Distance Metric Learning and Tensor Methods: Completion, Decomposition, Recovery, and Reconstruction

Maryam Bagherian
Department of Computational Medicine and Bioinformatics
Michigan Institue for Data Science
University of Michigan, Ann Arbor
bmaryam@umich.edu

Content

Introduction
- History
- Tensor
- Tensor Rank

Tensor Completion
- Low Rank Tensor Completion

Tensor Decomposition
- CP decomposition
- Tucker Decomposition

Tensor Recovery and Tensor Reconstruction
- Definition

Distance Metric Learning
- Why and What for
- Mahalanobis Distance

Applications

References

Introduction

History

- *The term tensor comes from the Latin "**tendere**", which means "**to stretch**."*
- *In 1822 **Cauchy** introduced the Cauchy stress tensor in continuum mechanics;*
- *In 1861 **Riemann** created the Riemann curvature tensor in geometry,*
- *Neither **Cauchn** nor **Riemann** used the term tensor, which was introduced later around 1900 (next slides).*

History

- *In 1884, **Gibbs** introduced tensor products of vectors in $\mathbb{R}^3$ with the label "indeterminate product" and applied it to study strain on a body. He extended the indeterminate product to n dimensions in 1886* [a].

- ***Voigt** used tensors to describe stress and strain on crystals in 1898, and the term tensor first appeared with its modern physical meaning there.*

- *In geometry **Ricci** used tensors in the late 1800s and his 1901 paper with **Levi-Civita** (in English) was crucial in **Einstein**'s work on general relativity.*

- *Wide use of the term "tensor" in physics and math is due to **Einstein**;*

[a] "Elements of Vector Analysis Arranged for the Use of Students in Physics,"

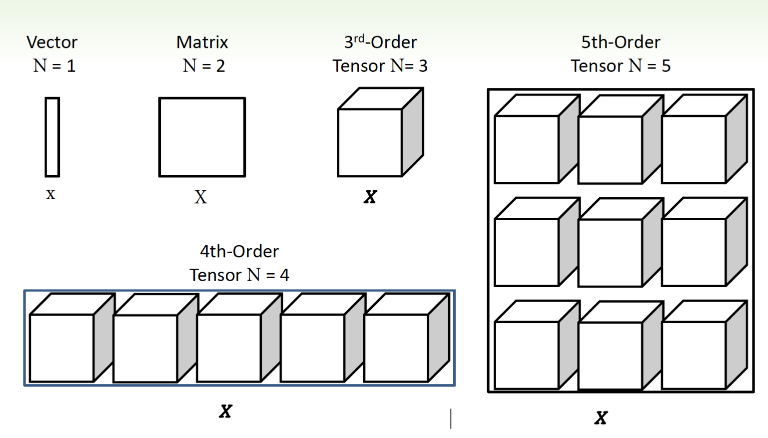
History

- ***Ricci*** *and* ***Levi-Civita*** *called tensors by the bland name "systems".*
- *The notation $\otimes$ is due to* ***Murray*** *and* ***Von Neumann*** *in 1936 for tensor products (they wrote "direct products") of Hilbert spaces.*
- *The tensor product of abelian groups G and H, with that name but written as $G \circ H$ instead of $G \otimes H$, is due to Whitney in 1938.*
- *Tensor products of modules over a commutative ring are due to* ***Bourbaki*** *in 1948.* [a]

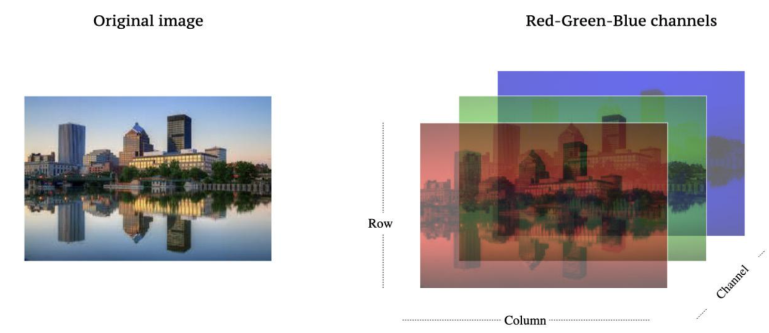
[a] see [0].

