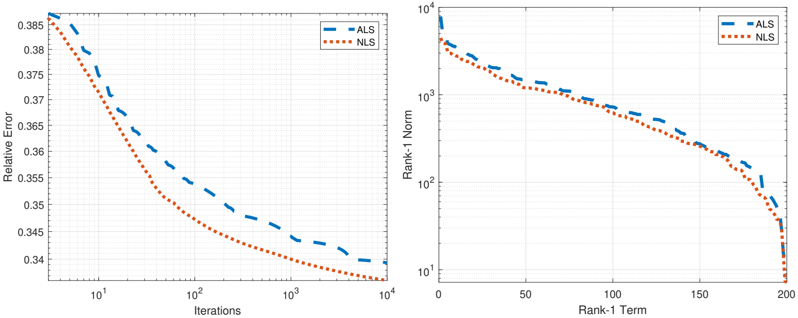


- Low-rank approximation reduces the number of parameters in convolutional layers. Thereby accelerate the inference of the network (Jaderberg et. al, 2014)
- However Lebedev et. al, 2015 observed that the fine-tuning of the whole network was *unstable*.
- Tensor decomposition will not be much efficient if NNs with factorized layers is unstable and sensitive to the change of the parameters.

Indeed it is not straighforward to apply TDs to compress CNNs.
What is the main reason?

Degeneracy

Degeneracy and Sensitivity in CPD of Alexnet I



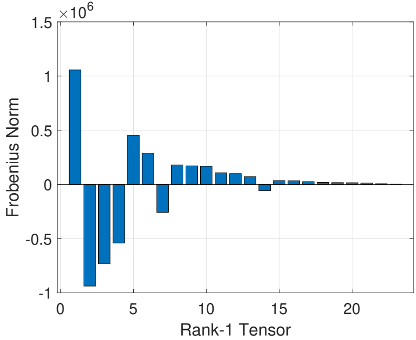
ALS: alternating least squares algorithm

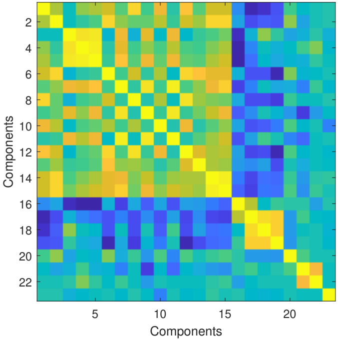
NLS: nonlinear least squares algorithm (TensorLab)

Algorithms converge slowly to false local minima after 10000 iterations

while norms of the estimated rank-1 tensors are very high.

Example 2: Multiplication of two matrices of size 3×3 I





$$\mathrm{vec}(\mathbf{AB}) = \mathcal{Y} \times_1 \mathrm{vec}\left(\mathbf{A}^T\right)^T \times_2 \mathrm{vec}\left(\mathbf{B}^T\right)^T$$

where $\mathcal{Y}$ is a tensor of size $9 \times 9 \times 9$ comprises only **1**s and **0**s.

- CPD of $\mathcal{Y}$ with rank-$R = 23$ often gets stuck in false local minima.
- Loading components are highly collinear,
- Some rank-1 tensors have very large norm

Degeneracy and Sensitivity in CPD of Resnet18 I

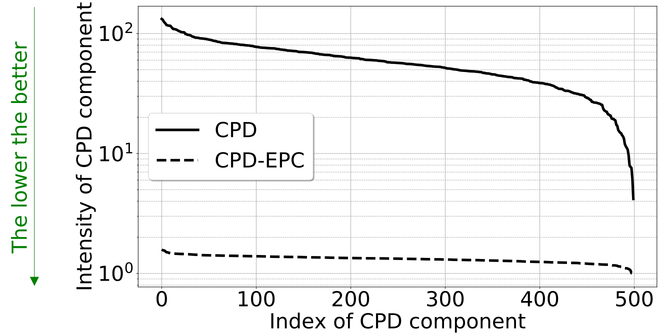


Figure: Intensity (Frobenius norm) of rank-1 tensors in CPDs of the kernel in the 4^{th} layer of ResNet-18.

Degeneracy and Sensitivity in CPD of Resnet18 II

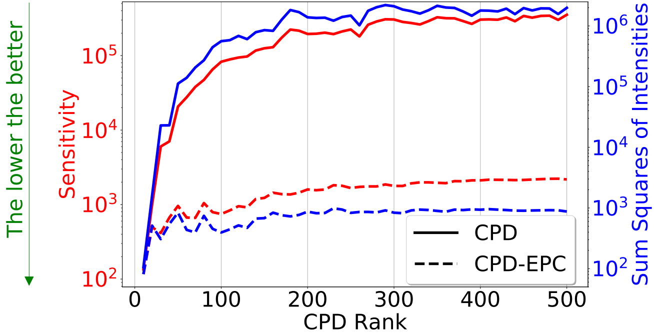


Figure: Sum of squares of the intensity and Sensitivity vs Rank of CPD.

Degeneracy occurs in most CPD of the convolutional kernels.

Degeneracy and Sensitivity in CPD I

Degeneracy causes

- Highly collinear components which are uninterpretable
- Instable decomposition
- Algorithms are getting stuck in local minima
  running a huge number of iterations does not improve the convergence
- Instable finetuning and hard to converge in practice.

Lundy et al. (1989); Harshman and Lundy (1984); Mitchell and Burdick (1994); Paatero (2000); Comon et al. (2009b); Harshman (2004); Krijnen et al. (2008a); Stegeman (2012).

Existing Methods to deal with Degeneracy I

- Improve stability and convergence by constraints,
  e.g., column-wise orthogonality Rayens and Mitchell (1997); Krijnen et al. (2008b),
  positivity or nonnegativity Paatero (1997); Lim and Comon (2009)
- Regularization, for example, a Tikhonov regularization

$$\min \quad \frac{1}{2}\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 + \frac{\mu}{2}\sum_n \|\mathbf{U}^{(n)}\|_F^2$$

  μ is adaptively adjusted and should converge to zero.
- Iterated Tikhonov regularization Navasca et al. (2008)

$$\min \quad \frac{1}{2}\|\mathcal{Y} - \hat{\mathcal{Y}}_{[i]}\|_F^2 + \frac{\mu}{2}\sum_n \|\mathbf{U}^{(n)}_{[i]} - \mathbf{U}^{(n)}_{[i-1]}\|_F^2$$

The above-mentioned methods may be useful in certain situations, but not always.

Sensitivity Measure in CPD I

Two possible measures for the degeneracy degree in the CPD

Definition (Norms of the rank-1 tensorsPhan et al. (2019))

$$\mathrm{sn}([\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]) = \sum_{r=1}^{R} \|\boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r\|_F^2 \tag{25}$$

Definition (Sensitivity Tichavský et al. (2019))

Given a tensor $\mathcal{T} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$, define the sensitivity as

$$\mathrm{ss}([\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]) = \lim_{\sigma^2 \to 0} \frac{1}{R\sigma^2} E\{\|\mathcal{T} - [\![\mathbf{A} + \delta\mathbf{A}, \mathbf{B} + \delta\mathbf{B}, \mathbf{C} + \delta\mathbf{C}]\!]\|_F^2\} \tag{26}$$

where $\delta\mathbf{A}$, $\delta\mathbf{B}$, $\delta\mathbf{C}$ have random i.i.d. elements from $N(0, \sigma^2)$.

Sensitivity Measure in CPD II

- Sensitivity of the tensor $\mathcal{T} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ is a measure with respect to perturbations in individual factor matrices.
- Decompositions with high sensitivity are usually useless.

Lemma

$$\mathrm{ss}([\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]) = K \,\mathrm{tr}\{(\mathbf{A}^T\mathbf{A}) \circledast (\mathbf{B}^T\mathbf{B})\} + I \,\mathrm{tr}\{(\mathbf{B}^T\mathbf{B}) \circledast (\mathbf{C}^T\mathbf{C})\} + J \,\mathrm{tr}\{(\mathbf{A}^T\mathbf{A}) \circledast (\mathbf{C}^T\mathbf{C})\}. \quad (27)$$

where $\circledast$ denotes the Hadamard element-wise product.

