A Brief Introduction to Low-Rank Tensor Decompositions

Joseph Nakao [1]

[1]Department of Mathematical Sciences, University of Delaware

August 2022

Table of Contents

Key References

This lecture/talk is based on the following two references:

T.G. Kolda and B.W. Bader, Tensor Decompositions and Applications, *SIAM Review*, **51:3** (2009), pp. 455-500.

K. Kormann, Low-rank tensor discretization for high-dimensional problems, *Vorlesung*, **SS 2017** (2017).

Motivation

GOAL: to solve problems of high dimensionality with low storage and computational complexity.

ISSUE: the curse of dimensionality.

- Oftentimes, data for $d-$dimensional problem is stored as an order-d tensor, $\mathscr{X} \in \mathbb{R}^{N_1 \times \ldots \times N_d}$, $\mathscr{X}_{i_1,\ldots,i_d} \approx f(x_{i_1}, \ldots, x_{i_d})$.
- Computational cost and storage complexity quickly increase as d increases.

FIX: low rank tensor decompositions.

- tensor decompositions reduce storage complexity.

  e.g., SVD $\leftrightarrow$ $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$
- truncation for low-rank approximations of $\mathscr{X}$.

  e.g., truncated SVD

Example 1

$$f(x, y, z) = 1 + \epsilon \cos(x) \cos(y) \cos(z)$$

Store as $\mathscr{X} \in \mathbb{R}^{N_x \times N_y \times N_z}$, $\mathscr{X}_{i,j,k} = f(x_i, y_j, z_k)$.
Storage complexity is N^3. Not great!

$$f(x, y, z) = 1 \cdot 1 \cdot 1 + \epsilon \cos(x) \cdot \cos(y) \cdot \cos(z)$$

$\Updownarrow$ (Can think of as separation of variables)

$$\mathscr{X} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} + \begin{pmatrix} \epsilon \cos(x_1) \\ \vdots \\ \epsilon \cos(x_{N_x}) \end{pmatrix} \otimes \begin{pmatrix} \cos(y_1) \\ \vdots \\ \cos(y_{N_y}) \end{pmatrix} \otimes \begin{pmatrix} \cos(z_1) \\ \vdots \\ \cos(z_{N_z}) \end{pmatrix}$$

Rank-2 tensor (two basis elements).
Store as three $N \times 2$ frames/matrices; storage complexity $6N$.

What now?

ISSUE: in general, an analytic tensorization is not possible.

Q: how do we attain a (low-rank) tensor approximation?

Layout of the remainder of the lecture/talk:

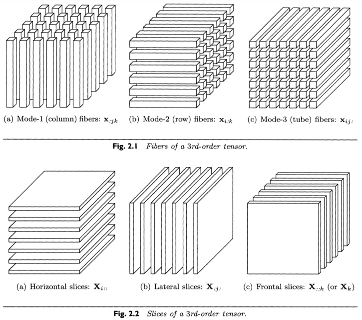
1. Lots of definitions and notation.
2. Three standard tensor decompositions.
3. Sprinkle in approximation issues, error estimates, and comparisons of all three tensor decompositions.

Definitions and notation

An order-d tensor is $\mathscr{X} \in \mathbb{R}^{N_1 \times \dots \times N_d}$ with entries $\mathscr{X}_{i_1,\dots,i_d}$.

A mode-n fiber fixes all but the n^{th}-index. Stored as column vectors.

A frame/slice fixes all but two indices. Stored as matrices.



(Kolda and Bader, pp. 458)

cont...

We call $\mathscr{X}$ a rank-one tensor if $\mathscr{X} = \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \ldots \circ \mathbf{a}^{(d)}$, for some vectors $\mathbf{a}^{(n)} \in \mathbb{R}^{N_n}$, $n = 1, \ldots, d$. Elementwise, $\mathscr{X}_{i_1,\ldots,i_d} = \prod_{n=1}^{d} a_{i_k}^{(n)}$.

Note: here $\circ$ denotes the outer product, but some authors use $\otimes$.

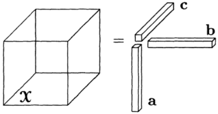


Fig. 2.3 *Rank-one third-order tensor,* $\mathfrak{X} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$. *The* (i,j,k) *element of* $\mathfrak{X}$ *is given by* $x_{ijk} = a_i b_j c_k$.