Introduction

High dimensional arrays



Introduction
Example: 3rd order Tensor



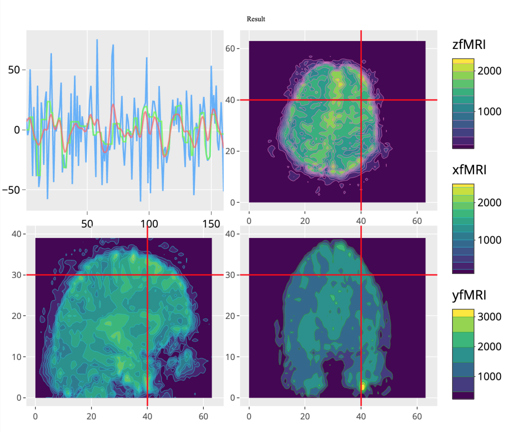


Figure: Functional magnetic resonance imaging (fMRI) with front view, side view, and top view.

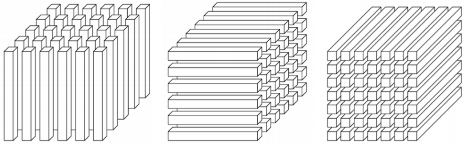
1

[1]Source: Spacekime Analytics (Time Complexity and Inferential Uncertainty)

Definition (Tensor subarrays)

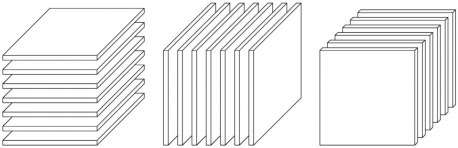
Analogous to *row* and *column* of the matrices, we may define tensor *fiber* and *slice* as following:

- **Fiber**: Fibers are defined by fixing every index of the tensor but one.
- **Slice**: Slices are two-dimensional sections of a tensor, defined by fixing all but two indices.



(a) Mode-1 (column) fibers: $\mathbf{x}_{:jk}$ (b) Mode-2 (row) fibers: $\mathbf{x}_{i:k}$ (c) Mode-3 (tube) fibers: $\mathbf{x}_{ij:}$

Fig. 2.1 *Fibers of a 3rd-order tensor.*



(a) Horizontal slices: $\mathbf{X}_{i::}$ (b) Lateral slices: $\mathbf{X}_{:j:}$ (c) Frontal slices: $\mathbf{X}_{::k}$ (or $\mathbf{X}_k$)

Fig. 2.2 *Slices of a 3rd-order tensor.*

One may find several definition for the rank of a tensor. In general, the rank of a tensor is defined as following:

Definition (Rank of a tensor)

The rank of a tensor $\mathcal{X}$ is the minimal number of rank-1 tensors that yield $\mathcal{X}$ in a linear combination.

Definition (Order of a tensor)

Number of modes is called order of the tensor. For instance, $\mathcal{X} \in \mathbb{R}^{I \times J \times K \times L}$ is a tensor of order 4 (a.k.a 4-way tensor, 4th order tensor).

- The mode-n rank of a tensor $\mathcal{X}$ is the dimension of the subspace spanned by its mode-n vectors.
- The mode-n rank of a higher-order tensor is the obvious generalization of the column (row) rank of a matrix.

Remark (Rank of a tensor)

- *Even for third order tensors, the problem of finding the rank of the tenor is NP-hard. "Approximation" is the best we can do.*
- *We know some upper bound for the rank of the tensor. For instance, for a tensor $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$, the rank of the tensor $\mathcal{X}$ is never more than*

$$\min\{IJ, IK, JK\}.$$

2

[2] NP: Non-deterministic Polynomial-time hard. You may see: Hastad, Johan. "Tensor rank is NP-Complete." In International Colloquium on Automata, Languages, and Programming, pp. 451-460. Springer, Berlin, Heidelberg, 1989.

Tensor Completion

Definition (Completion Problem)

In general, the task of filling in the missing entries of a partially observed array is called *completion* task. One of the variants of the completion problem is to find the lowest rank array that matches the partially observed one and is called *low rank completion* problem.

Low rank is often a necessary hypothesis to restrict the degree of freedoms of the missing entries.