Proof.

$\square$

Sensitivity Measure in CPD III

First, the perturbed tensor in (26) can be expressed as sum of 8 Kruskal terms

$$\begin{aligned}[\![\mathbf{A} + \delta\mathbf{A}, \mathbf{B} + \delta\mathbf{B}, \mathbf{C} + \delta\mathbf{C}]\!] &= [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] + [\![\delta\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] + [\![\mathbf{A}, \delta\mathbf{B}, \mathbf{C}]\!] \\ &+ [\![\mathbf{A}, \mathbf{B}, \delta\mathbf{C}]\!] + [\![\delta\mathbf{A}, \delta\mathbf{B}, \mathbf{C}]\!] + [\![\delta\mathbf{A}, \mathbf{B}, \delta\mathbf{C}]\!] \\ &+ [\![\mathbf{A}, \delta\mathbf{B}, \delta\mathbf{C}]\!] + [\![\delta\mathbf{A}, \delta\mathbf{B}, \delta\mathbf{C}]\!].\end{aligned}$$

Since these Kruskal terms are uncorrelated and expectation of the terms composed by two or three factor matrices $\delta\mathbf{A}$, $\delta\mathbf{B}$ and $\delta\mathbf{C}$ are negligible, the expectation in (26) can be expressed in the form

$$\begin{aligned}E\{\|\mathcal{T} - [\![\mathbf{A} + \delta\mathbf{A}, \mathbf{B} + \delta\mathbf{B}, \mathbf{C} + \delta\mathbf{C}]\!]\|_F^2\} &= E\{\|[\![\delta\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]\|_F^2\} + \\ &+ E\{\|[\![\mathbf{A}, \delta\mathbf{B}, \mathbf{C}]\!]\|_F^2\} + E\{\|[\![\mathbf{A}, \mathbf{B}, \delta\mathbf{C}]\!]\|_F^2\}\,. \end{aligned} \tag{28}$$

Sensitivity Measure in CPD IV

Next we expand the Frobenius norm of the three Kruskal tensors

$$
\begin{aligned}
E\{\|[\![\delta\mathbf{A},\mathbf{B},\mathbf{C}]\!]\|_F^2\} &= E\{\|\,((\mathbf{C}\odot\mathbf{B})\otimes\mathbf{I})\,\mathrm{vec}(\delta\mathbf{A})\,\|^2\} \\
&= E\{\mathrm{tr}((\mathbf{C}\odot\mathbf{B})\otimes\mathbf{I})^T\,((\mathbf{C}\odot\mathbf{B})\otimes\mathbf{I})\,\mathrm{vec}(\delta\mathbf{A})\,\mathrm{vec}(\delta\mathbf{A})^T)\} \\
&= \sigma^2 I\,\mathrm{tr}((\mathbf{C}\odot\mathbf{B})^T(\mathbf{C}\odot\mathbf{B})\otimes\mathbf{I}) \\
&= RI\sigma^2\,\mathrm{tr}((\mathbf{C}^T\mathbf{C})\circledast(\mathbf{B}^T\mathbf{B})) \\
E\{\|[\![\mathbf{A},\delta\mathbf{B},\mathbf{C}]\!]\|_F^2\} &= RJ\sigma^2\,\mathrm{tr}((\mathbf{C}^T\mathbf{C})\circledast(\mathbf{A}^T\mathbf{A})) \\
E\{\|[\![\mathbf{A},\mathbf{B},\delta\mathbf{C}]\!]\|_F^2\} &= RK\sigma^2\,\mathrm{tr}((\mathbf{B}^T\mathbf{B})\circledast(\mathbf{A}^T\mathbf{A}))
\end{aligned}
$$

where $\odot$ and $\otimes$ are Khatri-Rao and Kronecker products, respectively.

Finally, we replace these above expressions into (28) to obtain the compact expression of sensitivity.

Sensitivity Measure in Tensor Decomposition I

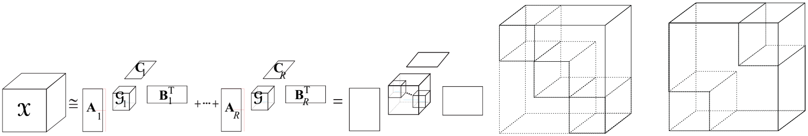


Figure: Illustration of non-overlapping and overlapping core tensors in BTD.

- CPD is not the only one tensor decomposition facing digeneracy, unstability
- Block term decompositionDe Lathauwer (2008), i.e., Tucker decompostion with block diagonal core tensor
- Most structured Tucker decomposition
- Tensor networks with loop, e.g., Tensor chain Landsberg (2012); Handschuh (2015).

Sensitivity Measure in Structured TKD I

$$\mathcal{T} \approx \mathcal{K} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$$

For Structured Tucker decomposition of order-3 tensors we define the sensitivity as

$$s(\mathcal{K}, \mathbf{A}, \mathbf{B}, \mathbf{C}) = \lim_{\sigma^2 \to 0} \frac{1}{\sigma^2} \mathrm{E}\{\| [[\mathcal{K} + \delta\mathcal{K}; \mathbf{A} + \delta\mathbf{A}, \mathbf{B} + \delta\mathbf{B}, \mathbf{C} + \delta\mathbf{C}]] - [[\mathcal{K}, \mathbf{A}, \mathbf{B}, \mathbf{C}]] \|_F^2\} \,, \quad (29)$$

where $\delta\mathcal{K}, \delta\mathbf{A}, \delta\mathbf{B}$ and $\delta\mathbf{C}$ are random Gaussian-distributed perturbations of the core tensor and the factor matrices with independently distributed elements of zero mean and variance σ^2 If some core tensor elements are constrained to be zero, the corresponding elements of $\delta\mathcal{K}$ are zero as well.

Sensitivity Measure in Structured TKD II

$$
\begin{aligned}
\mathrm{ss}(\Theta) &= \frac{1}{\sigma^2}\mathrm{E}\{\|[[\delta\mathcal{K}, \mathbf{A}, \mathbf{B}, \mathbf{C}]]\|_F^2\} + \frac{1}{\sigma^2}\mathrm{E}\{\|[[\mathcal{K}, \delta\mathbf{A}, \mathbf{B}, \mathbf{C}]]\|_F^2\} \\
&\quad + \frac{1}{\sigma^2}\mathrm{E}\{\|[[\mathcal{K}, \mathbf{A}, \delta\mathbf{B}, \mathbf{C}]]\|_F^2\} + \frac{1}{\sigma^2}\mathrm{E}\{\|[[\mathcal{K}, \mathbf{A}, \mathbf{B}, \delta\mathbf{C}]]\|_F^2\} \\
&= \mathbf{1}^T \mathrm{vec}\,[[\mathcal{L}, \mathbf{A} \star \mathbf{A}, \mathbf{B} \star \mathbf{B}, \mathbf{C} \star \mathbf{C}]] \\
&\quad + I_1 \,\|[[\mathcal{K}, \mathbf{I}_{R_1}, \mathbf{B}, \mathbf{C}]]\|_F^2 + I_2 \,\|[[\mathcal{K}, \mathbf{A}, \mathbf{I}_{R_2}, \mathbf{C}]]\|_F^2 \\
&\quad + I_3 \,\|[[\mathcal{K}, \mathbf{A}, \mathbf{B}, \mathbf{I}_{R_3}]]\|_F^2 \,, \qquad (30)
\end{aligned}
$$

where $\mathbf{1}$ is a vector composed of 1's, and $\mathcal{L}$ is the tensor indicating the nonzero elements of the core tensor $\mathcal{K}$.

