CSE 392: Matrix and Tensor Algorithms for Data

Instructor: Shashanka Ubaru

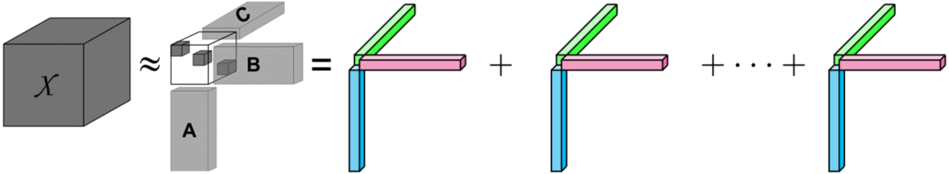
University of Texas, Austin

Spring 2024

Lecture 19: Tucker decomposition, HOSVD.

Outline

CP-Decomposition



- Find the best **tensor rank-r** fit:

$$\min_{\mathbf{a}_i,\mathbf{b}_i,\mathbf{c}_i} \|\mathcal{X} - \sum_{i=1}^{r} \sigma_i \cdot \mathbf{a}_i \circ \mathbf{b}_i \circ \mathbf{c}_i\|_F$$

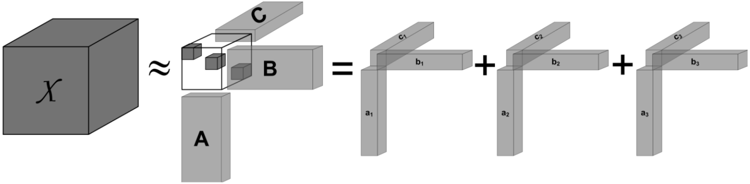
  - Extension of matrix rank
  - Interpretable
  - Summing r factors is sub-optimal
  - Determining rank is NP-hard

CP Decomposition - Existence and Ill-Posedness

- For a problem to be **well-posed**, the following conditions are required from its solution :
  - Existence
  - Uniqueness
  - Stability
- If either criterion is not satisfied, the problem is rendered **ill-posed** [1]
- Often, existence is taken for granted and an ill-posedness refers to either the lack of uniqueness or stability in the solution
- For CP, ill-posedness is more acute, as the **existence** of a solution is in question [2]
- The set of tensors of a given size that do not have a best rank-k approximation has **positive volume** (i.e., positive Lebesgue measure) for at least some values of k, which implies that **lack of best approximation** is rather common.

[1] Hadamard, Sur les problèmes aux dérivées partielles et leur signification physique. Princeton University Bulletin. 1902
[2] de Silva, Lim, Tensor rank and ill-posedness of the best low-rank approximation problem, 2008

CP - Uniqueness



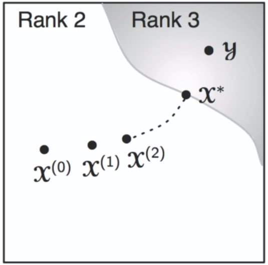
- $\mathcal{M} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ is essentially unique if

$$\mathrm{rank}_k(\boldsymbol{A}) + \mathrm{rank}_k(\boldsymbol{B}) + \mathrm{rank}_k(\boldsymbol{C}) \geq 2r + 2$$

- $\mathrm{rank}_k(\boldsymbol{A})$ = maximum value of k such that any k columns of $\boldsymbol{A}$ are linearly independent.
- Matrix factorization does not share this property! Usually need orthogonality constraint.

Inconsistencies with Tensor Rank

- Rank of real-valued tensor may be different over $\mathbb{R}$ or $\mathbb{C}$
- Determining rank of tensor is NP-hard
- Eckart-Young does not hold
- The best rank-k approximation may not exist

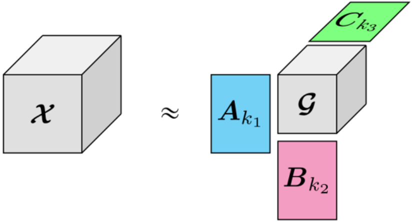


Best approximation is on the boundary of the space of rank-2 and rank-3 tensors. Since the space of rank-2 tensors is not closed, the sequence may converge to a tensor $\mathcal{X}^*$ of rank other than 2

Kruskal, Harshman, Lundy, How 3-MFA can cause degenerate PARAFAC solutions, among other relationships, in Multiway Data Analysis, Coppi, Bolasco, eds., North-Holland, Amsterdam, 1989

