Tensor Tutorial

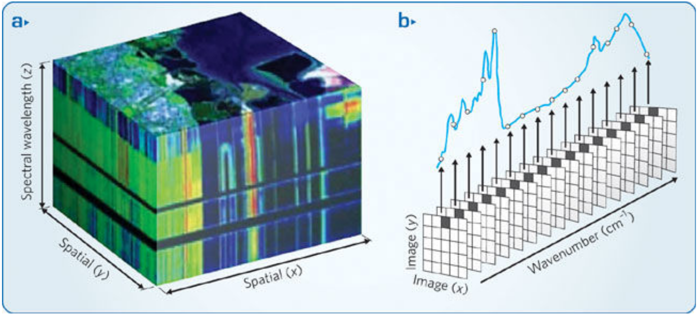
Misha Kilmer
Department of Mathematics
Tufts University

Research Thanks: NSF 0914957, NSF 1319653, NSF 1821148
IBM JSA

Motivation

Real-world data naturally multidim., w/ different characteristics:

Hyperspectral images (classification)[1]

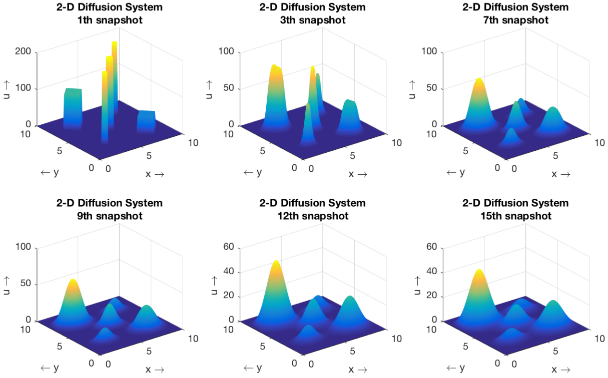


[1] Bannon,"Hyperspectral imaging: Cubes and Slices," Nature Photonics, 2009.

Motivation

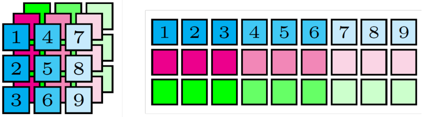
Real-world data naturally multidim., w/ different characteristics:

Discrete solutions, $\mathbf{u}(x_j, y_i, t_k)$ to PDEs[1]



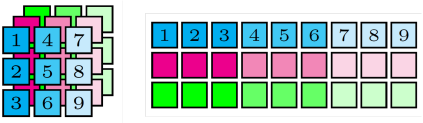
[1] Jiani Zhang, Tufts Mathematics Ph.D. Thesis, "Design and Application of Tensor Decompositions to Problems in Model and Image Compression and Analysis," 2017.

Motivation

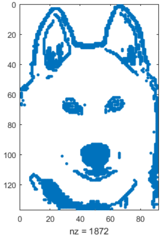


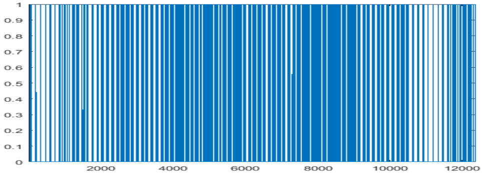
Traditional algorithms for compressing, analyzing, clustering data done by 'unfolding' this data into a <u>matrix</u>, or 2D array, and employing matrix algebra tools.

Motivation



Traditional algorithms for compressing, analyzing, clustering data done by ‘unfolding’ this data into a <u>matrix</u>, or 2D array, and employing matrix algebra tools.





Motivation

CLAIM: Traditional matrix-based methods for dim reduction, classification, training, based on vectorizing data generally do not make the most of possible high dimensional correlations/structure for compression and analysis.

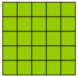
There is much to be gained by designing mathematical and computational techniques for the data in its natural form.

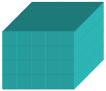
Review current mathematical definitions, constructs, theory, algorithms, for multiway data compression + applications.

Tensors: Definition

$$\mathcal{X} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_j} \quad \leftarrow \quad j\text{-th order tensor}$$

1st-order tensor: 

2nd-order tensor: 

3rd-order tensor: 

4th-order tensor: 

Notation

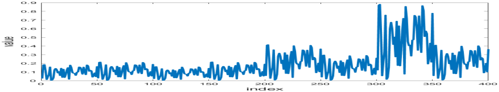
Uppercase Script: $\mathcal{A}$, is a 3rd order tensor.

Uppercase Bold: $\mathbf{X}$, is a matrix.

Bold lowercase: $\mathbf{y}$, is a vector OR a $1 \times 1 \times n$ tensor.

Data Organization Reveals Latent Structure

Suppose $\mathbf{y} \in \mathbb{R}^{mn}$



Reshape as $m \times n$ matrix,

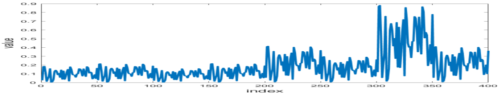
$$\mathbf{Y} = \mathbf{u}\mathbf{v}^\top = \mathbf{u} \circ \mathbf{v}$$

$$\Rightarrow \mathbf{y} = \mathbf{v} \otimes \mathbf{u} = \begin{bmatrix} v_1\mathbf{u} \\ v_2\mathbf{u} \\ \vdots \\ v_n\mathbf{u} \end{bmatrix}$$

Implies storage is reduced from mn to $m + n$ numbers.

Data Organization Reveals Latent Structure

Suppose $\mathbf{y} \in \mathbb{R}^{mn}$



Reshape as $m \times n$ matrix,

$$\mathbf{Y} = \mathbf{u}\mathbf{v}^\top = \mathbf{u} \circ \mathbf{v}$$

$$\Rightarrow \mathbf{y} = \mathbf{v} \otimes \mathbf{u} = \begin{bmatrix} v_1\mathbf{u} \\ v_2\mathbf{u} \\ \vdots \\ v_n\mathbf{u} \end{bmatrix}$$

Implies storage is reduced from mn to $m+n$ numbers.

Moving to higher dimensions reveals compressible structure.

Goals

- Uncover hidden patterns in data by computing an appropriate tensor decomposition/approximation?
- Use this to compress or constrain data in applications.
- Patterns are application dependent, the type of tensor decomposition should respect this.
- Consider tensor decompositions that are synonymous with 'factorization' in a matrix-mimetic sense vs. those that are not.

Reference, Toolbox

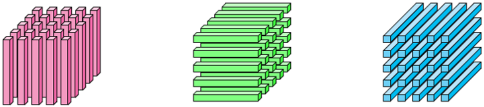
Required reading for my students: Kolda and Bader, "Tensor Decompositions and Applications," SIAM Review, Vol. 51, 2009.

MATLAB Tensor Toolbox Version 3.1, Available online, June 2019. URL: `https://gitlab.com/tensors/tensor_toolbox`

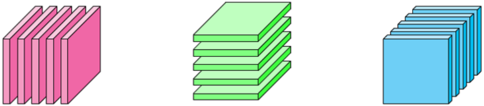
There are other free toolboxes as well that use slightly different constructs.

Notation - The Basics[2]

- Modes: the different dimensions
- Fibers: hold all indicies fixed except 1
- Slices: hold all indicies fixed except 2



(a) Mode-1 fibers. (b) Mode-2 fibers. (c) Mode-3 fibers.