(Kolda and Bader, pp. 459)

The rank of a tensor, $\text{rank}(\mathscr{X})$, is the smallest number of rank-one tensors that generate $\mathscr{X}$ as their sum.

Vectorization and Matricization

Vectorization is the process of expressing a tensor as a vector.

$$\mathscr{X} \quad \longrightarrow \quad \mathrm{vec}(\mathscr{X})$$

IDEA: order the mode-n fibers. The specific ordering doesn't matter so long as you're consistent.

Matricization is the process of expressing a tensor as a matrix.
A mode-n matricization is

$$\mathscr{X} \quad \longrightarrow \quad \mathbf{X}_{(n)}$$

IDEA: order the mode-n fibers as the columns of a matrix. The specific ordering doesn't matter so long as you're consistent.

Note: there is a general matricization where the columns contain information from more than one dimension, $\mathbf{X}_{(\alpha)}$, where $\alpha \subseteq \{1, ..., d\}$.

Example 2.

Let $\mathscr{X} \in \mathbb{R}^{2\times3\times2}$ with frontal slices $\begin{bmatrix}1 & 3 & 5\\ 2 & 4 & 6\end{bmatrix}$ and $\begin{bmatrix}7 & 9 & 11\\ 8 & 10 & 12\end{bmatrix}$.

$$\mathbf{X}_{(1)} = \begin{bmatrix}1 & 3 & 5 & 7 & 9 & 11\\ 2 & 4 & 6 & 8 & 10 & 12\end{bmatrix} \in \mathbb{R}^{2\times6}$$

$$\mathbf{X}_{(2)} = \begin{bmatrix}1 & 2 & 7 & 8\\ 3 & 4 & 9 & 10\\ 5 & 6 & 11 & 12\end{bmatrix} \in \mathbb{R}^{3\times4}$$

$$\mathbf{X}_{(3)} = \begin{bmatrix}1 & 2 & 3 & 4 & 5 & 6\\ 7 & 8 & 9 & 10 & 11 & 12\end{bmatrix} \in \mathbb{R}^{2\times6}$$

$$\text{vec}(\mathscr{X}) = [1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9\ 10\ 11\ 12]^T \in \mathbb{R}^{12\times1}$$

Tensor multiplication

We only consider "multiplying" a tensor by a matrix (or vector).

The mode-n (matrix) product of $\mathscr{X} \in \mathbb{R}^{N_1 \times \ldots \times N_d}$ and $\mathbf{U} \in \mathbb{R}^{M \times N_n}$ is

$$\mathscr{X} \times_n \mathbf{U} \in \mathbb{R}^{N_1 \times \ldots \times N_{n-1} \times M \times N_{n+1} \times \ldots \times N_d},$$

with entries

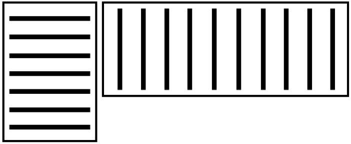
$$(\mathscr{X} \times_n \mathbf{U})_{i_1, \ldots, i_{n-1}, m, i_{n+1}, \ldots, i_d} \doteq \sum_{i_n=1}^{N_n} \mathscr{X}_{i_1, \ldots, i_d} U_{m, i_k}.$$

Think of this as taking the projection/inner product of $\mathscr{X}$ and $\mathbf{U}$ in the n^{th}-dimension.

cont...

Visualization:

$$\mathscr{Y} = \mathscr{X} \times_n \mathbf{U} \qquad \Leftrightarrow \qquad \mathbf{Y}_{(n)} = \mathbf{U}\mathbf{X}_{(n)}$$



Continuous interpretation:

Let $f(x_1, x_2, x_3) \leftrightarrow \mathscr{X}$ and $g(x_2, y) \leftrightarrow \mathbf{U}$, with $x_n \in I_n$ and $y \in I_y$.

$$\mathscr{X} \times_2 \mathbf{U} \quad \Leftrightarrow \quad \underbrace{\int_{I_2} f(x_1, x_2, x_3) g(x_2, y) dx_2}_{\langle f,g \rangle_2 \;=\; \text{function of } x_1,\, y,\, x_3}$$