Kolda and Bader, Tensor decompositions and applications, SIAM, 2009

Tucker Decomposition[3]



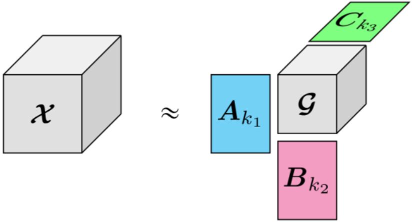
- Find the best **multi-linear rank-**(k_1, k_2 ,k_3) fit:

$$\min_{\mathbf{A}_{k_1},\mathbf{B}_{k_2},\mathbf{C}_{k_3}} \|\mathcal{X} - \mathcal{G} \times_1 \mathbf{A}_{k_1} \times_2 \mathbf{B}_{k_2} \times_3 \mathbf{C}_{k_3}\|_F$$

  - Higher-order PCA
  - Compressible
  - Truncation of full orth. sub-optimal
  - Hard to interpret

[3] Tucker, Problems in Measuring Change, 1963

Tucker Decomposition - notation



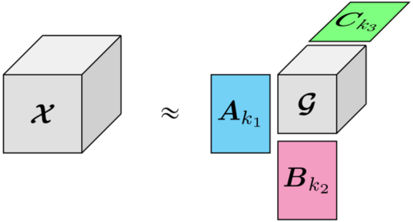
- The *Tucker decomposition* of a three-mode tensor $\mathcal{X} \in \mathbb{R}^{m \times n \times p}$ is given by:

$$\mathcal{X} \approx \mathcal{G} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C} =: [\![\mathcal{G}; \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!],$$

  where $\mathcal{G} \in \mathbb{R}^{k_1 \times k_2 \times k_3}$ is called the core tensor and $\boldsymbol{A} \in \mathbb{R}^{m \times k_1}$, $\boldsymbol{B} \in \mathbb{R}^{n \times k_2}$ and $\boldsymbol{C} \in \mathbb{R}^{p \times k_3}$ are factor matrices.
- Elementwise:

$$x_{ij\ell} \approx \sum_{q=1}^{k_1} \sum_{r=1}^{k_2} \sum_{s=1}^{k_3} g_{qrs} d_{iq} b_{jr} c_{\ell s} \text{ for } i \in [m], j \in [n], \ell \in [p]$$

Tucker Decomposition - matricized forms



- The matricized forms (one per mode) of *Tucker decomposition* are:

$$\mathcal{X}_{(1)} \approx \boldsymbol{A}\boldsymbol{G}_{(1)}(\boldsymbol{C} \otimes \boldsymbol{B})^\top,$$
$$\mathcal{X}_{(2)} \approx \boldsymbol{B}\boldsymbol{G}_{(2)}(\boldsymbol{C} \otimes \boldsymbol{A})^\top,$$
$$\mathcal{X}_{(3)} \approx \boldsymbol{C}\boldsymbol{G}_{(3)}(\boldsymbol{B} \otimes \boldsymbol{A})^\top$$

TUCKER-ALS algorithm

- Minimize the objective function:

$$F(\mathcal{G}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}) = \|\mathcal{X} - [\![\mathcal{G}; \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]\|_F^2$$

- The canonical TUCKER-ALS - repeatedly solve until convergence:
  - $\boldsymbol{A}_{t+1} = \arg\min_{\boldsymbol{A}} F\left(\mathcal{G}_t, \boldsymbol{A}, \boldsymbol{B}_t, \boldsymbol{C}_t\right) = \arg\min_{\boldsymbol{A}} \left\|\left(\boldsymbol{C}_t \otimes \boldsymbol{B}_t\right) \boldsymbol{G}_{(1),t}^\top \boldsymbol{A}^\top - \boldsymbol{X}_{(1)}^\top\right\|_F^2$
  - $\boldsymbol{B}_{t+1} = \arg\min_{\boldsymbol{B}} F\left(\mathcal{G}_t, \boldsymbol{A}_{t+1}, \boldsymbol{B}, \boldsymbol{C}_t\right) = \arg\min_{\boldsymbol{B}} \left\|\left(\boldsymbol{C}_t \otimes \boldsymbol{A}_{t+1}\right) \boldsymbol{G}_{(2),t}^\top \boldsymbol{B}^\top - \boldsymbol{X}_{(2)}^\top\right\|_F^2$
  - $\boldsymbol{C}_{t+1} = \arg\min_{\boldsymbol{C}} F\left(\mathcal{G}_t, \boldsymbol{A}_{t+1}, \boldsymbol{B}_{t+1}, \boldsymbol{C}\right) = \arg\min_{\boldsymbol{C}} \left\|\left(\boldsymbol{B}_{t+1} \otimes \boldsymbol{A}_{t+1}\right) \boldsymbol{G}_{(3),t}^\top \boldsymbol{C}^\top - \boldsymbol{X}_{(3)}^\top\right\|_F^2$
  - $\mathcal{G}_{t+1} = \arg\min_{\mathcal{G}} \left\|(\boldsymbol{C}_{t+1} \otimes \boldsymbol{B}_{t+1} \otimes \boldsymbol{A}_{t+1}) \boldsymbol{g}_{(:)} - \boldsymbol{x}_{(:)}\right\|_2^2$