Sensitivity for CPD I

For CPD, the core tensor, $\mathcal{K}$, comprises only R nonzero elements $\lambda_1, \lambda_2, \ldots, \lambda_R$ on its diagonal.

$$\begin{aligned} \mathtt{ss}_{\mathtt{CPD}}(\mathbf{A}, \mathbf{B}, \mathbf{C}) &= \sum_{r=1}^{R} \|\boldsymbol{a}_r\|^2 \|\boldsymbol{b}_r\|^2 \|\boldsymbol{c}_r\|^2 + I_1 \sum_{r=1}^{R} \lambda_r^2 \|\boldsymbol{b}_r\|^2 \|\boldsymbol{c}_r\|^2 \\ &\quad + I_2 \sum_{r=1}^{R} \lambda_r^2 \|\boldsymbol{a}_r\|^2 \|\boldsymbol{c}_r\|^2 + I_3 \sum_{r=1}^{R} \lambda_r^2 \|\boldsymbol{a}_r\|^2 \|\boldsymbol{b}_r\|^2 \end{aligned}$$

Due to scaling ambiguity, $\boldsymbol{a}_r$, $\boldsymbol{b}_r$ and $\boldsymbol{c}_r$ can be normalized $\|\boldsymbol{a}_r\| = \|\boldsymbol{b}_r\| = \|\boldsymbol{c}_r\| = 1$.
The sensitivity for CPD is then proportional to sum-of-square of λ_r, or norm of rank-1 tensors Phan et al. (2019)

$$\mathtt{ss}_{\mathtt{CPD}}(\mathbf{A}, \mathbf{B}, \mathbf{C}) = R + (I_1 + I_2 + I_3) \sum_{r=1}^{R} \lambda_r^2 \,,$$

Sensitivity for CPD II

Another form of the sensitivity for CPD is when the core tensor does not explicitly exist

$$\mathtt{ss}(\mathbf{A}, \mathbf{B}, \mathbf{C}) \quad = \sum_{r=1}^{R} I_1 \, ||\boldsymbol{b}_r||^2 ||\boldsymbol{c}_r||^2 + I_2 \, ||\boldsymbol{a}_r||^2 ||\boldsymbol{c}_r||^2 + I_3 \, ||\boldsymbol{a}_r||^2 ||\boldsymbol{b}_r||^2 \ .$$

Stable CP Decomposition I

- Deal with degeneracy
  Seeking a new tensor, $\hat{\mathcal{Y}}$ with minimal sensitivity, while preserving the approximation error
- Correct the rank-1 tensors if the estimation encounters degeneracy

$$\min_{\{\mathbf{A},\mathbf{B},\mathbf{C}\}} \quad ss([\![\mathbf{A},\mathbf{B},\mathbf{C}]\!]) \tag{31}$$
$$\text{s.t.} \quad \|\mathcal{K} - [\![\mathbf{A},\mathbf{B},\mathbf{C}]\!]\|_F^2 \leq \delta^2 \,.$$

The bound, δ^2, can represent the approximation error of the decomposition with diverging components.

Phan, Tichavsky and Cichocki, Error Preserving Correction: A Method for CP Decomposition at a Target Error Bound, IEEE Transaction on Signal Processing, 67(5), 1175-1190,2019.

Phan, et al., Stable low-rank tensordecomposition for compression of convolutional neural network, in ECCV 2020.

Alternating Correction Method for Stable CPD I

- Update **A** while **B** and **C** are kept fixed

$$\begin{aligned} \min_{\mathbf{A}} \quad & \operatorname{tr}\{(\mathbf{A}^T\mathbf{A}) \circledast \mathbf{W}\} = \|\mathbf{A}\operatorname{diag}(\boldsymbol{w})\|_F^2 \\ \text{s.t.} \quad & \|\mathbf{K}_{(1)} - \mathbf{A}\mathbf{Z}^T\|_F^2 \le \delta^2, \end{aligned}$$

where $\mathbf{Z} = \mathbf{C} \odot \mathbf{B}$ and $\mathbf{W} = \mathbf{B}^T\mathbf{B} + \mathbf{C}^T\mathbf{C}$, $\boldsymbol{w} = [\sqrt{w_{1,1}}, \ldots, \sqrt{w_{R,R}}]$ is a vector of length R taken from the diagonal of **W**.

Remark

The decomposition can be reformulated as regression with bound constraint which can be solved in closed form through the quadratic programming over a sphere Phan et al. (2019)

$$\begin{aligned} \min_{\widetilde{\mathbf{A}}} \quad & \|\widetilde{\mathbf{A}}\|_F^2 \\ \text{s.t.} \quad & \|\mathbf{K}_{(1)} - \widetilde{\mathbf{A}}\widetilde{\mathbf{Z}}^T\|_F^2 \le \delta^2, \end{aligned}$$

where $\widetilde{\mathbf{A}} = \mathbf{A}\operatorname{diag}(\boldsymbol{w})$ and $\widetilde{\mathbf{Z}} = \mathbf{Z}\operatorname{diag}(\boldsymbol{w}^{-1})$.

Alternating Correction Method for Stable CPD II

Remark

If **B** and **C** are normalized $\|\boldsymbol{b}_r\|_2 = \|\boldsymbol{c}_r\|_2 = 1$, then all entries of the diagonal of **W** are identical, we seek a matrix, **A**, with minimal norm

$$\begin{aligned} \min_{\mathbf{A}} \quad & \|\mathbf{A}\|_F^2 \\ \text{s.t.} \quad & \|\mathbf{K}_{(1)} - \mathbf{A}\mathbf{Z}^T\|_F^2 \leq \delta^2. \end{aligned} \tag{32}$$

Error Preserving Correction Method I

EPC

Correct the rank-1 tensors if their norms are relatively high

$$\begin{array}{ll} \min & f(\boldsymbol{\theta}) = \|\boldsymbol{\eta}\|_2^2 = \sum_{r=1}^{R} \eta_r^2 \\ \text{s.t.} & c(\boldsymbol{\theta}) = \|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 \leq \delta^2, \end{array}$$

- where $\boldsymbol{\theta} = [\boldsymbol{u}_r^{(n)}, \boldsymbol{\eta}]$, $\|\boldsymbol{u}_r^{(n)}\| = 1$

$$\mathcal{Y} \approx \hat{\mathcal{Y}} = \sum_{r=1}^{R} \eta_r \, \boldsymbol{u}_r^{(1)} \circ \boldsymbol{u}_r^{(2)} \circ \cdots \circ \boldsymbol{u}_r^{(N)}$$

- Constraint function is nonlinear in all the factor matrices, but linear in parameters in a single factor matrix, or in non-overlapping partitions of different factor matrices Tichavský et al. (2016); Phan et al. (2017).

Error Preserving Correction Method II

- Rewrite the objective function and especially the constraint function in a linear form.
- Then solve the problem using the alternating update scheme or the Sequential Quadratic Programming method Boggs and Tolle (1995); Fletcher (2000); Phan et al. (2019).