Tensor products

The Kronecker product of matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{K \times L}$ is

$$\mathbf{A} \otimes \mathbf{B} \doteq \begin{pmatrix} A_{11}\mathbf{B} & \dots & A_{1J}\mathbf{B} \\ A_{21}\mathbf{B} & \dots & A_{2J}\mathbf{B} \\ \vdots & \ddots & \vdots \\ A_{I1}\mathbf{B} & \dots & A_{IJ}\mathbf{B} \end{pmatrix} \in \mathbb{R}^{IK \times JL}$$

Note: $\mathbf{A} \otimes \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1, \mathbf{a}_1 \otimes \mathbf{b}_2, \mathbf{a}_1 \otimes \mathbf{b}_3, \dots, \mathbf{a}_J \otimes \mathbf{b}_{L-1}, \mathbf{a}_J \otimes \mathbf{b}_L]$

The Hadamard product of matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{I \times J}$ is

$$\mathbf{A} * \mathbf{B} \doteq \begin{pmatrix} A_{11}B_{11} & \dots & A_{1J}B_{1J} \\ A_{21}B_{21} & \dots & A_{2J}B_{2J} \\ \vdots & \ddots & \vdots \\ A_{I1}B_{I1} & \dots & A_{IJ}B_{IJ} \end{pmatrix} \in \mathbb{R}^{I \times J}$$

A quick recap

- High-order tensors are expensive in storage and computation.
- We can flatten a tensor into a matrix (or vector).
- The idea of matricization will be important moving forward for these tensor decompositions.

Any questions?

Table of Contents

CANonical DECOMPosition / PARAllel FACtors

IDEA: express $\mathscr{X}$ as a sum of rank-one tensors; loosely similar to a SVD.

$$\mathscr{X} \approx \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \ldots \circ \mathbf{a}_r^{(d)}$$

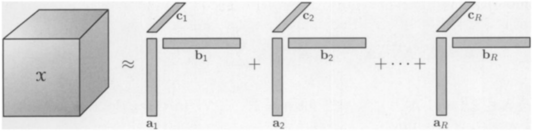


Fig. 3.1 *CP decomposition of a three-way array.*

(Kolda and Bader, pp. 463)

- Can store $\{\mathbf{a}_r^{(n)} : r = 1, \ldots, R\}$ as frames $\mathbf{A}^{(n)}$ for $n = 1, \ldots, d$.
- The R that gives equality is the rank($\mathscr{X}$).
- Storage complexity is RdN; much less than N^3 is naturally low rank.

About those low-rank tensor approximations...

With the CP format we can start discussing low-rank tensor approximations of $\mathscr{X}$.

Q: is it possible to find the best rank-k approximation to $\mathscr{X}$? (Ideally with $k < \text{rank}(\mathscr{X})$).
A: no.

Consider a singular value decomposition of a matrix, $\mathbf{A} = \sum_{r=1}^{R} \sigma_r \mathbf{u}_r \circ \mathbf{v}_r$.

The rank-k approximation that minimizes $\|\mathbf{A} - \mathbf{B}\|$ is $\mathbf{B} = \sum_{r=1}^{k} \sigma_r \mathbf{u}_r \circ \mathbf{v}_r$.

The higher-order tensor analogue is not true! (The issue is a bit more complicated).

Border rank

ISSUE: a best rank-k approximation might not exist.
REASON: degeneracy – $\mathscr{X}$ may be approximated arbitrarily well by a rank-k factorization.
Example 3. The space of rank-2 tensors is not closed.

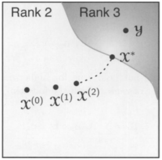


Fig. 3.2 *Illustration of a sequence of tensors converging to one of higher rank* [144].

Approximating a rank-3 tensor (Kolda and Bader, pp. 463)

FIX: Consider the border rank.

$$\widetilde{\text{rank}}(\mathscr{X}) \doteq \min\{k : \forall \epsilon > 0, \exists \mathscr{E} \text{ s.t. } \|\mathscr{E}\| < \epsilon, \text{rank}(\mathscr{X} + \mathscr{E}) = k\},$$

where $\mathscr{E} = -\mathscr{X} + \sum_{r=1}^{k} \lambda_r \mathbf{a}_r^{(1)} \circ \ldots \circ \mathbf{a}_r^{(d)}$.