(d) Lateral slices. (e) Horizontal slices. (f) Frontal slices.

[2] graphics: Elizabeth Newman, "A Step in the Right Dimension," Tufts Ph.D. Thesis, 2019

Norms

Norm is extension of Frobenius norm:

$$\|\mathcal{A}\| = \sqrt{\sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_N=1}^{I_N} a^2_{i_1,\ldots,\imath_N}}.$$

If $\mathfrak{X}, \mathfrak{Y}$ of same dimension, can take an inner-product (collapsing along dimensions) to a scalar:

$$< \mathfrak{X}, \mathfrak{Y} >= \sum_{i_1=1}^{I_1}\sum_{i_2=1}^{I_2}\cdots\sum_{i_N=1}^{I_N} x_{i_1,\ldots,\imath_N} y_{i_1,\ldots,i_N}.$$

Matricization[3]

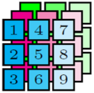
A tensor "matricization" refers to (specific) mappings of the tensor to a matrix. The nth mode unfolding maps $\boldsymbol{\mathcal{A}}$ to $\mathbf{A}$ via $(i_1, \ldots, i_N) \to (i_n, j)$, and

$$j = 1 + \sum_{k=1, k\neq n}^{N} (i_k - 1) \left(\prod_{m=1, m\neq n}^{k-1} I_m \right).$$

A graphical illustration is illuminating:

[3]graphics: Elizabeth Newman, Tufts Mathematics Ph.D. Thesis, "A Step in the Right Dimension," 2019

Matricization[3]



(a) Original $\mathcal{A}$.

(b) Mode-1 unfolding $\mathcal{A}_{(1)}$.

(c) Mode-2 unfolding $\mathcal{A}_{(2)}$.

(d) Mode-3 unfolding $\mathcal{A}_{(3)}$.

---

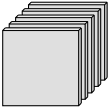
[3] graphics: Elizabeth Newman, Tufts Mathematics Ph.D. Thesis, "A Step in the Right Dimension," 2019

Tensor-Matrix products

$$\mathcal{C} = \mathcal{A} \times_n \mathbf{X} \longleftrightarrow \mathbf{C}_{(n)} = \mathbf{X} \cdot \mathbf{A}_{(n)}$$

Note that

$$\mathcal{A} \times_m \mathbf{X} \times_n \mathbf{Y} = \mathcal{A} \times_n \mathbf{Y} \times_m \mathbf{X}.$$



Frontal slice $\mathcal{A}_{:,:,k}$

Example: $\widetilde{\mathcal{A}} := \mathcal{A} \times_1 \mathbf{X} \times_2 \mathbf{Y} \Rightarrow \widetilde{\mathcal{A}}_{:,:,i} = \mathbf{X}\mathcal{A}_{:,:,i}\mathbf{Y}^\top$

Step Back to the Matrix SVD

Traditional workhorse, dim reduction/feature extraction: matrix SVD

- PCA - directions of most variability; projections in 'dominant' directions allows for dim reduction/relative comparison
- Compression (reduce near redundancies) via truncated SVD expansion is optimal (Eckart-Young Theorem)

$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^\top = \sum_{i=1}^{r} \sigma_i(\mathbf{u}^{(i)} \circ \mathbf{v}^{(i)})$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$

$$\mathbf{B} = \sum_{i=1}^{p} \sigma_i(\mathbf{u}^{(i)} \circ \mathbf{v}^{(i)}) \qquad \text{solves}$$

$$\min \|\mathbf{A} - \mathbf{B}\|_F \qquad \text{s.t. } \mathbf{B} \text{ has rank } p \leq r$$

Step Back to the Matrix SVD

Traditional workhorse, dim reduction/feature extraction: matrix SVD

- PCA - directions of most variability; projections in 'dominant' directions allows for dim reduction/relative comparison
- Compression (reduce near redundancies) via truncated SVD expansion is optimal (Eckart-Young Theorem)

$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^\top = \sum_{i=1}^{r} \sigma_i(\mathbf{u}^{(i)} \circ \mathbf{v}^{(i)})$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$

$$\mathbf{B} = \sum_{i=1}^{p} \sigma_i(\mathbf{u}^{(i)} \circ \mathbf{v}^{(i)}) \qquad \text{solves}$$

$$\min \|\mathbf{A} - \mathbf{B}\|_F \qquad \text{s.t. } \mathbf{B} \text{ has rank } p \leq r$$

Implicit storage: for an $m \times n$, $p(n+m)$ numbers stored, vs mn.

Step Back to the Matrix SVD

Traditional workhorse, dim reduction/feature extraction: matrix SVD

- PCA - directions of most variability; projections in 'dominant' directions allows for dim reduction/relative comparison
- Compression (reduce near redundancies) via truncated SVD expansion is optimal (Eckart-Young Theorem)

$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^\top = \sum_{i=1}^{r} \sigma_i (\mathbf{u}^{(i)} \circ \mathbf{v}^{(i)})$, $\sigma_1 \geq \sigma_2 \geq \cdots \geq 0$

$$\mathbf{B} = \sum_{i=1}^{p} \sigma_i (\mathbf{u}^{(i)} \circ \mathbf{v}^{(i)}) \qquad \text{solves}$$

$$\min \|\mathbf{A} - \mathbf{B}\|_F \qquad \text{s.t. } \mathbf{B} \text{ has rank } p \leq r$$

Question: What's the right high-dimensional analogue? (history, see Kolda & Bader)

Rank-1 Tensor

Idea 1 (Hitchcock, 1927): Like SVD, try to decompose as a sum of rank-1 tensors.

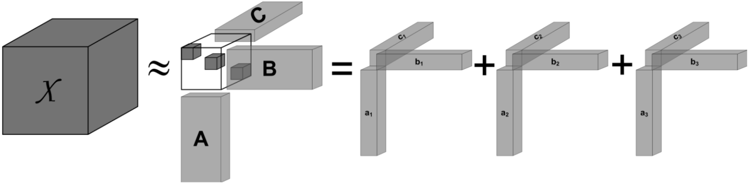
$$\mathcal{X} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c} \Rightarrow \mathcal{X}_{\ell,j,k} = a_\ell b_j c_k$$

Note that $\text{vec}(\mathcal{X}) = \mathbf{c} \otimes \mathbf{b} \otimes \mathbf{a}$.

Thus, some papers use Kronecker in place of outer-product notation.

Tensor Decompositions - CP

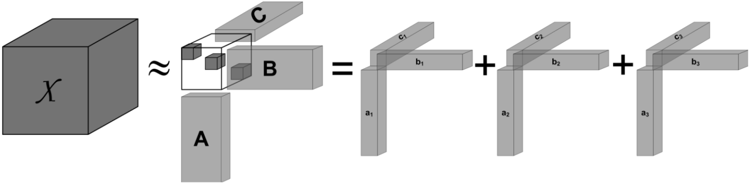
- **CP (CANDECOMP-PARAFAC) Decomposition** :



$$\mathcal{X} \approx \sum_{i=1}^{r} \mathbf{a}^{(i)} \circ \mathbf{b}^{(i)} \circ \mathbf{c}^{(i)}$$

- If equality & r minimal, then r is called the *rank* of the tensor
- Not generally orthogonal
- Not based on a 'product based factorization'
- Finding the rank is NP hard!
- No perfect procedure for fitting CP model to k terms

Kruskal Notation



$$\mathcal{X} \approx \sum_{i=1}^{r} \mathbf{a}^{(i)} \circ \mathbf{b}^{(i)} \circ \mathbf{c}^{(i)}$$

Kruskal notation: $[\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ or, if unit-normalized $[\![\lambda; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$.