Alternating Correction Method I

EPC

$$\min \quad f(\boldsymbol{\theta}) = \|\boldsymbol{\eta}\|_2^2 \quad \text{s.t.} \quad c(\boldsymbol{\theta}) = \|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 \le \delta^2,$$

- At each iteration, new estimate of $\mathbf{U}^{(n)}$ is found by solving a Spherical Constrained Quadratic Programming problem
- Let $\mathbf{U}_\eta^{(n)} = \mathbf{U}^{(n)} \operatorname{diag}(\boldsymbol{\eta})$, the objective function becomes

$$\|\boldsymbol{\eta}\|^2 = \|\mathbf{U}^{(n)} \operatorname{diag}(\boldsymbol{\eta})\|_F^2 = \|\mathbf{U}_\eta^{(n)}\|_F^2 .$$

Alternating Correction Method II

- EPC becomes *linear regression with bound constraint*

$$\begin{aligned} \min \quad & \|\mathbf{U}_\eta^{(n)}\|_F^2 \\ \text{s.t.} \quad & \|\mathbf{F}_n \Sigma^{\frac{-1}{2}} - \mathbf{U}_\eta^{(n)} \mathbf{V}_n \Sigma^{\frac{1}{2}}\|_F^2 \leq \delta_n^2 \end{aligned}$$

where $\delta_n^2 = \delta^2 + \|\mathbf{F}_n \Sigma^{\frac{-1}{2}}\|_F^2 - \|\mathcal{Y}\|_F^2$
$\mathbf{T}_n = \bigodot_{k \neq n} \mathbf{U}^{(k)}$,
$\mathbf{G}_n = \mathbf{Y}_{(n)} \mathbf{T}_n$ of size $I_n \times R$,
$\Gamma_{-n} = \mathbf{T}_n^T \mathbf{T}_n = \circledast_{k \neq n} (\mathbf{U}^{(k)T} \mathbf{U}^{(k)})$ of size $R \times R$
EVD of $\Gamma_{-n} = \mathbf{V}_n \Sigma \mathbf{V}_n^T$
$\mathbf{F}_n = \mathbf{G}_n \mathbf{V}_n$,

Alternating Correction Method III

- The inequality constraint can be replaced by an equality constraint.

Lemma

The minimizer in

$$\min_{\boldsymbol{x}} \quad \|\boldsymbol{x}\|^2 \quad s.t. \quad \|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\| \leq \delta,$$

is equivalent to the minimizer of the following optimization problem

$$\min_{\boldsymbol{x}} \quad \|\boldsymbol{x}\|^2 \quad s.t. \quad \|\boldsymbol{y} - \mathbf{A}\boldsymbol{x}\| = \delta.$$

Alternating Correction Method IV

- Optimization problem for update of $\mathbf{U}_\eta^{(n)}$

$$\begin{array}{cc} \min & \|\mathbf{U}_\eta^{(n)}\|_F^2 \\ \text{s.t.} & \|\mathbf{F}_n\Sigma^{\frac{-1}{2}} - \mathbf{U}_\eta^{(n)}\mathbf{V}_n\Sigma^{\frac{1}{2}}\|_F^2 = \delta_n^2 \end{array}$$

- Perform a reparameterization

$$\begin{aligned} \mathbf{Z}_n &= \frac{1}{\delta_n}(\mathbf{F}_n\Sigma^{\frac{-1}{2}} - \mathbf{U}_\eta^{(n)}\mathbf{V}_n\Sigma^{\frac{1}{2}}), \\ \mathbf{U}_\eta^{(n)} &= (\mathbf{F}_n\Sigma^{\frac{-1}{2}} - \delta_n \mathbf{Z}_n)\,\Sigma^{\frac{-1}{2}}\,\mathbf{V}_n^T, \end{aligned}$$

and represent the Frobenius norm of $\mathbf{U}_\eta^{(n)}$ as

$$\|\mathbf{U}_\eta^{(n)}\|_F^2 = \|\mathbf{F}_n\Sigma^{-1}\|_F^2 + \delta_n^2\,\mathrm{tr}(\mathbf{Z}_n\,\Sigma^{-1}\,\mathbf{Z}_n^T) - 2\delta_n\,\mathrm{tr}(\mathbf{F}_n\Sigma^{\frac{-3}{2}}\mathbf{Z}_n^T)\,.$$

Alternating Correction Method V

Spherical Constrained Quadratic Programming (SCQP)

Instead of updating $\mathbf{U}_{\eta}^{(n)}$, we seek $\mathbf{Z}_n$ of size $I_n \times R$ in an SCQP for matrix-variate

$$\begin{array}{ll} \min & \delta_n \operatorname{tr}(\mathbf{Z}_n \Sigma^{-1} \mathbf{Z}_n^T) - 2 \operatorname{tr}(\mathbf{F}_n \Sigma^{\frac{-3}{2}} \mathbf{Z}_n^T) \\ \text{s.t.} & \|\mathbf{Z}_n\|_F^2 = 1. \end{array}$$

- The above SCQP has $I_n R$ terms and can be solved in closed-form Gander et al. (1989); Phan et al. (2019).
- We next formulate another equipvalent SCQP problem but with smaller number of parameters

Alternating Correction Method VI

Lemma (SCQP with Matrix-variates)

A SCQP for a matrix $\mathbf{X}$ *of size* $I \times R$ *given in the form of*

$$\min_{\mathbf{X}} \quad f(\mathbf{X}) = \frac{1}{2}\,\mathrm{tr}(\mathbf{X}^T\mathbf{Q}\mathbf{X}) + \mathrm{tr}(\mathbf{B}^T\mathbf{X}) \quad s.t. \quad \|\mathbf{X}\|_F^2 = 1\,, \tag{33}$$

where $\mathbf{Q}$ *is a positive semidefinite matrix of size* $I \times I$ *and has EVD* $\mathbf{Q} = \mathbf{U}\,\mathrm{diag}(\boldsymbol{\sigma})\mathbf{U}^T$, *and* $\mathbf{B}$ *is of size* $I \times R$.
Then the optimal solution

$$\mathbf{X}^\star = \mathbf{B}^T\mathbf{U}\,\mathrm{diag}(\boldsymbol{z}^\star \oslash \boldsymbol{c})$$

can be derived from the minimizer $\boldsymbol{z}^\star$ *of length* I *of an SCQP*

$$\min \quad \frac{1}{2}\boldsymbol{z}^T\,\mathrm{diag}(\boldsymbol{\sigma})\boldsymbol{z} + \boldsymbol{c}^T\boldsymbol{z} \quad s.t. \quad \boldsymbol{z}^T\boldsymbol{z} = 1\,, \tag{34}$$

where $\boldsymbol{c} = [c_1, \ldots, c_I]$, $c_i = \|\mathbf{B}^T\boldsymbol{u}_i\|$.

Alternating Correction Method VII

- The minimizer $\mathbf{Z}_n^\star$ can be derived from the minimizer $\boldsymbol{z}^\star = [z_1^\star, \ldots, z_R^\star]^T$ of an SCQP of a smaller scale

$$\min \qquad \delta_n \boldsymbol{z}^T \, \Sigma^{-1} \, \boldsymbol{z}^T - 2\boldsymbol{c}^T \boldsymbol{z} \quad \text{s.t.} \quad \boldsymbol{z}^T \boldsymbol{z} = 1$$

where $\boldsymbol{c} = [\ldots, \sigma_r^{\frac{-3}{2}} \|\boldsymbol{f}_r^{(n)}\|, \ldots]$.

  - For $c_r \neq 0$, the r-th column of $\mathbf{Z}_n^\star$ is the r-th column of $\mathbf{F}_n$ scaled by a factor of $\frac{z_r^\star}{\|\boldsymbol{f}_r^{(n)}\|}$

$$\boldsymbol{z}_r^{(n)\star} = \frac{z_r^\star}{\|\boldsymbol{f}_r^{(n)}\|} \, \boldsymbol{f}_r^{(n)} \, .$$

  - For $c_r = 0$, $\boldsymbol{z}_r^{(n)\star}$ can be any vector of length $\|\boldsymbol{z}_r^{(n)\star}\|^2 = (z_r^\star)^2$.
  - If $c_r = 0$ for $r > 1$, then $z_r^\star = 0$, hence $\boldsymbol{z}_r^{(n)\star}$ is a zero vector, Phan et al. (2019).