Remarks and takeaways

- Rank degeneracy is not an uncommon occurrence.
- Rank degeneracy causes problems in practice. (See Kolda/Bader).
- The vectors $\mathbf{a}_r^{(n)}$ are stored in frames/matrices $\mathbf{A}^{(n)} \in \mathbb{R}^{N_n \times R_n}$.
- Alternating Least Squares (ALS) is the general workhorse algorithm for computing the CP format of a tensor.
- The CP format is straightforward despite this rank degeneracy. And, there have been substantial developments on variations of the CP format (e.g., enforce nonnegativity constraints).

Any questions?

Table of Contents

Tucker decomposition (a.k.a. higher-order SVD / HOSVD)

IDEA: a type of higher-order PCA in which $\mathscr{X}$ is decomposed into a core tensor $\mathscr{G}$ that is transformed by a matrix along each dimension.

$$\mathscr{X} \approx \mathscr{G} \times_{n=1}^{d} \mathbf{A}^{(n)}$$

$$\mathscr{X}_{i_1,\dots,i_d} \approx \sum_{j_1=1}^{J_1} \dots \sum_{j_d=1}^{J_d} \mathscr{G}_{j_1,\dots,j_d} A^{(1)}_{i_1,j_1} \dots A^{(d)}_{i_d,j_d}$$

As per above, the <u>multilinear rank</u> of the Tucker decomposition is $(J_1, \dots, J_d)$.

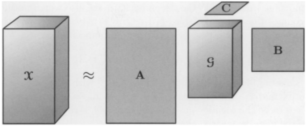


Fig. 4.1 *Tucker decomposition of a three-way array.*

$\mathscr{X} \approx \mathscr{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$ (Kolda and Bader, pp. 475)

Storage complexity is $J^d + NJd$.

Truncating the Tucker decomposition

Computing the HOSVD is based on the SVD of each $\mathbf{X}_{(n)}$, $n = 1, ..., d$.

$$\text{vec}(\mathscr{X}) \approx (\mathbf{A}^{(d)} \otimes ... \otimes \mathbf{A}^{(1)})\text{vec}(\mathscr{G})$$

$$\mathbf{X}_{(n)} = \mathbf{A}^{(n)}\mathbf{G}_{(n)} \left(\mathbf{A}^{(d)} \otimes ... \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes ... \otimes \mathbf{A}^{(1)}\right)^T$$

DESIRE: a HOSVD with multilinear rank $\mathbf{r} = (R_1, ..., R_d)$.

colrank$(\mathbf{X}_{(n)}) \leq J_n \Rightarrow R_n$ leading left singular vectors from the SVD of $\mathbf{X}_{(n)}$.

RESULT: a low-rank (in the multilinear rank sense) Tucker decomposition of $\mathscr{X}$, denoted by $\widetilde{\mathscr{X}}$.

Remarks and takeaways

$$\left\| \mathscr{X} - \widetilde{\mathscr{X}} \right\| \leq \sqrt{\sum_{j=1}^{d} \sum_{i=r_j+1}^{N_j} \left(\sigma_i^{(j)} \right)^2} \leq \sqrt{d} \ \text{min}\{\| \mathscr{X} - \mathscr{Y} \| : \mathscr{Y} \in \mathcal{T}_{\mathbf{r}}\},$$

where $\mathcal{T}_{\mathbf{r}}$ is the set of all Tucker decompositions of multilinear rank $\mathbf{r}$.

- The Tucker decomposition comes with closedness with the notion of multilinear rank $\mathbf{r}$.
- The truncated Tucker decomposition from the HOSVD-based algorithm is not optimal.
- The information held in the frames $\mathbf{A}^{(n)}$, $n = 1, ..., d$ saves storage.
- The storage requirement for $\mathscr{G}$ is not very attractive for large d.

Any questions?

Table of Contents

A starting visual

A dimension tree is a binary tree $\mathcal{T}$ with the following properties:

(i) each node α is a cluster of modes from $\{1, ..., d\}$.

(ii) the root node is $\{1, ..., d\}$.