Tucker Decompositions - Non-Uniqueness

- Consider the three-way Tucker decomposition of $\mathcal{X}$, also denoted $[\![\mathcal{G}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$
- Let $\mathbf{U} \in \mathbb{R}^{k_1 \times k_1}$, $\mathbf{V} \in \mathbb{R}^{k_2 \times k_2}$, and $\mathbf{W} \in \mathbb{R}^{k_3 \times k_3}$ be non-singular. Then

$$[\![\mathcal{G}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!] = [\![\widetilde{\mathcal{G}}; \mathbf{A}\mathbf{U}^{-1}, \mathbf{B}\mathbf{V}^{-1}, \mathbf{C}\mathbf{W}^{-1}]\!]$$

  where $\widetilde{\mathcal{G}} := \mathcal{G} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}$
- The core $\mathcal{G}$ can be modified without affecting the overall fit as long as an **inverse modification** is applied to the factor matrices
- Offers freedom to choose transformations that **simplify** the **core structure** in some way so that most of the elements of $\mathcal{G}$ are zero.

Towards the HOSVD

Recall: Let $\mathbf{A}$ be an $m \times n$ real-valued matrix, then $\mathbf{A}$ has a singular value decomposition:

$$\boldsymbol{A} = \boldsymbol{U}\Sigma\boldsymbol{V}^{\top},$$

where $\mathbf{U}$ is $m \times m$ orthogonal, $\mathbf{V}$ is $n \times n$ orthogonal, and Σ is $m \times n$ diagonal with diagonal elements the singular values $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_r > 0$

The matrix $\boldsymbol{U}$ contains the left singular vectors

HOSVD

Use left singular vectors of the SVDs of the matricizations (assuming ranks r_1, r_2, r_3):

- Compute $\boldsymbol{U}^{(1)}$ from SVD of $\boldsymbol{A}_{(1)}$, keep first r_1 cols
- Compute $\boldsymbol{U}^{(2)}$ from SVD of $\boldsymbol{A}_{(2)}$, keep first r_2 cols.
- Compute $\boldsymbol{U}^{(3)}$ from SVD of $\boldsymbol{A}_{(3)}$, keep first r_3 cols.
- $\mathcal{G} := \mathcal{A} \times_1 (\boldsymbol{U}^{(1)})^\top \times_2 (\boldsymbol{U}^{(2)})^\top \times_3 (\boldsymbol{U}^{(3)})^\top$ which means, e.g.,

$$\mathcal{G}_{(1)} = (\boldsymbol{U}^{(1)})^\top \mathcal{A}_{(1)} (\boldsymbol{U}^{(3)} \otimes \boldsymbol{U}^{(2)})$$

Now $\mathcal{G}$ is $r_1 \times r_2 \times r_3$ and this is an EXACT representation:

$$\mathcal{A} = \mathcal{G} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times_3 \boldsymbol{U}^{(3)}.$$

Three SVDs, independent of one another

Another notation $\mathcal{A} = [\![\mathcal{G}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)}]\!]$

HOSVD Algorithm

Inputs: Tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, ranks $\{r_1, \ldots, r_d\} \in \mathbb{N}$.