Demo - Chemical Mixing

- Bro, R, Multi-way Analysis in the Food Industry. Models, Algorithms, and Applications. 1998. Ph.D. Thesis, Univ. of Amsterdam (NL) & Royal Veterinary and Agricultural University (DK). (see `http://www.models.kvl.dk/amino_acid_fluo`)
- 5, simple lab-made samples.
- Each sample: vary amts. tyrosine, tryptophan and phenylalanine dissolved in phosphate buffered water.
- Samples measured by fluorescence (excitation 250-300 nm, emission 250-450 nm, 1 nm intervals)
- $51 \times 201 \times 5$ tensor
- Brett W. Bader, Tamara G. Kolda and others. MATLAB Tensor Toolbox Version 3.1, Available online, June 2019. URL: `https://gitlab.com/tensors/tensor_toolbox`
- Matlab script: Thanks, T. Kolda, July 2019

Math Interpretation

Each of the three chemicals has fluorescence signature described as $\mathbf{u}^{(i)} \circ \mathbf{v}^{(i)}$, $i = 1, 2, 3$.

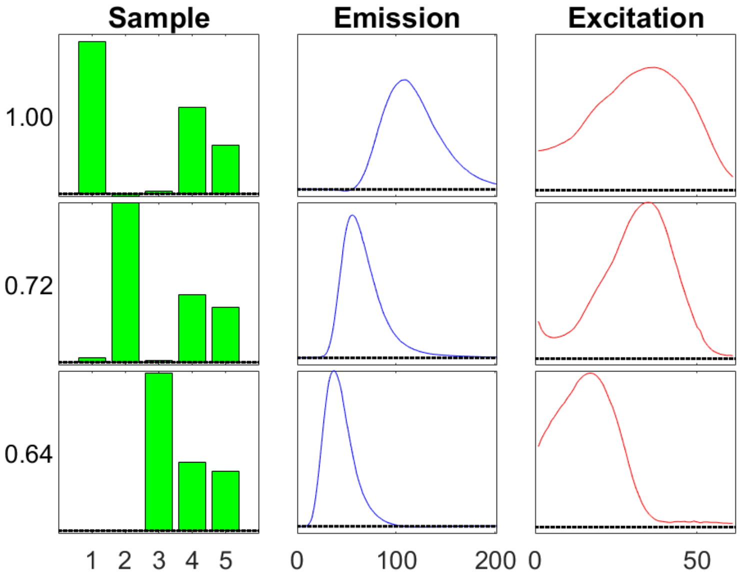
The jth sample is $w_1^{(j)} \mathbf{u}^{(1)} \circ \mathbf{v}^{(1)} + w_2^{(j)} \mathbf{u}^{(2)} \circ \mathbf{v}^{(2)} + w_3^{(j)} \mathbf{u}^{(3)} \circ \mathbf{v}^{(3)}$.

Then, if the samples are the frontal slices, we ideally should have

$$\mathcal{A} = \sum_{i=1}^{3} \mathbf{u}^{(i)} \circ \mathbf{v}^{(i)} \circ \mathbf{w}^{(i)}$$

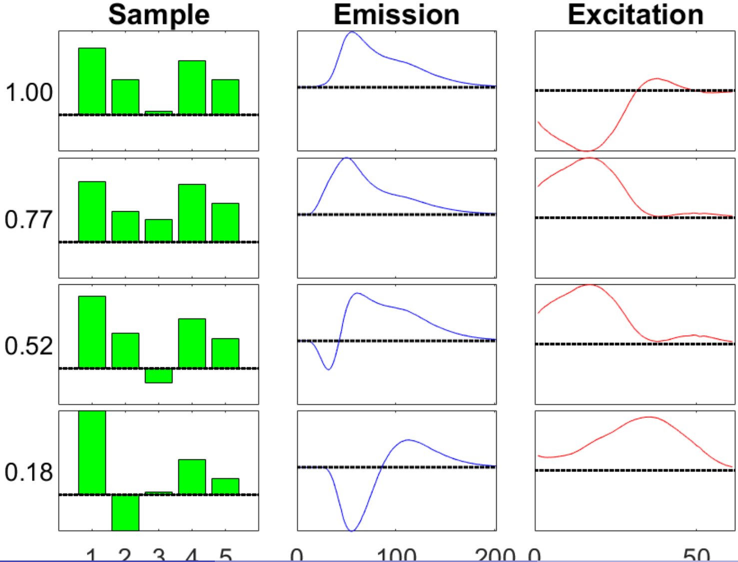
Independent of orientation...

Some Results



CP Example

Importance of fitting right number of terms; starting guesses.



Other Decompositions

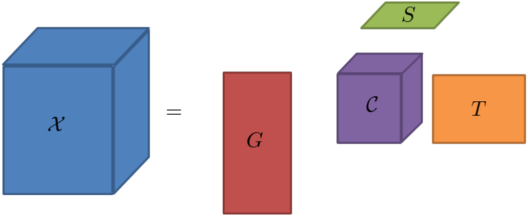
Other decompositions in the literature:

- Tucker (and HOSVD)
- Tensor Train (TT), hierarchical TT (ex: Tensor-Train Decomposition, Ivan Oseledets, SISC, 2011)
- Matrix-mimetic decompositions based on tensor-tensor products (K. & Martin 2011; Kernfeld, K., Aeron 2015) and corresponding algebraic framework.
  - Highly parallelizable
  - Amenable to orientation dependent data
  - Robust (e.g. to overfitting)

Each has advantages/disadvantages. The choice of decomposition should be application dependent!

Truncated Tucker/HOSVD

- **Tucker-3 Decomposition** :



$$\mathfrak{X} \approx \mathfrak{C} \times_1 \mathbf{G} \times_2 \mathbf{T} \times_3 \mathbf{S} = \sum_{i=1}^{r_1} \sum_{j=1}^{r_2} \sum_{k=1}^{r_3} c_{ijk} (\mathbf{g}^{(i)} \circ \mathbf{t}^{(j)} \circ \mathbf{s}^{(k)})$$

- $\mathfrak{C}$ is the *core* tensor, not generally diagonal or non-neg.
- $\mathbf{G}$, $\mathbf{T}$, $\mathbf{S}$ w/ orthonormal cols = HOSVD (De Lathauwer, et. al)
- Specify 3 ranks (r_1, r_2, r_3); truncation not-optimal

HOSVD, ST-HOSVD

Computing the HOSVD for a 3rd-order tensor based on using the left singular vectors of the SVDs of the matricizations:

- Compute $\mathbf{U}^{(1)}$ from SVD of $\boldsymbol{\mathcal{A}}^{(1)}$.
- Compute $\mathbf{U}^{(2)}$ from SVD of $\boldsymbol{\mathcal{A}}^{(2)}$
- Compute $\mathbf{U}^{(3)}$ from SVD of $\boldsymbol{\mathcal{A}}^{(3)}$.
- $\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{A}} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)}$

We can truncate terms to get a compressed representation. For $m \times p \times n$, numbers stored:

$$O(mk_1 + pk_2 + nk_3 + k_1k_2k_3)$$

We can also sequentially truncate[4] In our experience, little difference on performanace for applications. (Will depend on processing order).