Alternating Correction Algorithm

Algorithm 3: `Alternating Error Preserving Correction`

Input: Data tensor $\mathcal{Y}$: $(I_1 \times I_2 \times \cdots \times I_N)$, and a rank R and error bound δ

Output: $\mathcal{X} = [\![\boldsymbol{\eta}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!]$ s.t. $\min \|\boldsymbol{\eta}\|_2^2$ s.t. $\|\mathcal{Y} - \mathcal{X}\|_F^2 \leq \delta^2$

begin

- Initialize $\mathcal{X} = [\![\boldsymbol{\eta}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!]$ such that $\|\mathcal{Y} - \mathcal{X}\|_F^2 \leq \delta^2$
- **repeat**
  - **for** $n = 1, 2, \ldots, N$ **do**
    - Compute $\mathbf{G}_n = \mathbf{Y}_{(n)} \left(\bigodot_{k \neq n} \mathbf{U}^{(k)} \right)$
    - Compute EVD of $\boldsymbol{\Gamma}_{-n} = \circledast_{k \neq n} (\mathbf{U}^{(k)T} \mathbf{U}^{(k)}) = \mathbf{V}_n \boldsymbol{\Sigma}_n \mathbf{V}_n^T$ and $\mathbf{F}_n = \mathbf{G}_n \mathbf{V}_n$
    - Solve an SCQP: $\min \quad \delta_n \boldsymbol{z}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{z}^T - 2\boldsymbol{c}^T \boldsymbol{z} \quad$ s.t. $\boldsymbol{z}^T \boldsymbol{z} = 1$
    - $\mathbf{Z}_n = [\ldots, \frac{z_r}{\|\boldsymbol{f}_r^{(n)}\|} \boldsymbol{f}_r^{(n)}, \ldots]$
    - $\mathbf{U}_\eta^{(n)} = (\mathbf{F}_n \boldsymbol{\Sigma}^{\frac{-1}{2}} - \delta_n \mathbf{Z}_n) \boldsymbol{\Sigma}^{\frac{-1}{2}} \mathbf{V}_n^T$
    - Update $\boldsymbol{\eta}$ and $\mathbf{U}^{(n)}$: $\eta_r = \|\boldsymbol{u}_{\eta,r}^{(n)}\|$, $\boldsymbol{u}_r^{(n)} = \frac{\boldsymbol{u}_{\eta,r}^{(n)}}{\eta_r}$
  - **end**
- **until** *a stopping criterion is met*

end

Relation between ACEP and ALS I

- Assume the first column of $\mathbf{F}_n$ is non-zero, i.e., $c_1 \neq 0$, hence, $z_1^\star \neq 0$, and

$$\mathbf{Z}_n^\star = \bar{\mathbf{F}}_n \operatorname{diag}([\ldots, z_r^\star, \ldots]), \tag{35}$$

where columns of $\bar{\mathbf{F}}_n$ are $\frac{\boldsymbol{f}_r^{(n)}}{\|\boldsymbol{f}_r^{(n)}\|}$ for non zero columns $\boldsymbol{f}_r^{(n)}$, and zero vectors elsewhere.

- Since $\boldsymbol{f}_1^{(n)}$ is non-zero, the minimizer $\boldsymbol{z}^\star$ of the SCQP in (35) is given in closed-form as

$$z_r^\star = \frac{c_r}{\|\boldsymbol{c}\|(s_r - \lambda)} = \frac{\|\boldsymbol{f}_r(n)\|\sigma_r^{\frac{-3}{2}}}{\|\boldsymbol{c}\|(s_r - \lambda)} \tag{36}$$

where $s_r = 1 + \frac{\delta_n}{\|\boldsymbol{c}\|}(\sigma_r^{-1} - \sigma_1^{-1})$, and λ is a unique root in $[0, 1)$ of a secular function Gander et al. (1989); Phan et al. (2019)

$$\sum_r (z_r^\star)^2 = 1$$

Relation between ACEP and ALS II

- The new update of $\mathbf{U}_\eta^{(n)}$ can be expressed in a compact form as

$$\begin{aligned}\mathbf{U}_\eta^{(n)} &= \left(\mathbf{F}_n \Sigma^{\frac{-1}{2}} - \delta_n \bar{\mathbf{F}}_n \operatorname{diag}([\ldots, z_r^\star, \ldots])\right) \Sigma^{\frac{-1}{2}} \mathbf{V}_n^T \\ &= \mathbf{F}_n \operatorname{diag}\left(1 - \frac{\delta_n}{\|\boldsymbol{c}\| \sqrt{\sigma_r}(s_r - \lambda)}\right) \Sigma^{-1} \mathbf{V}_n^T . \qquad (37)\end{aligned}$$

- Observed that only when $\delta_n = 0$, the above update (37) boils down to the ordinary ALS update for CPD

$$\mathbf{U}_\eta^{(n)} = \mathbf{Y}_{(n)} \mathbf{T}_n \Gamma_{-n}^{-1}.$$

Algorithm 4: `CPD with EPC`

Input: Data tensor $\mathcal{Y}$: $(I_1 \times I_2 \times \cdots \times I_N)$, and a rank R and rank-1 norm bound ε

Output: $\mathcal{X} = [\![\boldsymbol{\eta}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \ldots, \mathbf{U}^{(N)}]\!]$ of rank R

begin

- Initialize $\mathcal{X}_{[0]} = [\![\boldsymbol{\eta}_0; \mathbf{U}_0^{(1)}, \mathbf{U}_0^{(2)}, \ldots, \mathbf{U}_0^{(N)}]\!]$
- **repeat**
  - **if** $\|\boldsymbol{\eta}_k\|_2^2 \geq \varepsilon^2$ **then**
    - Apply EPC to find $\mathcal{X}_{[k+1]}$: $\min \quad \|\boldsymbol{\eta}_{k+1}\|_2^2$ s.t $\|\mathcal{Y} - \mathcal{X}_{[k+1]}\|_F \leq \|\mathcal{Y} - \mathcal{X}_{[k]}\|_F$
  - **else**
    - Find $\mathcal{X}_{[k+1]}$ in CPD of $\mathcal{Y}$ initialized by $\mathcal{X}_{[k]}$
  - **end**
- **until** *a stopping criterion is met*

end

Tensor Decomposition with bounded sensitivity

- Instead of correcting solutions with divegerging loading components, we can seek a decomposition with bounded sensitivity

$$\begin{aligned} \min \quad & \|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 \\ \text{s.t.} \quad & \mathrm{ss}([\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]) \leq \delta. \end{aligned}$$

Low-rank Tensor Approximation for Alexnet I

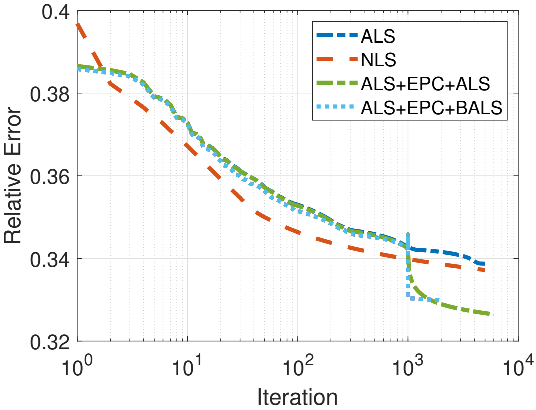


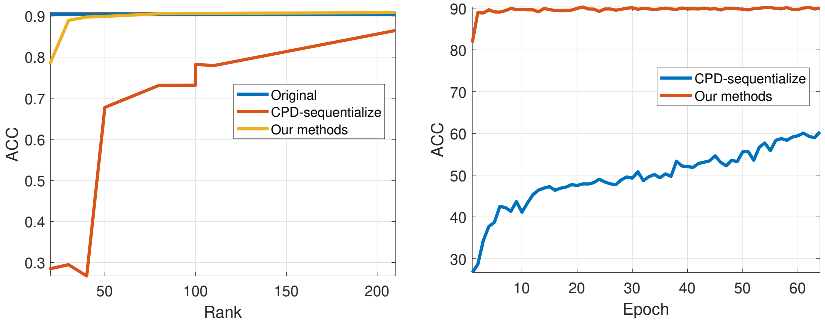
Figure: Convolution tensor at Layer-10 in AlexNet of size $(3 \times 3) \times 256 \times 384$. Both ALS and NLS converge slowly because of the high norm of rank-1 tensors. We apply EPC after 1000 iterations of ALS, then continue CPD. The algorithm attained significantly lower approximation error.