1. **for** $\ell = 1, \ldots, d$ **do**
2. $\quad \boldsymbol{U}^{(\ell)} \leftarrow r_\ell$ leading left singular vectors of $\boldsymbol{A}_{(\ell)}$
3. **end for**
4. $\mathcal{G} = \mathcal{A} \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} \cdots \times_d \boldsymbol{U}^{(d)\top}$
5. **return** $\mathcal{G}, \boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \cdots, \boldsymbol{U}^{(d)}$

HOOI Algorithm

Inputs: Tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$, ranks $\{r_1, \ldots, r_d\} \in \mathbb{N}$.

1. Initialize $\boldsymbol{U}^{(\ell)} \in \mathbb{R}^{n_\ell \times r_\ell}$ for all $\ell \in [d]$
2. **repeat**
3.   **for** $\ell = 1, \ldots, d$ **do**
4. $\mathcal{Y} = \mathcal{A} \times_1 \boldsymbol{U}^{(1)\top} \cdots \times_{\ell-1} \boldsymbol{U}^{(\ell-1)\top} \times_{\ell+1} \boldsymbol{U}^{(\ell+1)\top} \cdots \times_d \boldsymbol{U}^{(d)\top}$
5. $\boldsymbol{U}^{(\ell)} \leftarrow r_\ell$ leading left singular vectors of $\boldsymbol{Y}_{(\ell)}$
6.   **end for**
7. **until** fit ceases to improve or maximum iterations exhausted
8. $\mathcal{G} = \mathcal{A} \times_1 \boldsymbol{U}^{(1)\top} \times_2 \boldsymbol{U}^{(2)\top} \cdots \times_d \boldsymbol{U}^{(d)\top}$
9. **return** $\mathcal{G}, \boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \cdots, \boldsymbol{U}^{(d)}$

Truncated HOSVD

Use left singular vectors of the SVDs of the matricizations:

- Compute $\boldsymbol{U}^{(1)}$ from SVD of $\mathcal{A}_{(1)}$, truncate to $k_1 \leq r_1$ cols.
- Compute $\boldsymbol{U}^{(2)}$ from SVD of $\mathcal{A}_{(2)}$, truncate to $k_2 \leq r_2$ cols.
- Compute $\boldsymbol{U}^{(3)}$ from SVD of $\mathcal{A}_{(3)}$, truncate to $k_3 \leq r_3$ cols.
- $\mathcal{C} := \mathcal{A} \times_1 (\boldsymbol{U}^{(1)})^\top \times_2 (\boldsymbol{U}^{(2)})^\top \times_3 (\boldsymbol{U}^{(3)})^\top$ which means, e.g.,

$$\mathcal{C}_{(1)} = (\boldsymbol{U}^{(1)})^\top \mathcal{A}_{(1)} (\boldsymbol{U}^{(3)} \otimes \boldsymbol{U}^{(2)})$$

so $\mathcal{A} \approx \widehat{\mathcal{A}} := \mathcal{C} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times_3 \boldsymbol{U}^{(3)}$

where $\mathcal{C}$ is now $k_1 \times k_2 \times k_3$

Truncating $\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \boldsymbol{U}^{(3)}$ to k_1, k_2, k_3 columns, resp, is not optimal, but can give a compressed representation that is “reasonable”.

Worst Case Error Bound

Theorem (Vannieuwenhoven et al, 2012)

Let $\widehat{\mathcal{A}} = [\![\mathcal{C}; \boldsymbol{U}^{(1)}, \ldots, \boldsymbol{U}^{(d)}]\!]$ *where* $\boldsymbol{U}^{(i)}$ *was truncated to* k_i *columns (i.e. the rank-*$(k_1, k_2, \ldots, k_d)$ *approximation to the dth order tensor), then*

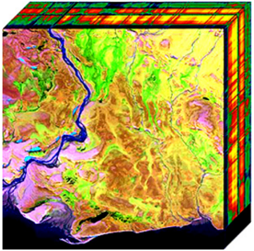
$$\|\mathcal{A} - \widehat{\mathcal{A}}\|_F^2 \le \sum_{j=1}^{d} \|\mathcal{A} \times_j (\mathbf{I} - \boldsymbol{U}^{(j)}(\boldsymbol{U}^{(j)})^\top)\|_F^2 = \sum_{j=1}^{d} \sum_{k_j+1}^{n_j} \sigma_i^2(\mathcal{A}_{(j)}).$$

That is, the squared approximation error is bounded by the **sum of the approximation errors on each mode unfolding**.

tr-HOSVD Illustration

A-priori selection of the truncation bounds is difficult - cannot afford time/space to compute the full and then use the error to truncate.