[4] N. Vannieuwenhoven, R. Vandebril, and K. Meerbergen, "A new truncation strategy for the higher-order singular value decomposition," SIAM J. Sci. Comput, pp, 2012.

Large Scale Data Compression

Ballard, Klinvex, Kolda, "TuckerMPI: A Parallel C++/MPI Software Package for Large-scale Data Compression via the Tucker Tensor Decomposition," arXiv, 2019.

"We test the software on 4.5 terabyte and 6.7 terabyte data sets distributed across 100s of nodes (1000s of MPI processes), achieving compression ratios between 100200,000 which equates to 99-99.999 % compression (depending on the desired accuracy) in substantially less time than it would take to even read the same dataset from a parallel file system"

Randomized Variants

Capitalizing on recent successes in randomized numerical linear algebra, develop randomized variants.

Che and Wei. "Randomized algorithms for the approximations of Tucker and the Tensor Train decompositions." Advances in Computational Mathematics, 2018.

Minster, Saibaba, K, "Randomized Algorithms for Low-rank Tensor Decompositions in the Tucker Format," SIAM J. Mathematics of Data Sci., to appear.

Randomized variants that respect sparsity of the datasets.

Randomized Variants that Handle Sparsity

Formidable Repository of Sparse Tensors and Tools database.

Tensor	Dimensions	Nonzeros
NELL-2	$12092 \times 9184 \times 28818$	76,879,419
Enron	$6066 \times 5699 \times 244268 \times 1176$	54,202,099

NELL-2: entity $\times$ relation $\times$ entity (NELL is a machine learning system that relates different categories)

Enron: sender $\times$ receiver $\times$ word $\times$ date (word counts in emails released during an investigation by the FERC)

Approximate truncated (r, r, r) HOSVD and ST-HOSVD

Results

	Relative Error		Runtime in seconds	
r	SP-STHOSVD	R-STHOSVD	SP-STHOSVD	R-STHOSVD
20	0.6015	0.2081	0.4086	31.5615
45	0.3854	0.1259	0.7965	34.5802
145	0.0976	0.0332	3.5659	42.0969
195	0.0578	0.0180	6.8285	50.2907

Table: Results, Subsampled Enron dataset.

Taking advantage of the sparsity structure allows for faster compression[5].

[5] R. Minster, A.K. Saibaba, and M. E. Kilmer, "Randomized Algorithms for low-rank Decompositions in the Tucker Format," SIMODS, to appear.

TT and TT-SVD[6]

Suppose we can express each element of $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ as

$$\mathcal{A}_{i_1, i_2, \ldots, i_d} = (\mathcal{G}_1)_{:,:,i_1} \cdot (\mathcal{G}_2)_{:,:,i_2} \cdots (\mathcal{G}_d)_{:,:,i_d}$$

where each $\mathcal{G}_k$ is a core of size $r_{k-1} \times r_k \times n_k$ and $(\mathcal{G}_k)_{:,:,i_k}$ is an $r_{k-1} \times r_k$ matrix, with $r_0 = r_d = 1$,

Then, the TT-rank is the length-$(d+1)$ tuple $r = (r_0, r_1, \ldots, r_d)$

$\mathcal{G}_k$ is a stack of n_k matrices of size $r_{k1} \times r_k$.

Storage: $\sum_{k=1}^{d} r_{k-1} n_k r_k$

[6] V. Oseledets, "Tensor-train decomposition," SIAM J. Sci. Comput, 2011

3rd Order Example

Example: 3rd order,

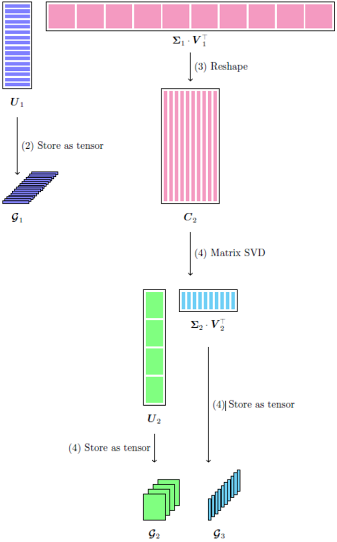
$$\mathcal{A}_{i,j,k} = (\mathcal{G}_1)_{1,:,i} \cdot (\mathcal{G}_2)_{:,:,j} \cdot (\mathcal{G}_3)_{:,1,k}$$

and $(\mathcal{G}_1)_{1,:,i}$ is $1 \times r_1$, $(\mathcal{G}_2)_{:,:,j}$ is $r_1 \times r_2$, $(\mathcal{G}_3)_{:,1,k}$ is $r_2 \times 1$.

If $r_1 = r_2 = 1$, then this reduces to a CP decomposition format.

TT SVD, 3rd Order[7]

From a mode-wise unfolding:



[7] Graphics: Newman, Tufts Ph.D. Thesis, 2019

So Far...

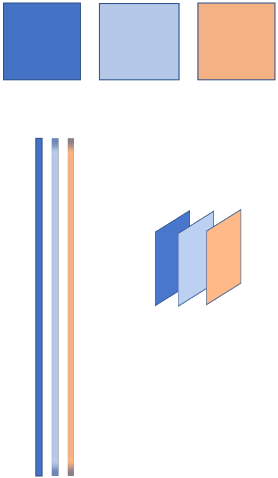
We have seen:

- CP, which is orientation independent, but no orthogonality, hard to find k, difficulties with algorithms
- HOSVD, orientation independent, orthogonal factor matrices, but no optimality on truncation with dense core.
- ST-HOSVD, process orientation dependent, orthogonal factor matrices, truncations prespecified
- TT-SVD, repeated unfoldings (process orientation dependent) and accumulating truncation errors, can be highly compressive

None relates to a framework wherein there is a product-based factorization of tensors. Optimality bounds, but no Eckart-Young like results.

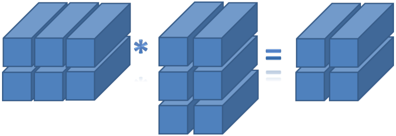
Tensor-Tensor Products

Orientation Dependent Data: Storage as $mn \times J$ matrix $\mathbf{A}$ or $m \times J \times n$ tensor $\boldsymbol{\mathcal{A}}$? Which is more compressible/interpretable?



Tensor-Tensor Products

Products between tensors of appropriate dimension that are well defined[8]

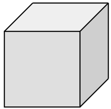


This allows us to define different tensor decompositions!

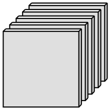
[8] K. and Martin, LAA (2011); Kernfeld, K, Aeron, LAA (2015)

Notation

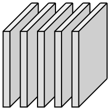
The basics



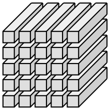
Tensor $\mathcal{A}$



Frontal slice $\mathbf{A}^{(k)}$



Lateral slice $\vec{\mathcal{A}}_j$



Tube fibers $\mathbf{a}_{ij}$

Indexing also done using MATLAB-like notation: e.g. $\vec{\mathcal{A}}_j = \mathcal{A}_{:,j,:}$.

Find a way to express a tensor that leads to the possibility for compressed representation that maintains important features of the original tensor.