Low-rank Tensor Approximation for Alexnet II

Table: Performance comparison for algorithms.

	ALS +line search	NLS	ALS+EPC +ALS	ALS+EPC +BALS
Relative Error	0.3387	0.3372	0.3254	0.3240
Running time (s)	254.2	1087.7	283.5	1080.8
Norm of Rank-one tensors	2.08×10^8	4.75×10^8	3.14×10^4	4.43×10^3

Example: Stable CPD of convolutional kernels in Alexnet I



Example: Stable BTD of convolutional kernels in Alexnet CNN I

- We used four convolutional kernels in the Alexnet. These tensors have the order 4, but were reshaped to order-3 tensors by folding their first two dimensions.
- This example demonstrates an application of the `BTD` with sensitivity control in compression of CNN.
- Four tensors are decomposed by `BTD` models with 15 blocks of the size $2 \times 2 \times 2$, so that the core tensor had the size $30 \times 30 \times 30$.

Example: Stable BTD of convolutional kernels in Alexnet CNN II

Table: Comparison of KLM-bnd and SDF-NLS algorithms on example with real-world tensors.

layer no./ size	Algorithm	Sensitivity	Relative fitting error
2	KLM-bnd	**6.5307e+06**	0.75058
$25 \times 48 \times 256$	SDF-NLS	1.8305e+12	0.75302
3	KLM-bnd	**1.7378e+08**	0.85923
$9 \times 256 \times 384$	SDF-NLS	1.0404e+12	0.85952
4	KLM-bnd	**1.3588e+08**	0.90215
$9 \times 192 \times 384$	SDF-NLS	1.055e+11	0.90224
5	KLM-bnd	**2.1769e+07**	0.85361
$9 \times 192 \times 256$	BTD-NLS	6.2318e+10	0.85397

Example: Stable BTD of convolutional kernels in Alexnet CNN III

- NLS for BTDSorber et al. (2016) and KLM with bounded sensitivity achieve approximately the same fitting error, but our algorithm provides the estimate with a much smaller sensitivity measure.

Example: ResNet-18 with Stable Factorized-Layers I

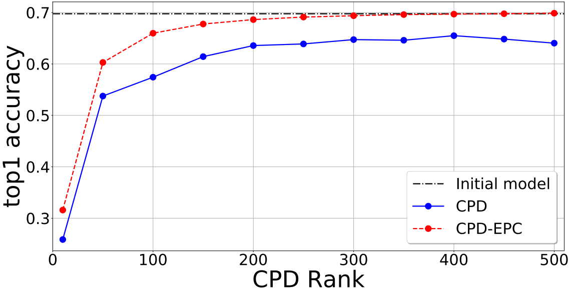


Figure: Performance evaluation of ResNet-18 on ILSVRC-12 dataset after replacing `layer4.1.conv1` by its approximation using CPD and stable CPD. The networks are fine-tuned after compression

Example: ResNet-18 with Stable Factorized-Layers II

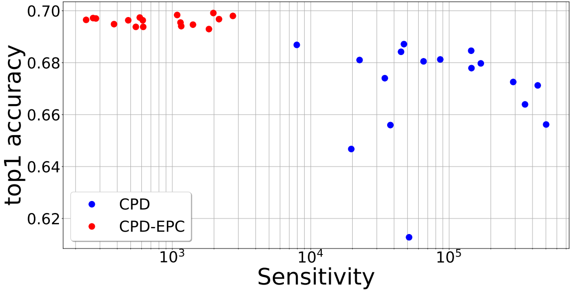


Figure: Top-1 accuracy and sensitivity of the models estimated using CPD (blue) and stable CPD(red). Each model has a single decomposed layer with the best CP rank and was fine-tuned after compression. StableCPD outperforms CPD in terms of accuracy/sensitivity trade-off.

Example: ResNet-18 with Stable Factorized-Layers III

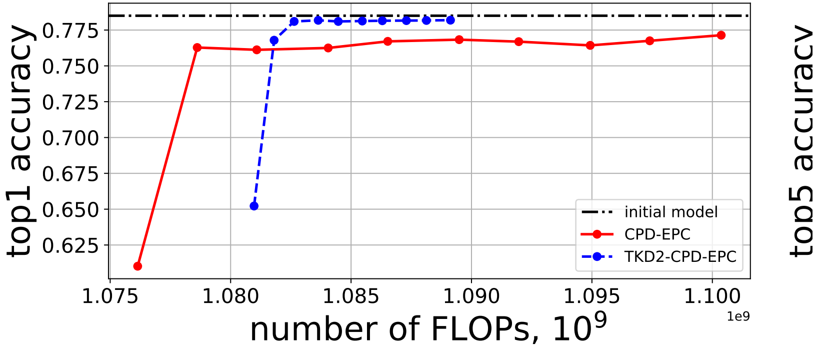


Figure: Top1 accuracy using stable CPD and stable TKD-CPD in compression of the layer `4.0.conv1` in the pre-trained ResNet-18 on ILSVRC-12 dataset. Initial model has $\approx 1.11 \times 10^9$ FLOPs.

Stable TKD-CPD-EPC shows better accuracy recovery with a relatively low number of FLOPs.

Example: Full model compression I

Table: Comparison of different model compression methods on ILSVRC-12 validation dataset. The baseline models are taken from Torchvision.

Model	Method	↓ FLOPs	Δ top-1	Δ top-5
VGG-16	Asym. **?**	≈ 5.00	-	-1.00
	TKD+VBMF **?**	4.93	-	-0.50
	Our (EPS[1]=0.005)	**5.26**	**-0.92**	**-0.34**
ResNet-18	Channel Gating **?**	1.61	-1.62	-1.03
	Channel Pruning **?**	1.89	-2.29	-1.38
	FBS **?**	1.98	-2.54	-1.46
	MUSCO **?**	2.42	-0.47	-0.30
	Our (EPS[1]=0.00325)	**3.09**	**-0.69**	**-0.15**
ResNet-50	**Our** (EPS[1]=0.0028)	**2.64**	**-1.47**	**-0.71**

[1] EPS: accuracy drop threshold. Rank of the decomposition is chosen to maintain the drop in accuracy lower than EPS.

ResNet-50 for Image Classification I

We use a pretrained model shipped with Torchvision as a baseline for ILSVRC-12.

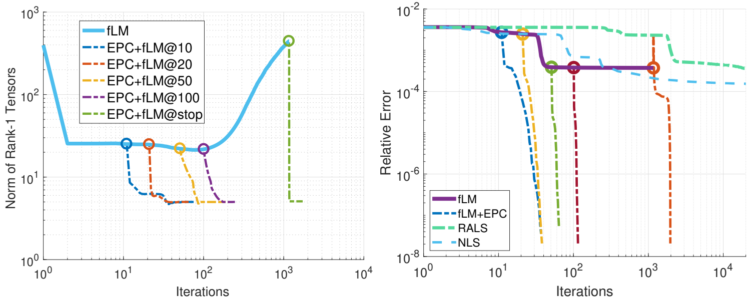
Table: ResNet-50 models for image classification on ImageNet. Original model has 25.6M parameters and 8.2B FLOPs. Rank search over the grid with rank step 1 and 16.

Model	FLOPs	No.param.	Top-1 acc., %	Top-5 acc., %
Original	1.00×	1.00×	76.13	92.86
Compressed 1^0	2.64×	2.51×	74.66(-1.47)	92.15(-0.71)
Compressed 16^0	2.32×	2.39×	74.66(-1.47)	92.35(-0.51)

Table: Inference time and acceleration for ResNet-50 on different platforms.