Definition (Low-Rank Tensor Completion Problem)

Let a data tensor $\mathcal{X}$ and the index set Ω denoting the indices of observations be given. The completion task can be formulated as the following well-known optimization problem:

$$\begin{aligned} \min_{\mathcal{S}} \quad & \text{rank}(\mathcal{S}) \\ \text{s.t.} \quad & \Omega_{\mathcal{X}} = \Omega_{\mathcal{S}}, \end{aligned}$$

where $\mathcal{S}$ is the completed low rank tensor w.r.t. $\mathcal{X}$.

Remark

The completion optimization problem is non-convex, since the function rank$(\cdot)$ is non-convex. A common approach is to use a proper norm as a convex relaxation of the problem to to approximate the rank of the tensor. Therefore, we can instead minimize the following objective:

$$\begin{aligned} \min_{\mathcal{S}} \quad & \|\mathcal{S}\|_\mu \\ \textit{s.t.} \quad & \Omega_{\mathcal{X}} = \Omega_{\mathcal{S}}, \end{aligned}$$

where $\|\cdot\|_\mu$ is a norm which has to be identified and $\mathcal{S}$ is the completed low rank tensor w.r.t. $\mathcal{X}$.

Remark (A few examples for further studies)

- *Decomposition Based Approaches (we will see them in the next section!)*
- *Trace Norm Based Approaches :*
  - *A Simple Low Rank Tensor Completion (SILRTC)*
  - *High Accuracy Low Rank Tensor Completion (HALRTC): similar to SILRTC but using ADMM*
- *Other Variants*
  - *Non-Negative Constrained Approaches*
  - *Robust Tensor Completion Methods*
  - *Riemannian Optimization*

[3]

[3]For more details you may refer to: Song *et al.* "Tensor completion algorithms in big data analytics." ACM Transactions on Knowledge Discovery from Data (TKDD) 13, no. 1 (2019): 1-48.

Tensor Decomposition

Remark

- *Tensor decompositions (a.k.a factorization) are the new matrix factorizations!*
  One may categorize them into two main classes:
  - *CP-based models*
  - *Tucker-based models*
- *They can be unique under certain conditions.*

Theorem (Weierstrass)

A general tensor $\mathcal{X}$ of the size $n \times n \times 2$ has a "unique" tensor decomposition as a sum of n decomposable tensors

Note: The decomposition is unique *up to reordering the summands*.

CP Decomposition

- Classification
- Denoising
- Image compression and classification
- etc.

Definition (CP decomposition)

A canonical or parallel factor decomposition **C**ANDECOMP/**P**ARAFAC Decomposition (CP decomposition) decomposes a tensor into linear combination of rank-1 components. Given a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, CP decomposition writes the tensor as

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{u}_r^{(1)} \circ \cdots \circ \mathbf{u}_r^{(N)},$$

where vector $\mathbf{u}_r^{(n)} \in \mathbb{R}^{I_n}$.

Similar to the *outer product of two vectors*, we may define:

Definition (The outer product)

Given two tensors $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$ and $\mathcal{Y} \in \mathbb{R}^{J_1 \times J_2 \cdots \times J_M}$ (not necessarily of the same size), denoted by "$\circ$", the outer product of $\mathcal{X}$ and $\mathcal{Y}$ results in a tensor of the size $(I_1, I_2, \cdots, I_N, J_1, J_2, \cdots, J_M)$ and is defined as:

$$(\mathcal{X} \circ \mathcal{Y})_{i_1 i_2 \cdots i_N j_1 j_2 \cdots j_M} = x_{i_1 i_2 \cdots i_N} \, y_{j_1 j_2 \cdots j_M}.$$

Remark (On CP decomposition)

- *One way to find/approximate the rank of a tensor is the number R in CP decomposition of the tensor.*
- *Unlike matrix factorization, CP decomposition may be unique under a mild rank condition:*

Theorem (Kruskal Condition)

For a three-way tensor in $\mathcal{X} \in \mathbb{C}^{I \times J \times K}$, the PARAFAC decomposition is essentially unique if

$$2R \leq rankk(\mathbf{v}_1) + rankk(\mathbf{v}_2) + rankk(\mathbf{v}_3) - 2,$$

where $rankk(\cdot)$ denotes the Kruskal rank.