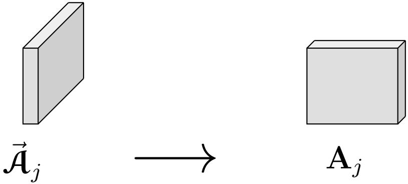
Outline for Remainder

- Algebraic framework for tensors as operators
  - Tensor-tensor products
  - Identities, transposes, orthogonality, etc.
- Tensor-tensor SVDs reminiscent of matrix SVD
- Eckart-Young theorem
- Randomized variants
- Applications (incl. POD)

K. & Martin, LAA 2011 K., Braman, Hoover, Hao, SIMAX 2013 Kernfeld, K, Aeron, 2015

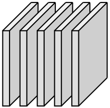
Operations for Tensor Manipulation

If $\vec{\mathcal{A}}_j$ is $m \times 1 \times n$, then sq$(\vec{\mathcal{A}}_j) = \mathbf{A}_j$ is $m \times n$.



Inverted by 'twisting'.

Mode-3 Multiplication



Lateral slices $\vec{\mathcal{A}}_j$ of $m \times p \times n$ tensor$\mathcal{A}$

$$\mathcal{A}_{(3)} := [\mathtt{sq}(\vec{\mathcal{A}}_1)^\top, \mathtt{sq}(\vec{\mathcal{A}}_2)^\top, \ldots, \mathtt{sq}(\vec{\mathcal{A}}_p)^\top]$$

Let $\mathbf{M}$ be $r \times n$. To find $\mathcal{A} \times_3 \mathbf{M}$:

- Compute matrix-matrix product $\mathbf{M}\mathcal{A}_{(3)}$,
- Reshape the result to an $m \times p \times r$ tensor.

Equivalent to applying $\mathbf{M}$ along tube fibers.

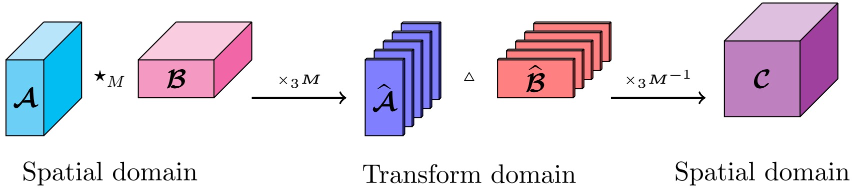
Star-M Product

Let $\mathbf{M}$ be any invertible, $n \times n$ matrix. Then

$$\widehat{\mathcal{A}} = \mathcal{A} \times_3 \mathbf{M} \text{ and } \mathcal{A} = \widehat{\mathcal{A}} \times_3 \mathbf{M}^{-1}.$$

Definition

Given any invertible, $n \times n$ $\mathbf{M}$, $\mathcal{A} \in \mathbb{C}^{m \times p \times n}$ and $\mathcal{B} \in \mathbb{C}^{p \times \ell \times n}$, $\mathcal{C} = \mathcal{A} \star_M \mathcal{B}$ is defined via $\widehat{\mathcal{C}}_{:,:,i} = \widehat{\mathcal{A}}_{:,:,i} \widehat{\mathcal{B}}_{:,:,i}$.



9

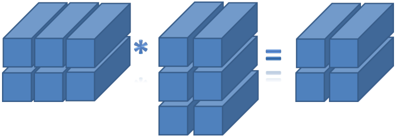
[9] Kernfeld,K, Aeron, LAA 2015

Special Case: The t-product

Special Case: Let $\mathbf{M}$ be the unnormalized DFT matrix[10].

The t-product can be computed in-place using FFTs:

- $\widehat{\mathcal{A}} \leftarrow \texttt{fft}(\mathcal{A}, [\,], 3)$
- $\widehat{\mathfrak{B}} \leftarrow \texttt{fft}(\mathfrak{B}, [\,], 3)$
- $\widehat{\mathfrak{C}}_{:,:,i} = \widehat{\mathcal{A}}_{:,:,i} \cdot \widehat{\mathfrak{B}}_{:,:,i}$, $i = 1, \dots, n$
- $\mathfrak{C} = \texttt{ifft}(\widehat{\mathfrak{C}}, [\,], 3)$



[10]K. and Martin, 2011

Other Properties

Definition (Conjugate Transpose)

Given $\mathcal{A} \in \mathbb{C}^{m \times p \times n}$ its $p \times m \times n$ **conjugate transpose** under $\star_M$ $\mathcal{A}^{\mathrm{H}}$ is defined such that $(\widehat{\mathcal{A}}^{\mathrm{H}})^{(i)} = (\widehat{\mathcal{A}}^{(i)})^{\mathrm{H}}, \qquad i = 1, \ldots, n.$

Definition (Unitary/Orthogonal Tensors)

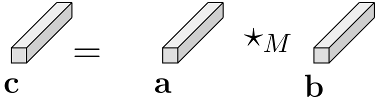
$\mathbf{\mathfrak{Q}} \in \mathbb{C}^{m \times m \times n}$ ($\mathbf{\mathfrak{Q}} \in \mathbb{R}^{m \times m \times n}$) is called $\star_M$-unitary ($\star_M$-orthgonal) if

$$\mathbf{\mathfrak{Q}}^{\mathrm{H}} \star_M \mathbf{\mathfrak{Q}} = \mathbf{\mathfrak{I}} = \mathbf{\mathfrak{Q}} \star_M \mathbf{\mathfrak{Q}}^{\mathrm{H}},$$

where H is replaced by transpose for real tensors. Note that $\mathbf{\mathfrak{I}}$ also defined under $\star_M$.

Kernfeld,K, Aeron, LAA 2015

Entry-wise $\mathbf{M}$-product

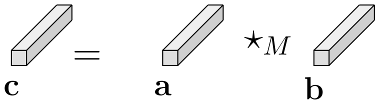


Tube fiber interpretation:

$$
\begin{aligned}
\mathbf{c} &= \texttt{fold}\left((\mathbf{M}^{-1}\text{diag}(\hat{\mathbf{a}})\mathbf{M})\texttt{vec}(\mathbf{b})\right) \\
&= \texttt{fold}\left((\mathbf{M}^{-1}\text{diag}(\hat{\mathbf{b}})\mathbf{M})\texttt{vec}(\mathbf{a})\right)
\end{aligned}
$$

Commutativity, and characterization using set of diagonal matrices diagonalized by $\mathbf{M}$ and its inverse.

Entry-wise $\mathbf{M}$-product



Tube fiber interpretation:

$$\begin{aligned} \mathbf{c} &= \texttt{fold}\left((\mathbf{M}^{-1}\text{diag}(\hat{\mathbf{a}})\mathbf{M})\texttt{vec}(\mathbf{b})\right) \\ &= \texttt{fold}\left((\mathbf{M}^{-1}\text{diag}(\hat{\mathbf{b}})\mathbf{M})\texttt{vec}(\mathbf{a})\right) \end{aligned}$$

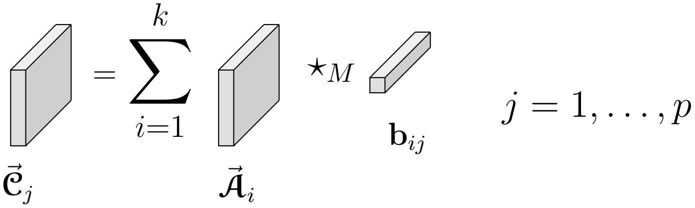
Commutativity, and characterization using set of diagonal matrices diagonalized by $\mathbf{M}$ and its inverse.