(iii) each leaf node is a singleton set $\{n\}$.

(iv) each parent node α is the disjoint union of its two children nodes α_ℓ, α_r.

(assume α_ℓ indices are lower than α_r indices).

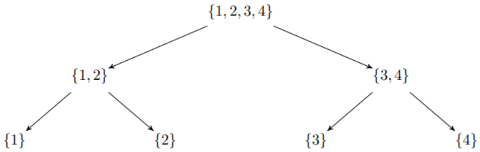
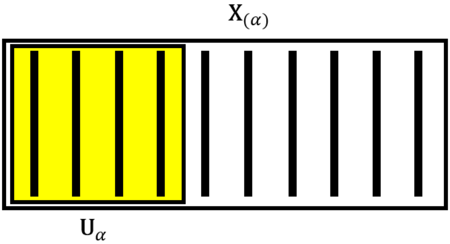


Figure 3: Dimension tree for four-way hierarchical Tucker tensor (balanced tree).

(Kormann, pp. 15)

The HT decomposition (HTD)

Let $\mathbf{U}_\alpha$ have r_α columns that form a basis for colspace$\left(\mathbf{X}_{(\alpha)}\right)$.



IDEA: each parent node is associated with the following decomposition:

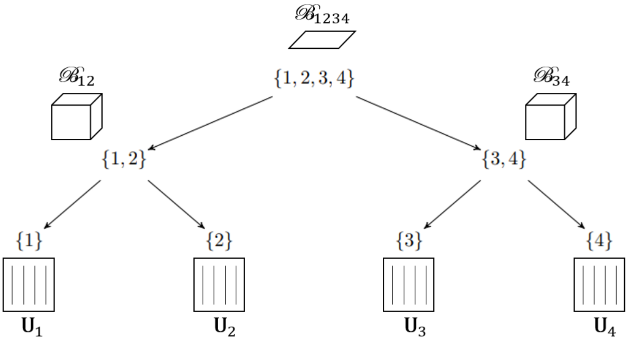
$$\mathbf{U}_\alpha = (\mathbf{U}_{\alpha_r} \otimes \mathbf{U}_{\alpha_\ell})\mathbf{B}_\alpha,$$

for some transfer matrix/tensor $\mathbf{B}_\alpha \in \mathbb{R}^{r_{\alpha_r} r_{\alpha_\ell} \times r_\alpha} \Leftrightarrow \mathscr{B}_\alpha \in \mathbb{R}^{r_{\alpha_\ell} \times r_{\alpha_r} \times r_\alpha}$.

Example 4

$$\mathbf{U}_\alpha = (\mathbf{U}_{\alpha_r} \otimes \mathbf{U}_{\alpha_\ell})\mathbf{B}_\alpha$$

Start by considering $\alpha = \{1, 2, 3, 4\}$, $\alpha_\ell = \{1, 2\}$, $\alpha_r = \{3, 4\}$.



Example 4 (cont...)

$$\begin{aligned}\text{vec}(\mathscr{X}) &= \mathbf{X}_{(\{1,2,3,4\})} = \mathbf{U}_{1234} \\ &= (\mathbf{U}_{34} \otimes \mathbf{U}_{12})\mathbf{B}_{1234} \\ &= \Big(\big((\mathbf{U}_4 \otimes \mathbf{U}_3)\mathbf{B}_{34}\big) \otimes \big((\mathbf{U}_2 \otimes \mathbf{U}_1)\mathbf{B}_{12}\big)\Big)\,\mathbf{B}_{1234} \\ &= (\mathbf{U}_4 \otimes \mathbf{U}_3 \otimes \mathbf{U}_2 \otimes \mathbf{U}_1)(\mathbf{B}_{34} \otimes \mathbf{B}_{12})\mathbf{B}_{1234}\end{aligned}$$

Storage complexity for Example 4 is $4Nr + 2r^3 + r^2$.

Storage complexity in general is $dNr + (d-2)r^3 + r^2$.

Takeaway: For larger d, storing the $d-1$ HT transfer tensors is much cheaper than storing the Tucker decomposition transfer tensor.

HTD (cont...)