As an example, consider hyperspectral image data - 2 spatial dimensions, and wavelength. For each spatial location, the wavelength 'signature' tells the composition.



commons.wikimedia.org/wiki/File:HyperspectralCube.jpg, NASA, 2007.

tr-HOSVD Example: Hyperspectral Imaging

191 flyover images of the Washington DC mall. Downsampled images to 320×307.
HOSVD is orientation independent. Chose tensor as $320 \times 307 \times 191$.

D. Landgrebe and L. Biehl, An introduction and reference for multispec., March 2019.

tr-HOSVD Example

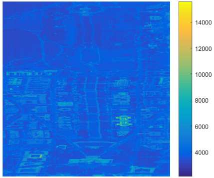
In the absence of any other information, arbitrarily chose to reduce each dimension by about 80% (i.e. core is $64 \times 62 \times 39$).

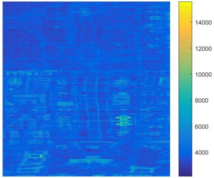
$$\frac{\|\mathcal{A} - \widehat{\mathcal{A}}\|_F}{\|\mathcal{A}\|_F} = .18$$

Exercise: What percent of the original storage is required by the new (truncated) one ?

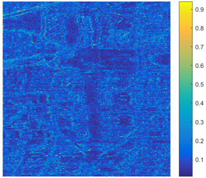
tr-HOSVD Example

Difference in one wavelength:





Angles between spectral signatures at each of the 320 x 307 spatial positions.



Variations on tr-HOSVD

Computing individual/independent full (or partial) SVDs can be costly. What if we give up the independence of the actions, and project as we go?

Sequential Truncated HOSVD (ST-HOSVD)

- Choose an ordering in which to visit the modes
- Once left singular vectors for a mode are computed, immediately project. Then only operate on the projected core result
- Example (ordering 1,2,3 and truncation (k_1, k_2, k_3)):
  - Compute $\boldsymbol{U}^{(1)}$ from SVD of $\mathcal{A}_{(1)}$
  - Compute $\boldsymbol{U}^{(2)}$ from SVD of $\widehat{\mathcal{C}} := \mathcal{A} \times_1 (\boldsymbol{U}^{(1)})^\top$
  - Compute $\boldsymbol{U}^{(3)}$ from SVD of $\tilde{\mathcal{C}} := \widehat{\mathcal{C}} \times_2 (\boldsymbol{U}^{(2)})^\top$
  - $\mathcal{C} = \tilde{\mathcal{C}} \times_3 (\boldsymbol{U}^{(3)})^\top$
- Now let $\mathcal{A} \approx [\mathcal{C}; \mathbf{U}^{(1)}, \mathbf{U}^{(2)}, \mathbf{U}^{(3)}]$. Worst case error bound is same as for tr-HOSVD.
- Computing on successively smaller objects, more efficient; often near-comparable, or better, behavior than tr-HOSVD!

Best Approximation?

- Let $\mathcal{S} = \{\mathcal{Y} \in \mathbb{R}^{n_1 \times \cdots n_d} | \mathcal{Y}_{(j)}$ has rank $r_j \leq n_j\}$
- Define $\mathcal{A}_{opt} := \arg\min_{\mathcal{Y} \in \mathcal{S}} \|\mathcal{A} - \mathcal{Y}\|_F$
- Existence of $\mathcal{A}_{opt}$ is guaranteed[4] but not unique since Tucker representations are not unique (see previous slides)
- Generally, computing $\mathcal{A}_{opt}$ requires solving an optimization problem via iteration
- High Order Orthogonal Iteration (HOOI) attempts to find it, iterates by cycling, but expensive
- HOOI offer quasi-optimality[4]

$$\|\mathcal{A} - \widehat{\mathcal{A}}\|_F \leq \sqrt{d}\|\mathcal{A} - \mathcal{A}_{opt}\|_F$$

[4] Hackbusch, 2012

Hierarchical Tucker

Storage for truncated HOSVD on an $m \times n \times p$ tensor $\mathcal{A}$:

- The $m \times k_1$, $n \times k_2$ and $p \times k_3$ factor matrices
- The $k_1 \times k_2 \times k_3$ core tensor.

If we repeat the factorization/truncation process on the core tensor, we get a hierarchical Tucker approach.

Matlab Demo