Remark (Why uniqueness is important?)

- *Consider Principle Component Analysis (PCA):*
  - *"If two or more roots [eigenvalues] are equal, the directions of the associated axes are not unique and may be chosen in an infinity of orthogonal positions." (Morrison (1990, Def. 8.3).)*
  - *In order to make sense of the principal components, they must be unique otherwise they are not more than random axes, and may not describe anything that is meaningful;*
- *Consider Non-negative Matrix Factorization (NMF):*
  - *NMF is, in general, non-unique. One can inquire about existence and uniqueness of NMF without any other side information;*
  - *There is always the question of whether or not these true latent factors are the only interpretation of the data, or alternative ones exist.*

The algorithms to compute the CP decomposition minimize the norm of the difference between the actual tensor, $\mathcal{X}$, and the approximation resulting from the potential decomposition, which we denote $\hat{\mathcal{X}}$. In particular, we look to solve the following optimization problem:

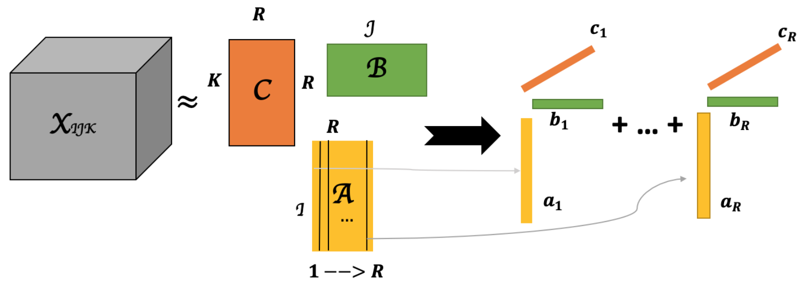
Definition

$$\min_{\{\mathbf{u}^{(n)}\}_{n=1}^{N}} \quad \left\| \mathcal{X} - \sum_{r=1}^{R} \lambda_r \, \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \cdots \circ \mathbf{u}_r^{(N)} \right\|_F$$

for some regularization scalar λ_r.

Remark

The CP decomposition is sometimes expressed in the form of factor matrices where the vectors from the rank one tensor components are combined to form factor matrices. For the decomposition expression shown in the previous slide, the factor matrices $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$ *will be formed as* $\mathbf{A} = [\mathbf{a}_1 \ \ \mathbf{a}_2 \ \ \cdots \ \ \mathbf{a}_R]$, *where R is the number of components.*



Remark

- *The most popular algorithm for CP decomposition, called Alternating Least Squares (CP–ALS)*
  - *It matricizes the tensor, $\mathcal{X}$, and arranges the factor vectors into columns of matrices. It then alternates between solving for each of the matrices until the desired stopping criterion is achieved.*
- *There are two other main group of algorithms for CP decomposition:*
  - *Block descent algorithms*
    - *It update whole factor matrices at each step*
  - *Power methods*
    - *greedy algorithms that perform ALS-type updates on one factor vector at a time and results in a series of rank one approximations*

Tucker Decomposition

- Image compression and classification
- Dimensionality reduction
- Edge computing
- etc.

Definition (Tucker Decomposition)

Let $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ be an N-mode tensor. Tucker method decomposes tensor $\mathcal{X}$ in to a core tensor

$$\mathcal{G} \in \mathbb{R}^{i_1 \times \cdots \times i_N}, \quad \text{s.t.} \quad \forall n,\ i_n \leq I_n,$$

and factor matrices $\mathcal{U}^{(n)} \in \mathbb{R}^{I_n \times i_n}$, satisfying

$$\mathcal{X} \approx \mathcal{G} \times \{\mathbf{U}\} = \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_N \mathbf{U}^{(N)}.$$

Remark

The Tucker decomposition generalizes principal component analysis for tensors. That is, it takes a tensor $\mathcal{X} \in \mathbb{R}^{I \times J \times K}$, let say 3-way tensor, and breaks it down into:

- *a smaller "core" tensor $\mathcal{G} \in \mathbb{R}^{i \times j \times k}$, with $i \leq I, j \leq J, k \leq K$,*

which is transformed along each mode by factor matrices

- $\mathbf{U}^{(1)} \in \mathbb{R}^{I \times i}$, $\mathbf{U}^{(2)} \in \mathbb{R}^{J \times j}$, *and* $\mathbf{U}^{(3)} \in \mathbb{R}^{K \times k}$,