Remarks:

- We typically desire the orthogonalization of a HTD of a tensor. (The columns of the mode frames $\mathbf{U}_n$ form an orthonormal basis for all the nodes except the root node.)

  i.e., $\mathbf{U}_n$ unitary for all $n = 1, ..., d$ implies $\mathbf{U}_\alpha$ unitary for all $\alpha \neq \{1, ..., d\}$.
- Complexity of orthogonalization algorithm is $\mathcal{O}(dNr^2 + dr^4)$.
- $\text{vec}(\mathscr{X}) = (\tilde{\mathbf{U}}_4 \otimes \tilde{\mathbf{U}}_3 \otimes \tilde{\mathbf{U}}_2 \otimes \tilde{\mathbf{U}}_1)(\tilde{\mathbf{B}}_{34} \otimes \tilde{\mathbf{B}}_{12})\tilde{\mathbf{B}}_{1234}$

Q: How can we compute a (low-rank) HTD?

A: Two approaches based on projections at each node.

The two approaches

IDEA: Define projections $\mathbf{W}_\alpha$ at each node α that map into a lower-dimensional subspace of colspace$\left(\mathbf{X}_{(\alpha)}\right)$.

Method 1 / Algorithm 6 (Kormann, pp. 20).
Input: a tensor $\mathscr{X}$ not in HTD and desired ranks $\{r_\alpha : \alpha \in \mathcal{T}\}$.
Output: a tensor $\widetilde{\mathscr{X}}$ in HTD with rank$\left(\widetilde{\mathbf{X}}_{(\alpha)}\right) \leq r_\alpha$ for all $\alpha \in \mathcal{T}$.

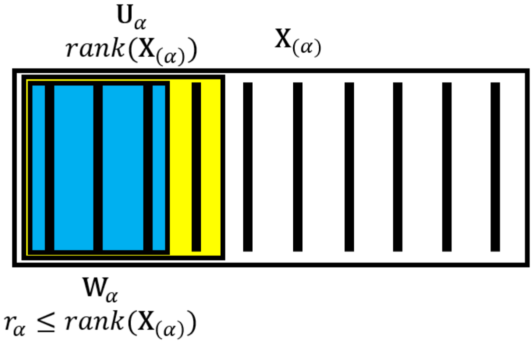
Method 2 / Algorithm 7 (Kormann, pp. 21).
Input: a tensor $\mathscr{X}$ in orthogonalized HTD and desired ranks $\{r_\alpha : \alpha \in \mathcal{T}\}$.
Output: a tensor $\widetilde{\mathscr{X}}$ in HTD with rank$\left(\widetilde{\mathbf{X}}_{(\alpha)}\right) \leq r_\alpha$ for all $\alpha \in \mathcal{T}$.

Projection $\mathbf{W}_\alpha$ at each node (same 4D example):

$$\text{vec}(\widetilde{\mathscr{X}}) = \left(\mathbf{W}_4\mathbf{W}_4^T \otimes \mathbf{W}_3\mathbf{W}_3^T \otimes \mathbf{W}_2\mathbf{W}_2^T \otimes \mathbf{W}_1\mathbf{W}_1^T\right) \left(\mathbf{W}_{34}\mathbf{W}_{34}^T \otimes \mathbf{W}_{12}\mathbf{W}_{12}^T\right) \text{vec}(\mathscr{X})$$

Method 1

IDEA: For each node $\alpha \in \mathcal{T}$, take r_α dominant left singular vectors of $\mathbf{X}_{(\alpha)}$.



$$\mathbf{X}_\alpha = \mathbf{U}_\alpha \mathbf{\Sigma}_\alpha \mathbf{V}_\alpha^T \qquad \text{(SVD)}$$

Method 1 (cont...)