Special Case: $\mathbf{M}$ is DFT $\Rightarrow$ convolution, circulant matrices

Matrix-mimeticity

<u>Observation</u>: overloading scalar products with $\star_M$ in matrix-matrix algorithms gives product for larger dimensional tensors.

If $\mathcal{A}$ is $m \times k \times n$ and $\mathcal{B}$ is $k \times p \times n$, then $\mathcal{C}$ is $m \times p \times n$, and



Unitary Invariance

Theorem

If $\mathbf{M}$ a non-zero multiple of a unitary/orthogonal matrix[a]

$$\|\mathcal{Q} \star_M \mathcal{A}\|_F = \|\mathcal{A}\|_F$$

[a] K., Horesh, Avron, Newman (2019)

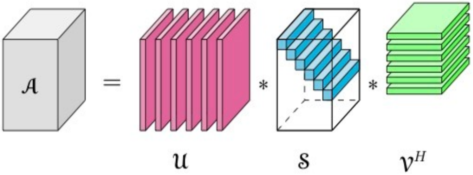
Tensor-tensor SVDs

Theorem (K., Horesh, Avron, Newman)

Let $\mathcal{A}$ be a $m \times p \times n$ tensor and $\mathbf{M}$ a non-zero multiple of a unitary/orthogonal matrix. The (full) $\star_M$ tensor SVD (t-SVDM) is

$$\mathcal{A} = \mathcal{U} \star_M \mathcal{S} \star_M \mathcal{V}^{\mathrm{H}} = \sum_{i=1}^{\min(m,p)} \mathcal{U}_{:,i,:} \star_M \mathcal{S}_{i,i,:} \star_M \mathcal{V}^{\mathrm{H}}_{:,i,:}$$

with $\mathcal{U}$, $\mathcal{V}$ $\star_M$-unitary, & $\|\mathcal{S}_{1,1,:}\|_F^2 \geq \|\mathcal{S}_{2,2,:}\|_F^2 \geq \ldots$



Algorithm

1: $\widehat{\mathcal{A}} \leftarrow \mathcal{A} \times_M \mathbf{M}$
2: **for all** $i = 1, \ldots, n$ **do**
3: $\quad [\widehat{\mathcal{U}}_{:,:,i}, \widehat{\mathcal{S}}_{:,:,i}, \widehat{\mathcal{V}}_{:,:,i}] = \texttt{svd}([\widehat{\mathcal{A}}_{:,:,i}])$ % note: rank $\widehat{\mathcal{A}}_{:,:,i}$ is ρ_i.
4: **end for**
5: $\mathcal{U} = \widehat{\mathcal{U}} \times_3 \mathbf{M}^{-1}$, $\mathcal{S} = \widehat{\mathcal{S}} \times_3 \mathbf{M}^{-1}$, $\mathcal{V} = \widehat{\mathcal{V}} \times_3 \mathbf{M}^{-1}$.

Perfectly parallelizable!

For face i, exist singular values $\hat{\sigma}_i^{(j)}$, $j = 1, .., \rho_i$

Eckart-Young

$\mathcal{A} \in \mathbb{R}^{m \times p \times n}$. For $k < \min(m,p)$, and $\mathbf{M}$ as previously, define

$$\mathcal{A}_k = \sum_{i=1}^{k} \mathcal{U}_{:,i,:} \star_M \left(\mathcal{S}_{i,i,:} \star_M \mathcal{V}^\top_{:,i,:}\right)$$

Then

$$\mathcal{A}_k = \arg\min_{\widetilde{\mathcal{A}} \in \Omega} \|\mathcal{A} - \widetilde{\mathcal{A}}\|_F$$

where $\Omega = \{\mathcal{X} \star_M \mathcal{Y} \,|\mathcal{X} \in \mathbb{R}^{m \times k \times n}, \mathcal{Y} \in \mathbb{R}^{k \times p \times n}\}$

Error: $\|\mathcal{A} - \mathcal{A}_k\|_F^2 = \sum_{j>k} \|\mathcal{S}_{j,j}\|_F^2 = c \sum_{i=1}^{n} \sum_{j>k} \hat{\sigma}_j^{(i)}$, c depends on $\mathbf{M}$.

Application: Facial Recognition when $\mathbf{M}$ is DFT[11]

- $\vec{\mathcal{X}}_j$, $j = 1, 2, \ldots, m$ are the training images
- $\vec{\mathcal{Y}}$ is the mean image
- $\vec{\mathcal{A}}_j = \vec{\mathcal{X}}_j - \vec{\mathcal{Y}}$ has the mean-subtracted images
- Left orthogonal $\mathcal{U}$ contains the principal components, so

$$\vec{\mathcal{A}}_j \approx \mathcal{U}_{:,1:k,:} \star_M \underbrace{(\mathcal{U}^\top_{:,1:k,:} \star_M \vec{\mathcal{A}}_j)}_{\text{tensor coeffs}}$$

- Compare tensor coefficients with $\mathcal{U}^\top_{:,1:k,:} \star_M \vec{\mathcal{B}}$, for a training image (tensor) $\vec{\mathcal{B}}$.

[11] Hao, K., Braman, Hoover, SIIMS (2013)

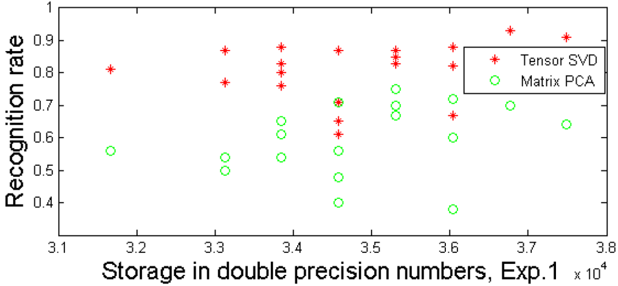
Facial Recognition Task

- Experiment 1: randomly select 15 images of each person as training, test all remaining images
- Experiment 2: randomly selected 5 images of each person as training, test all remaining images
- 20 trials for each experiment

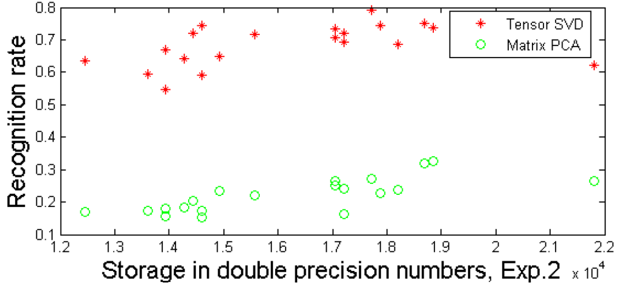
The Extended Yale Face Database B,

http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html

t-SVD vs. PCA



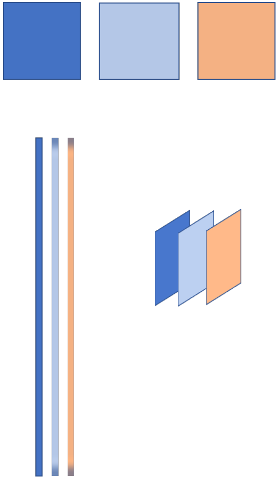
Recognition rate
0.4
0.5
0.6
0.7
0.8
0.9
1
3.1
3.2
3.3
3.4
3.5
3.6
3.7
3.8
Storage in double precision numbers, Exp.1
x 10^4
Tensor SVD
Matrix PCA



Recognition rate
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
1.2
1.3
1.4
1.5
1.6
1.7
1.8
1.9
2
2.1
2.2
Storage in double precision numbers, Exp.2
x 10^4
Tensor SVD
Matrix PCA

Data Comparison

In general, consider J pieces of 2D, $m \times n$ data. Storage as $mn \times J$ matrix $\mathbf{A}$ or $m \times J \times n$ tensor $\boldsymbol{\mathcal{A}}$. Which is more compressible?