Definition

For an order-3 tensor, the Tucker decomposition is written as

$$\min_{\mathcal{G}, \{\mathbf{U}^{(n)}\}_{n=1}^{3}} \left\| \mathcal{X} - \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \right\|_F .$$

Remark

- *Warning: Tucker decomposition is NOT unique! (Why?)*
- *Tucker decomposition is used as a modelling tool.*

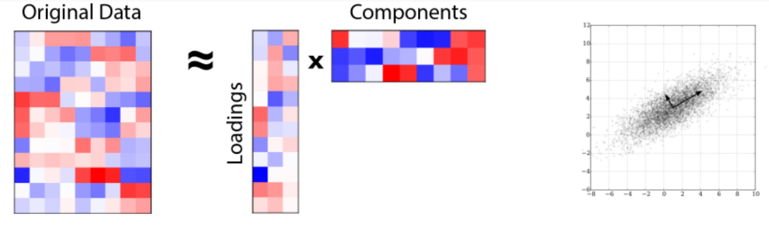
Tucker Decomposition

Remark

- *The most popular Tucker decomposition algorithm, called Higher Order Singular Value Decomposition (HOSVD)*
  - *It works with the matricization of this problem and solves for each factor matrix in turn.*
  - *It is not an iterative method*
- *There are two other main group of algorithms for Tucker decomposition:*
  - *Tensor sketching*
  - *Parallelization*

Singular Value Decomposition (SVD):

- SVD is one of the most important tools in multivariate analysis.
- Goal: Find the underlying low-rank structure from the data matrix
- Closely related to Principal component analysis (PCA): Find the one/multiple directions that explain most of the variance.

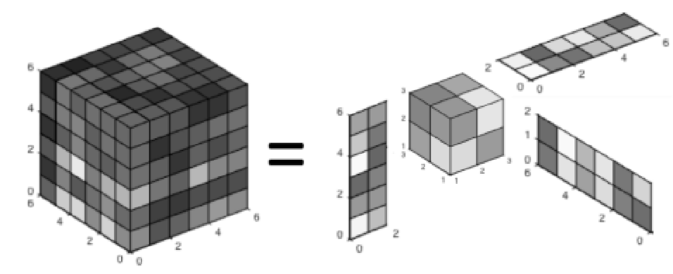


4

[4] Figure credit: Anru Zhang

Higher Order Singular Value Decomposition (HOSVD):

- Historically, much of the interest in higher-order SVDs was driven by the need to analyse empirical data, especially in psychometrics and chemometrics.
- The HOSVD can be built from several SVDs (see the next slide)



5

[5] Figure credit: Anru Zhang

Remark (HOSVD as generalization of SVD)

Let tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ *be given.*

I . *Construct the mode n matricization of tensor* $\mathcal{X}$, $\mathbf{X}_{(n)}$, *now we have a matrix to work with!*

II . *Compute the singular value decomposition (SVD) for matrix* $\mathbf{X}_{(n)}$, *that is*

$$\mathbf{X}_{(n)} = \mathbf{U}_n \Sigma_n \mathbf{V}_n^\top,$$

III . *Store the left singular values* $\mathbf{U}_n$

IV . *The core tensor* $\mathcal{G}$ *is then the projection of* $\mathcal{X}$ *onto the tensor basis formed by the factor matrices* $\{\mathbf{U}_n\}_{n=1}^N$, *i.e.,*

$$\mathcal{G} = \mathcal{X} \bigotimes_{n=1}^{N} \mathbf{U}_n^\top.$$

Tensor Recovery and Tensor Reconstruction

Definition (Tensor Reconstruction)

Given a data tensor $\mathcal{X}$, the aim of finding an unknown tensor $\mathcal{Z}$, that caries the decomposition structure of $\mathcal{X}$, is called tensor reconstruction.

Definition (Tensor Recovery)

A tensor reconstruction based on completion methods is called tensor recovery.

Distance Metric Learning

Definition (Distance Metric Learning)

Distance metric learning is a branch of machine learning that aims to learn distances from the data instead of using familiar distances.