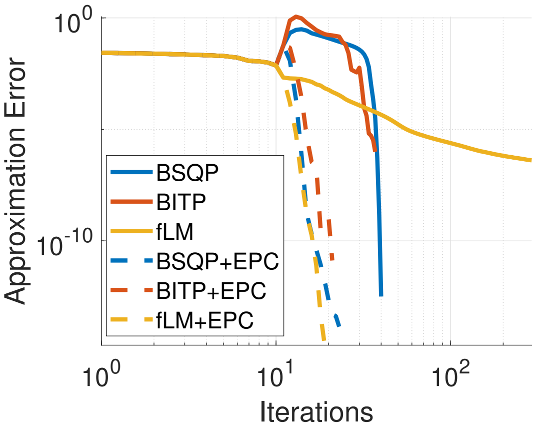
Platform	Model inference time	
	Original	Compressed
Intel® Xeon®Silver 4114 CPU 2.20 GHz	3.92 ± 0.02 s	2.84 ± 0.02 s
NVIDIA®Tesla®V100	102.3 ± 0.5 ms	89.5 ± 0.2 ms
Qualcomm®Snapdragon™845	221 ± 4 ms	171 ± 4 ms

Tensors with Highly Collinear Components



- Small tensor of size $4 \times 4 \times 4$ and rank-$R = 5$. The first four loading components were highly collinear, with $\boldsymbol{u}_r^{(n)T} \boldsymbol{u}_s^{(n)} = 0.99$ for all n and $1 \leq r \neq s \leq R - 1$.
- fLM, RALS and NLS do not converge, while the norm of rank-1 tensors increased dramatically (dashed blue curve in (left)).
- After running EPC, fLM converges quickly (dashed curves).
- EPC+fLM@10: apply EPC after 10 iterations of fLM

Decomposition of the Multiplication Tensor



- Decomposition of the multiplicative tensors of size $9 \times 9 \times 9$ with rank of $R = 23$
- EPC after 10 iterations of fLM boots the convergence of CPD. EPC+fLM converges after 20 iterations

EPC - A Tool for Decomposition at a Target Error Bound I

- When $\mathcal{Y}$ is corrupted by e.g., additive Gaussian noise, the constraint error bound in EPC $\delta^2 = \|\mathcal{E}\|_F^2$.
- Assuming that $\hat{\mathcal{Y}}^\star$ is with the smallest rank $R^\star$ such that $\|\mathcal{Y} - \hat{\mathcal{Y}}^\star\|_F^2 = \delta^2$.

When the estimated rank exceeds the true trank $R^\star$, the decomposition explains the noise, The higher the estimated tensor rank, the more noise the tensor $\hat{\mathcal{Y}}$ will explain.

When the noise is dominant to the data, approximation of $\mathcal{Y}$ with its true rank often yields an overestimate.

Determine numerical rank R which holds the error bound

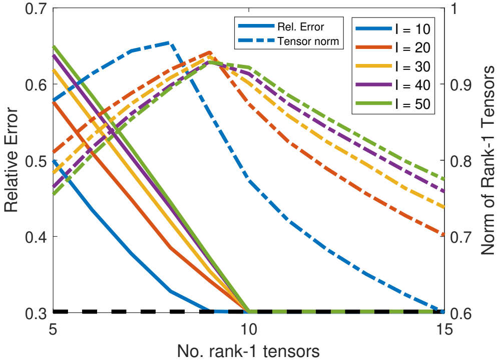


Figure: Tensors of rank 10 corrupted by Gaussian noise at SNR = 10dB.

EPC can maintain the approximation at a target error bound, i.e., $\|\mathcal{Y} - \hat{\mathcal{Y}}\|_F^2 = \delta^2$ when $R \geq R^\star$, the true rank.

- When EPC does not reach the bound, the estimated rank is relatively low, $R < R^\star$
- When EPC attains the error bound, the rank of $\hat{\mathcal{Y}}$ may not be minimal. Try a smaller rank.

Image denoising I

- Color images $\mathcal{T}$ degraded by additive Gaussian noise at SNR = 10 dB
- Constructed tensors, $\mathcal{Y}_{r,c}$, of a size $w \times w \times 3 \times (2d+1) \times (2d+1)$

$$\mathcal{Y}_{r,c}(:,:,:,d+1+i,d+1+j) = \mathcal{T}_{r+i,c+j}$$

  comprising $(2d+1)^2$ blocks, around the block $\mathcal{T}_{r,c} = \mathcal{T}(r:r+w-1, c:c+w-1, :)$, where $i,j = -d,\ldots,0,\ldots,d$, and d represents the neighbour width.
- Each tensor $\mathcal{Y}_{r,c}$ was then approximated by EPC, where $\delta^2 = 3\sigma^2 w^2 (2d+1)^2$, and σ the noise level.
- DCT spatial filtering used as a preprocessing.
- EPC with the tensor rank of 8, and adjusted it to attain the error bound.

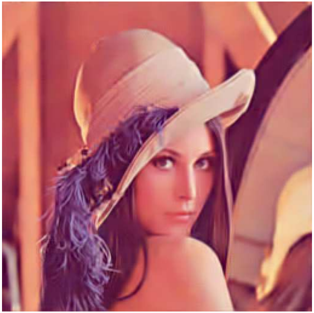
Image denoising II

Table: The performance comparison of algorithms considered in terms of PSNR (dB) and SSIM for image denoising when SNR = 10 dB.

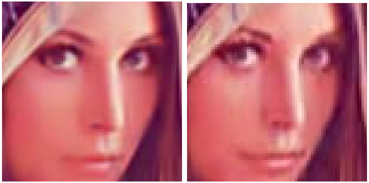
Algorithms	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
	Lena		Tiffany		Barbara	
EPC	**33.34**	**0.924**	**36.01**	**0.932**	**33.73**	**0.927**
TT-SVD	32.68	0.892	35.47	0.913	33.04	0.901
Tucker	32.74	0.919	35.47	0.926	32.92	0.919
BRTF	32.07	0.840	35.07	0.888	33.10	0.899
K-SVD	32.72	0.908	35.61	0.928	32.64	0.908
	Pepper		Pens		House	
EPC	**33.15**	**0.926**	**32.20**	**0.896**	**35.41**	**0.895**
TT-SVD	32.07	0.861	31.61	0.884	34.38	0.877
Tucker	32.23	0.917	31.27	0.884	34.40	0.885
BRTF	31.42	0.825	31.82	0.877	33.67	0.823
K-SVD	32.60	0.918	31.14	0.862	34.67	0.881

Image denoising III

(a) Noisy image at SNR = 10 dB



(b) EPC, SSIM = 0.924

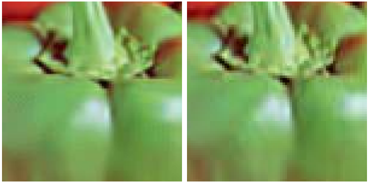


(c) From left to right, EPC(SSIM = 0.924), Tucker(0.919), TT-SVD(0.892) and K-SVD(0.908)

Image denoising IV

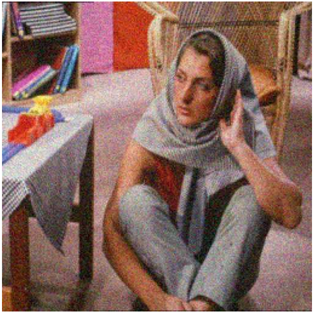
(d) Noisy image at SNR = 10 dB

(e) EPC, SSIM = 0.926



(f) From left to right, EPC(SSIM = 0.926), Tucker(0.917), TT-SVD(0.861) and K-SVD(0.918)

Image denoising V



(g) Noisy image at SNR = 10 dB

(h) EPC, SSIM = 0.927

(i) From left to right, EPC(SSIM = 0.927), Tucker(0.919), TT-SVD(0.901) and K-SVD(0.908)

Conclusions I

- Most tensor decompositions encounter diverging components (CPD, BTD, Tensor network with loop)
- Factorized layers with nonlinear ReLu layers work like a TD with nonnegativity constraints and can suffer the degeneracy as CPD or BTD.
- Stable tensor decomposition with bounded sensititivy make the decomposition solution interpretable, CNN with factorized layers easy to finetune and attain high accuracy
- Experiment results confirm superiority of the stable CPD over standard CPD in compression of CNNs.