$$\mathrm{vec}(\widetilde{\mathscr{X}}) = \left(\mathbf{W}_4\mathbf{W}_4^T \otimes \mathbf{W}_3\mathbf{W}_3^T \otimes \mathbf{W}_2\mathbf{W}_2^T \otimes \mathbf{W}_1\mathbf{W}_1^T\right) \left(\mathbf{W}_{34}\mathbf{W}_{34}^T \otimes \mathbf{W}_{12}\mathbf{W}_{12}^T\right)\mathrm{vec}(\mathscr{X})$$

$$= (\mathbf{W}_4 \otimes \mathbf{W}_3 \otimes \mathbf{W}_2 \otimes \mathbf{W}_1) \left(\underbrace{(\mathbf{W}_4^T \otimes \mathbf{W}_3^T)\mathbf{W}_{34}}_{\doteq \mathbf{B}_{34}} \otimes \underbrace{(\mathbf{W}_2^T \otimes \mathbf{W}_1^T)\mathbf{W}_{12}}_{\doteq \mathbf{B}_{12}}\right) \left(\underbrace{(\mathbf{W}_{34}^T \otimes \mathbf{W}_{12}^T)\mathrm{vec}(\mathscr{X})}_{\doteq \mathbf{B}_{1234}}\right)$$

$$\left\|\mathscr{X} - \widetilde{\mathscr{X}}\right\| \leq \sqrt{\sum_{\alpha \in \mathcal{T}'} \sum_{i=r_\alpha+1}^{n_\alpha} \left(\sigma_i^{(\alpha)}\right)^2} \leq \sqrt{2d-3}\,\min\{\|\mathscr{X} - \mathscr{Y}\| : \mathscr{Y} \in HT_{\mathbf{r}}\},$$

where $\mathcal{T}' = \mathcal{T} \backslash \{\alpha_{root}, \alpha_{root_\ell}\}$ and $HT_{\mathbf{r}}$ is the set of all HTDs of desired multilinear rank $\mathbf{r}$.

Method 2

IDEA: For each node $\alpha \in \mathcal{T}$, define $\mathbf{W}_\alpha$ using the so-called Gramians, $\mathbf{G}_\alpha$.

$$\mathbf{X}_{(\alpha)}\mathbf{X}_{(\alpha)}^T \doteq \mathbf{U}_\alpha \mathbf{G}_\alpha \mathbf{U}_\alpha^T = (\mathbf{U}_\alpha \mathbf{V}_\alpha)\,\mathbf{\Lambda}_\alpha\,(\mathbf{U}_\alpha \mathbf{V}_\alpha)^T$$

Define $\mathbf{W}_\alpha$ using the frames $\mathbf{U}_\alpha$ and leading $r_\alpha \leq \text{rank}\left(\mathbf{X}_{(\alpha)}\right)$ eigenvectors, $\mathbf{S}_\alpha$.

$$\mathbf{W}_\alpha \doteq \mathbf{U}_\alpha \mathbf{S}_\alpha$$

Note: $\mathbf{U}_\alpha$ are unitary by input assumption.

$$\text{vec}(\widetilde{\mathscr{X}}) = (\mathbf{U}_4\mathbf{S}_4 \otimes \mathbf{U}_3\mathbf{S}_3 \otimes \mathbf{U}_2\mathbf{S}_2 \otimes \mathbf{U}_1\mathbf{S}_1) \left(\underbrace{(\mathbf{S}_4^T \otimes \mathbf{S}_3^T)\mathbf{B}_{34}\mathbf{S}_{34}}_{\doteq \tilde{\mathbf{B}}_{34}} \otimes \underbrace{(\mathbf{S}_2^T \otimes \mathbf{S}_1^T)\mathbf{B}_{12}\mathbf{S}_{12}}_{\doteq \tilde{\mathbf{B}}_{12}} \right) \left(\underbrace{(\mathbf{S}_{34}^T \otimes \mathbf{S}_{12}^T)\mathbf{B}_{1234}}_{\doteq \tilde{\mathbf{B}}_{1234}} \right)$$

Summary

What did we cover?

- High-dimensional problems are expensive in storage and computation.
- Tensors can be flattened into matrices.
- CP format reduces storage complexity to RdN, but suffers from rank degeneracy. (But still has plenty of applications!)
- Multilinear rank $(J_1, ..., J_d)$; desire low-rank $(R_1, ..., R_d)$.
- Tucker decomposition reduces storage complexity to $J^d + NJd$. Still suffers from curse of dimensionality, just much less.
- Hierarchical Tucker decomposition reduces storage complexity to $dNr + (d-2)r^3 + r^2$. A solution to the curse of dimensionality for $d \geq 4$.
- Note: Tucker and HT are comparable for $d \leq 3$.

Thank you.