Remark

Similarity-based learning algorithms are among the earliest used in the field of ML. To name a few early applications: k-Nearest Neighbors (k-NN) rule, Clustering: k-means algorithm and etc.

Remark

- *To measure the similarity between data, it is necessary to introduce a distance;*
- *There is an infinite number of distances we can work with, and not all of them will adapt properly to our data;*
- *Distance metric learning arises in algorithms that are capable of searching for distances that are able to capture features or relationships hidden in data.*

Remark

- *In a K-dimensional Euclidean space, the squared distances can then be computed as*

$$d(\mathbf{a}, \mathbf{b}) = \|\mathbf{L}(\mathbf{a} - \mathbf{b})\|_F^2$$

  where $\mathbf{L}$ *is a surjective linear transformation.*
- *The distance may also be expressed in terms of a positive semidefinite square matrix* $\mathbf{M} = \mathbf{L}\mathbf{L}^\top$.
- *In case* $\mathbf{L}$ *is also surjective, which results in* $\mathbf{M}$ *being full rank, the matrix* $\mathbf{M}$ *parametrizes the distance* d.
- *Matrix* $\mathbf{M}$ *is referred to as Mahalanobis metric. In Gaussian distributions matrix* $\mathbf{M}$ *plays the role of the inverse covariance matrix.*

Remark

- *Mahalanobis distances come from the (semi-)dot products in* $\mathbb{R}^d$ *defined by the positive semidefinite matrix* $\mathbf{M}$.
- *When* $\mathbf{M}$ *is full-rank, Mahalanobis distances are proper distances. Otherwise, they are pseudodistances.*
- *The Euclidean usual distance is a particular example of a Mahalanobis distance, when* $\mathbf{M}$ *is the identity matrix* $\mathbf{I}$.

Remark

- *Mahalanobis distances have additional properties specific to distances over normed spaces,*
  - ***Homogeneousness***: $d(ax, ay) = |a|\, d(x, y)$, *for* $a \in \mathbb{R}$, $x, y \in \mathbb{R}^d$,
  - ***Translation invariance***: $d(x, y) = d(x + z, y + z)$, *for* $x, y, z \in \mathbb{R}^d$,

Definition

Let $\mathcal{X}_i \in \mathbb{R}^{N_1 \times N_2 \times \cdots \times N_K}$ be input sample tensors, for some $i \in \mathbb{N}$. Consider the multilinear transformation

$$\varphi : \quad \mathbb{R}^{N_1 \times N_2 \times \cdots \times N_K} \to \mathbb{R}^{N_1 \times N_2 \times \cdots \times N_K}$$
$$\varphi(\mathcal{X}) = \mathcal{X} \times_1 \mathbf{L}^{(1)} \times_2 \mathbf{L}^{(2)} \cdots \times_K \mathbf{L}^{(K)},$$

where the square matrices $\mathbf{L}^{(\ell)} \in \mathbb{R}^{N_\ell \times N_\ell}$, for $\ell = 1, \cdots, K$, are called the ℓ-mode matrices. The squared Mahalanobis distance can be computed as:

$$d_M(\mathcal{X}_i, \mathcal{X}_j) := \left\| (\mathcal{X}_i - \mathcal{X}_j) \times_1 \mathbf{L}^{(1)} \times_2 \mathbf{L}^{(2)} \cdots \times_K \mathbf{L}^{(K)} \right\|_F^2,$$

Lemma

Assuming the setting in previous definition, then

$$d_M(\mathcal{X}_i, \mathcal{X}_j) := Tr\left(\hat{\mathbf{L}}^{(\ell)} (\mathbf{X}_i - \mathbf{X}_j)_{(\ell)} \, \mathcal{L}_{\otimes}^{\neq \ell} \hat{\mathbf{L}}^{(k)}\right) (\mathbf{X}_i - \mathbf{X}_j)_{(\ell)}^{\top} \Big),$$

where $\otimes$ *denotes Kronecker product, and for each* ℓ, $\mathbf{X}_{(\ell)}$ *denotes the* ℓ*-th matricization of tensor* $\mathcal{X}$, *with*