References I

Berry, M., Browne, M., Langville, A., Pauca, P., and Plemmons, R. (2007). Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics and Data Analysis*, 52(1):155–173.

Boggs, P. T. and Tolle, J. W. (1995). Sequential quadratic programming. *Acta Numerica*, 4:1–51.

Booksh, K. and Kowalski, B. (1994). Error analysis of the generalized rank annihilation method. *Journal of Chemometrics*, 8.

Booksh, K., Lin, Z., Wang, Z., and Kowalski, B. (1994). Extension of trilinear decomposition method with an application to the flow probe sensor. *Analytical Chemistry*, 66.

Carroll, J. and Chang, J. (1970). Analysis of individual differences in multidimensional scaling via an *n*-way generalization of Eckart–Young decomposition. *Psychometrika*, 35(3):283–319.

References II

Comon, P., Luciani, X., and de Almeida, A. L. F. (2009a). Tensor decompositions, alternating least squares and other tales. *Journal of Chemometrics*, 23.

Comon, P., Luciani, X., and de Almeida, A. L. F. (2009b). Tensor decompositions, alternating least squares and other tales. *Journal of Chemometrics*, 23(7-8):393–405.

Comon, P. and Rajih, M. (2006). Blind identification of under-determined mixtures based on the characteristic function. *Signal Processing*, 86(9):2271 – 2281. Special Section: Signal Processing in {UWB} Communications.

De Lathauwer, L. (2008). Decompositions of a higher-order tensor in block terms – Part I and II. *SIAM Journal on Matrix Analysis and Applications (SIMAX)*, 30(3):1022–1066. Special Issue on Tensor Decompositions and Applications.

References III

Faber, N., Buydens, L., and Kateman, G. (1994). Generalized rank annihilation. iii: Practical implementation. *Journal of Chemometrics*, 8.

Fletcher, R. (2000). *Nonlinear Programming*, pages 229–258. John Wiley & Sons, Ltd.

Gander, W., Golub, G. H., and von Matt, U. (1989). A constrained eigenvalue problem. *Special Issue Dedicated to Alan J. Hoffman, Linear Algebra and its Applications*, 114:815 – 839.

Handschuh, S. (2015). *Numerical Methods in Tensor Networks*. PhD thesis, Facualty of Mathematics and Informatics, University Leipzig, Germany, Leipzig, Germany.

Harshman, R. and Lundy, M. (1984). *Data preprocessing and the extended Parafac model*. Praeger, New York, USA.

References IV

Harshman, R. A. (1972). Determination and proof of minimum uniqueness conditions for PARAFAC1. *UCLA Working Papers in Phonetics*, 22.

Harshman, R. A. (2004). The problem and nature of degenerate solutions or decompositions of 3-way arrays. In *Paper presented at the American Institute of Mathematics Tensor Decomposition Workshop*, Palo Alto, CA.

Hitchcock, F. (1927). Multiple invariants and generalized rank of a p-way matrix or tensor. *Journal of Mathematics and Physics*, 7:39–79.

Jaderberg, M., Vedaldi, A., and Zisserman, A. (2014). Speeding up convolutional neural networks with low rank expansions. *CoRR*, abs/1405.3866.

Kolda, T. and Bader, B. (2009). Tensor decompositions and applications. *SIAM Review*, 51(3):455–500.

References V

Krijnen, W., Dijkstra, T., and Stegeman, A. (2008a). On the non-existence of optimal solutions and the occurrence of "degeneracy" in the Candecomp/Parafac model. *Psychometrika*, 73:431–439.

Krijnen, W. P., Dijkstra, T. K., and Stegeman, A. (2008b). On the Non-Existence of Optimal Solutions and the Occurrence of Degeneracy in the CANDECOMP/PARAFAC Model. *Psychometrika*, 73:431–439.

Kruskal, J. (1977). Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra Appl.*, 18:95–138.

Landsberg, J. M. (2012). *Tensors: Geometry and Applications*, volume 128. American Mathematical Society, Providence, RI, USA.

Langville, A. N., Meyer, C. D., and Albright, R. (2006). Initializations for the nonnegative matrix factorization. In *Proc. of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Philadelphia, USA.

Lim, L.-H. and Comon, P. (2009). Nonnegative approximations of nonnegative tensors. *Journal of Chemometrics*, 23(7-8):432–441.

Lundy, M., Harshman, R., and Kruskal, J. (1989). A two-stage procedure incorporating good features of both trilinear and quadrilinear models. pages 123–130.

Mitchell, B. C. and Burdick, D. S. (1994). Slowly converging PARAFAC sequences: Swamps and two-factor degeneracies. *Journal of Chemometrics*, 8:155–168.

Navasca, C., Lathauwer, L. D., and Kindermann, S. (2008). Swamp reducing technique for tensor decomposition. In *Proc. 16th European Signal Processing Conference (EUSIPCO 2008)*.

References VII

Nion, D. and Sidiropoulos, N. (2009). Adaptive Algorithms to Track the PARAFAC Decomposition of a Third-Order Tensor. *IEEE Transactions on Signal Processing*, 57(6):2299–2310.

Paatero, P. (1997). A weighted non-negative least squares algorithm for three-way PARAFAC factor analysis. *Chemometrics Intelligent Laboratory Systems*, 38(2):223–242.

Paatero, P. (2000). Construction and analysis of degenerate PARAFAC models. *J. Chemometrics*, 14(3):285–299.

Phan, A.-H., Tichavský, P., and Cichocki, A. (2015). Tensor deflation for CANDECOMP/PARAFAC. Part 1: Alternating Subspace Update Algorithm. *IEEE Transaction on Signal Processing*, 63(12):5924–5938.

Phan, A.-H., Tichavský, P., and Cichocki, A. (2017). Partitioned hierarchical alternating least squares algorithm for CP tensor decomposition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2542–2546.

References VIII

Phan, A.-H., Tichavský, P., and Cichocki, A. (2019). Error preserving correction: A method for CP decomposition at a target error bound. *IEEE Transactions on Signal Processing*, 67(5):1175–1190.

Phan, A.-H., Yamagishi, M., and Cichocki, A. (2019). Quadratic programming over ellipsoids and its applications to linear regression and tensor decomposition. *Neural Computing and Applications*.

Rajih, M., Comon, P., and Harshman, R. A. (2008). Enhanced line search: A novel method to accelerate PARAFAC. *SIAM Journal of Matrix Analysis and Applications*, 30(3):1128–1147.

Rayens, W. and Mitchell, B. (1997). Two-factor degeneracies and a stabilization of PARAFAC. *Chemometrics Intelliggence Laboratory Systems*, 38(2):173–181.

References IX

Sidiropoulos, N., Bro, R., and Giannakis, G. (2000). Parallel factor analysis in sensor array processing. *IEEE Transactions on Signal Processing*, 48(8):2377–2388.

Sorber, L., Van Barel, M., and De Lathauwer, L. (2016). Tensorlab v3.0.

Stegeman, A. (2012). Candecomp/parafac: From diverging components to a decomposition in block terms. *SIAM J. Matrix Analysis Applications*, 33(2):291–316.

Stegeman, A. and Sidiropoulos, N. (2007). On Kruskal's uniqueness condition for the CANDECOMP/PARAFAC decomposition. *Linear Algebra and its Applications*, 420:540–552.

Strassen, V. (1969). Gaussian elimination is not optimal. *Numer. Math.*, 13(4):354–356.

References X

Tichavský, P., Phan, A.-H., and Cichocki, A. (2016). Partitioned alternating least squares technique for canonical polyadic tensor decomposition. *IEEE Signal Processing Letters*, 23(7):993–997.

Tichavský, P., Phan, A.-H., and Cichocki, A. (2019). Sensitivity in tensor decomposition. *IEEE Signal Processing Letters*, 26(11):1653–1657.

Yeredor, A. (2000). Blind source separation via the second characteristic function. *Signal Processing*, 80(5):897–902.