Theoretical Result

Theorem (K.,Horesh,Avron,Newman (2019))

Suppose $\mathcal{A}_k$ is optimal k-term t-SVDM approximation to $\mathcal{A}$, and let $\mathbf{A}_k$ is optimal k-term matrix SVD approximation to $\mathbf{A}$. Then

$$\|\mathcal{A} - \mathcal{A}_k\|_F \leq \|\mathbf{A} - \mathbf{A}_k\|_F,$$

where **strict inequality is achievable***.*

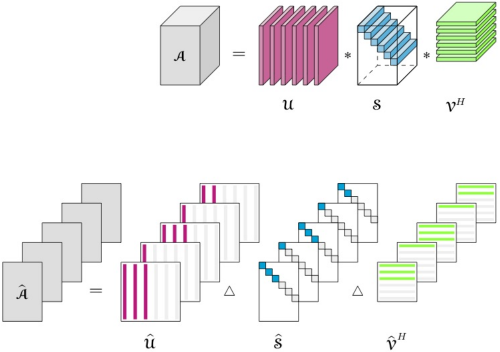
- Result works for any $\mathbf{M}$ that is multiple of unitary (orthogonal) matrix.
- Why? Takes advantage of latent structure in data.

t-SVDMII

Truncated t-SVDM ignores relative importance of faces.

Global approach: order $\hat{\sigma}_i^{(j)} := \widehat{\mathcal{S}}_{i,i,j}$, truncate on energy level.

Gives $\mathcal{A}_{\boldsymbol{\rho}}$, with $\rho_i = \text{rank}(\widehat{\mathcal{A}}^{(i)}.)$



Comparison

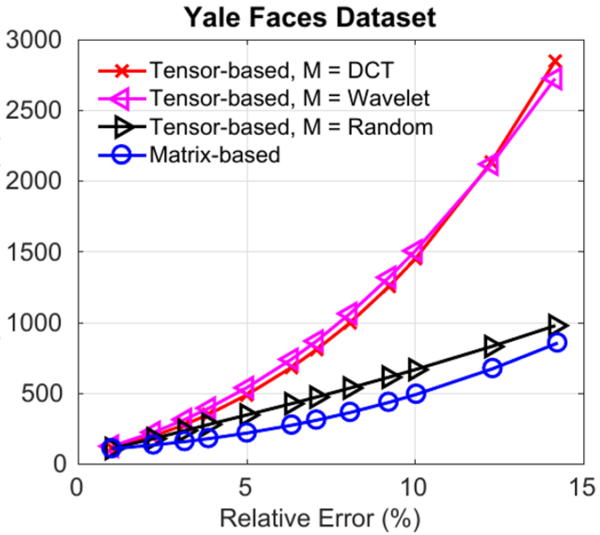
Implicit rank $=$ total number of non-zero $\hat{\sigma}_i^{(j)}$.

Theorem (K.,Horesh,Avron,Newman, 2019)

Let $\mathcal{A}_k$ be the t-SVDM t-rank k approximation to $\mathcal{A}$, and suppose its implicit rank is r. Define $\mu = \|\mathcal{A}_k\|_F^2 / \|\mathcal{A}\|_F^2$. There exists $\gamma \leq \mu$ such that the t-SVDMII approximation, $\mathcal{A}_{\boldsymbol{\rho}}$, obtained for this γ, has implicit rank less than or equal to the implicit rank of $\mathcal{A}_k$ and

$$\|\mathcal{A} - \mathcal{A}_{\boldsymbol{\rho}}\|_F \leq \|\mathcal{A} - \mathcal{A}_k\|_F \leq \|\mathbf{A} - \mathbf{A}_k\|_F.$$

Yale Example



Truncated-HOSVD in the $\star_M$ Framework

Define $\mathbf{M} = (\mathbf{U}^{(3)})^\top$ from the HOSVD. Then we can express the HOSVD in our tensor framework, and we can show that our t-SVDM, t-SVDMII are superior to tr-HOSVD for appropriate truncation levels, as well.

Data Compression

Not all tensor decompositions are created equal!

(a) Orig

(b) tr-tSVDM2

(c) tr-Mtx

(d) tr-H$(m, 25, n)$

(e) tr-H$(70, 53, 53)$

Other Data? An Application in POD.

Discretize dynamical system by n_x, n_y points in space.

$$\frac{\partial \bar{\mathbf{u}}(t)}{\partial t} = \mathbf{A}\bar{\mathbf{u}}(t) + f(\bar{\mathbf{u}}(t)) + \mathbf{q}(t), t \geq 0$$

Want $\bar{\mathbf{u}} \approx \mathbf{P}_r\bar{\mathbf{u}} = \mathbf{B}\underbrace{\mathbf{B}^\top\bar{\mathbf{u}}}_{\tilde{\mathbf{u}}}$ where $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_r]$ is orthonormal basis for projected state space.

Then we replace $\bar{\mathbf{u}}(t)$ by $\mathbf{B}\tilde{\mathbf{u}}$, and solve the projected problem:

$$\frac{\partial \tilde{\mathbf{u}}(t)}{\partial t} = \mathbf{B}^\top\mathbf{A}\mathbf{B}\tilde{\mathbf{u}}(t) + \mathbf{B}^\top f(\mathbf{B}\tilde{\mathbf{u}}(t)) + \mathbf{B}^\top\mathbf{q}(t)$$

An Application in POD

In practice, get a snapshot matrix $\mathbf{X} = [\bar{\mathbf{u}}^{(1)}, \ldots, \bar{\mathbf{u}}^{(s)}]$ & $\mathbf{B}$ solves

$$\min_{\mathbf{B} \in \mathbb{R}^{n \times r}} \|\mathbf{X} - \mathbf{B}\mathbf{B}^\top \mathbf{X}\|_F \text{ s.t. } \mathbf{B}^\top \mathbf{B} = \mathbf{I}.$$

Thus, $\mathbf{B}$ is the first r left singular vectors of $\mathbf{X}$.

An Application in POD

Our idea[12]: Compute the snapshot tensor, we construct $\mathbf{B}$ from the left singular tensor instead, since the t-SVDM (t-SVDMII) solves the optimization problem under $\star_M$.