$$\hat{\mathbf{L}}^{(\ell)} = \mathbf{L}^{(\ell)^{\top}} \mathbf{L}^{(\ell)},$$

and

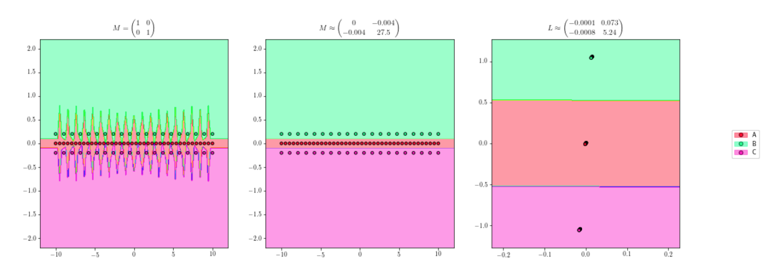
$$\mathcal{L}_{\otimes}^{\neq \ell} = \hat{\mathbf{L}}^{(K)} \otimes \cdots \hat{\mathbf{L}}^{(\ell+1)} \otimes \hat{\mathbf{L}}^{(\ell-1)} \otimes \cdots \otimes \hat{\mathbf{L}}^{(1)}.$$

Here, if $\mathbf{L}^{(\ell)}$, *for* $\ell = 1, \cdots, K$ *are orthogonal matrices ,the distance recovers Euclidean distance.*

Example (Improve the performance of distance-based classifiers)

- Suppose we have a dataset in the plane, where data can belong to three different classes, whose regions are defined by parallel lines. We want to classify new samples using the one nearest neighbor classifier.
- Euclidean distance (left): because there is a greater separation between each sample in class B and class C than there is between the regions.
- An adequate distance and try to classify with the nearest neighbor classifier again, we obtain much more effective classification regions, as shown in the center image.
- Learning a metric is equivalent to learning a linear map and to use Euclidean distance in the transformed space.
- We can also observe that data are being projected, except for precision errors, onto a line, thus we are also reducing the dimensionality of the dataset.

Example (Improve the performance of distance-based classifiers)



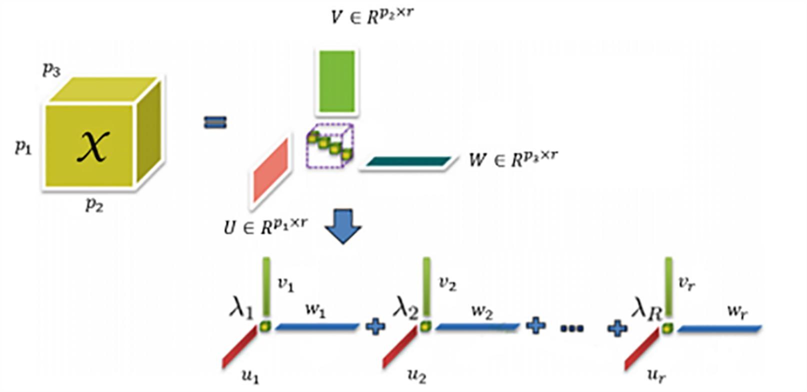
a

---

[a] Reference: [9]

Examples and Applications

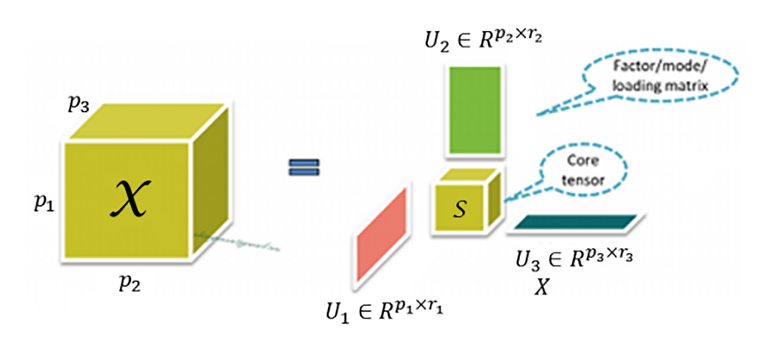
CP Model:



[6]

[6] Figure credit: Anru Zhang

Applications
Tucker Decomposition



[7]