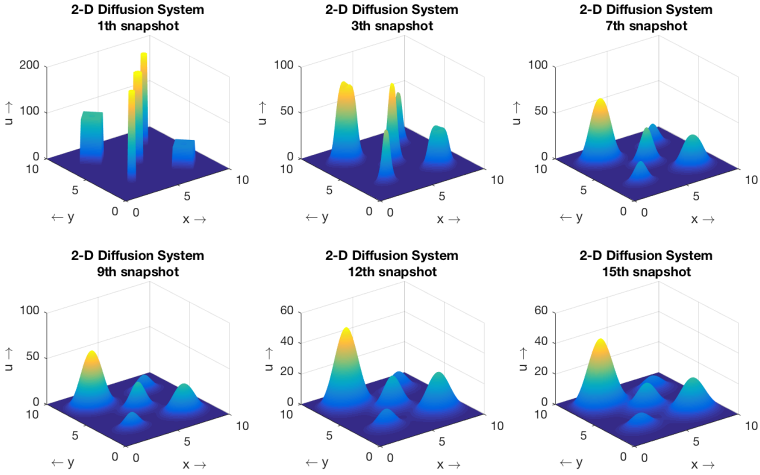
So far, we have tested this for $\mathbf{M}$ being the DFT matrix.

(Ultimately, forming the projected problem requires some manipulation back in 'matrix-vector' land.)

[12] See Jiani Zhang's Ph.D. Thesis, Tufts, 2017

Example

Diffusion Equation: $\frac{\partial u(\mathbf{r},t)}{\partial t} - \nabla \cdot \kappa \nabla u(\mathbf{r},t) = 0$



Better Basis? - Numerical Support

Diffusion Equation: $\frac{\partial u(\mathbf{r},t)}{\partial t} - \nabla \cdot \kappa \nabla u(\mathbf{r},t) = 0$

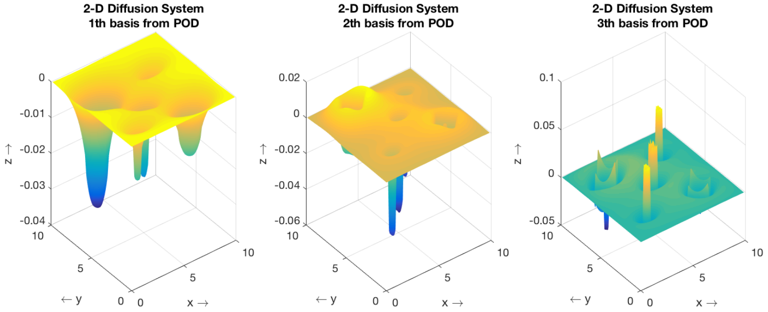


Figure: The first three basis vectors from SVD.

Better Basis? - Numerical Support

Diffusion Equation: $\frac{\partial u(\mathbf{r},t)}{\partial t} - \nabla \cdot \kappa \nabla u(\mathbf{r},t) = 0$

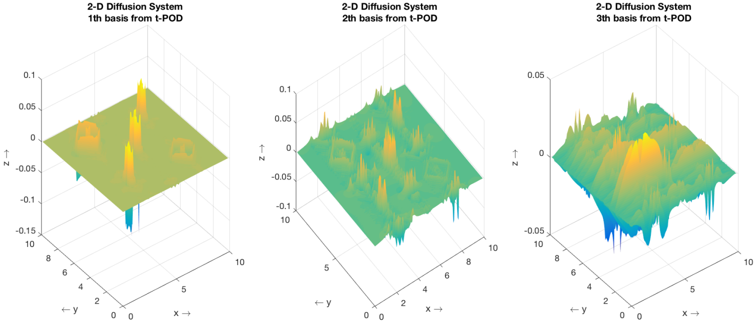
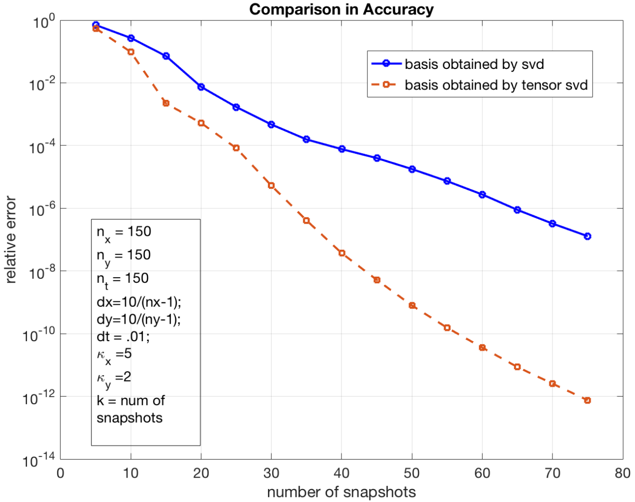


Figure: The first three basis slices from t-SVD.

Better Basis? - Numerical Support

Diffusion Equation: $\frac{\partial u(\mathbf{r},t)}{\partial t} - \nabla \cdot \kappa \nabla u(\mathbf{r}, t) = 0$



Computation Cost

What about the computational cost comparison?

Computation Cost

What about the computational cost comparison?
Computing basis: tensor SVD, independent matrix computations in the transform domain

Size of reduced model: $n_y k$ ("expensive" by comparison to k), if we assume same value of k needed in matrix vs. tensor case.

Computation Cost

What about the computational cost comparison?
Computing basis: tensor SVD, independent matrix computations in the transform domain

Size of reduced model: $n_y k$ ("expensive" by comparison to k), if we assume same value of k needed in matrix vs. tensor case.

Two improvements that address cost issue (details omitted)

- Use t-SVDMII.
- Reduce snapshot data from two directions (sequential truncation variant!).

Computation Cost

What about the computational cost comparison?
Computing basis: tensor SVD, independent matrix computations in the transform domain

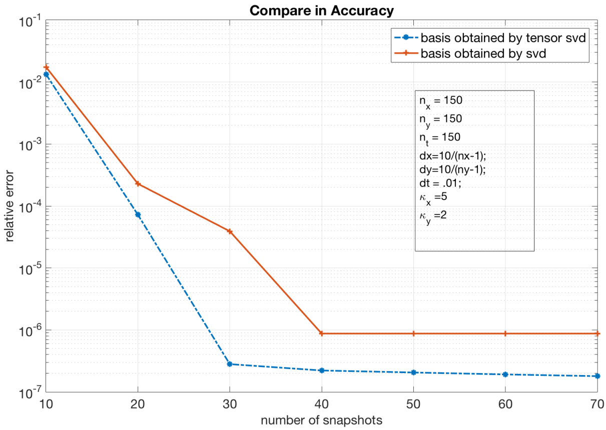
Size of reduced model: $n_y k$ ("expensive" by comparison to k), if we assume same value of k needed in matrix vs. tensor case.

Two improvements that address cost issue (details omitted)

- Use t-SVDMII.
- Reduce snapshot data from two directions (sequential truncation variant!).

Prelim Results with Enhancements, $k = 30$

Diffusion Equation: $\frac{\partial u(\mathbf{r},t)}{\partial t} - \nabla \cdot \kappa \nabla u(\mathbf{r},t) = 0$



Summary

- Multiway data can be compressed through various tensor decompositions; only covered a small number
- The various decompositions offer distinct features, some may be better than others on certain applications
- Randomized variants possible (see also Zhang et al, for randomized t-SVD) for speed
- Variants available the addess concerns with sparsity
- Parallelizable computations and memory efficient computations
- Showed only one POD example, but other uses of tensors in context of ROM are still under investigation
- Great deal of matrix-structure that I barely touched on, there may be more problems amenable to tensor treatment