---

[7]Figure credit: Anru Zhang

Example (CP for Simultaneous Analysis of Neurons, Time, and Trial)

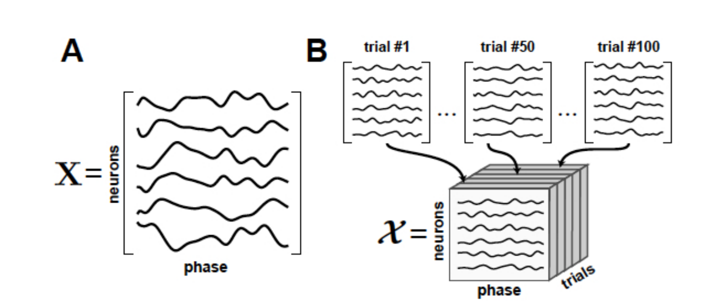


Figure: Neural activity data represented in matrix and tensor formats. (A) Activity of a neural population on a single trial. (B) Activity of a neural population across multiple trials.

8 Credit: Alex Willimas, "Notes on Tensor Decompositions", 2016.

Example (CP for Simultaneous Analysis of Neurons, Time, and Trial)

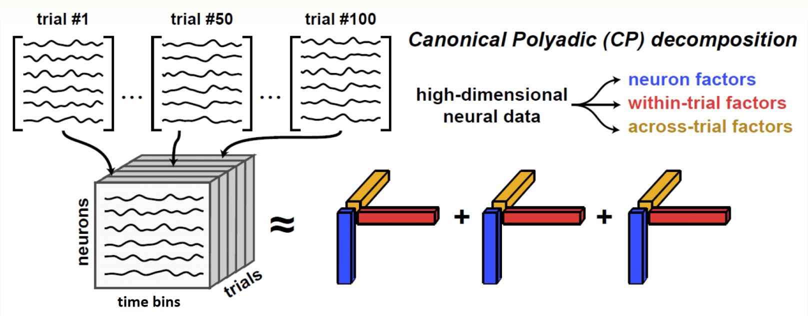


Figure: Past work could only look at 2 factors at once: Time and Neuron, Trial and Neuron, etc.

9

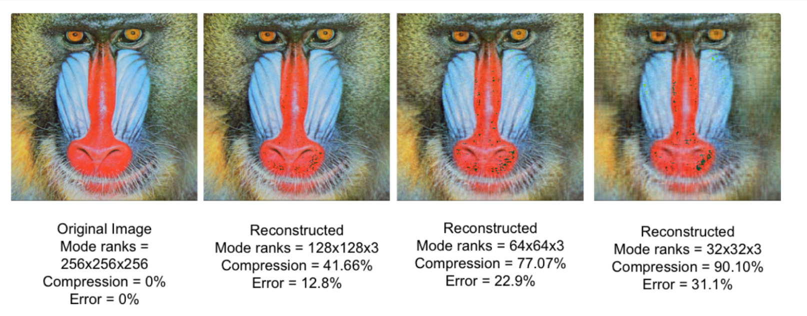
[9] Credit: Kolda [1]

Example

- For instance, given a three-way (or higher way) data, it is used to model the data by means of relatively small numbers of components for each mode.
- Components are linked to each other by a three- (or higher-) way core array.
- The model parameters are estimated in such a way that, given fixed numbers of components, the modelled data optimally resemble the actual data in the least squares sense.

The model gives a summary of the information in the data, in the same way as principal components analysis (PCA) does for two-way data.

Example (Application in Image Compression)

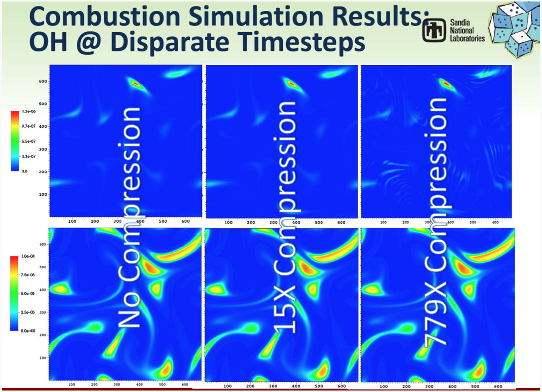


[10]

---

[10] Credit: *https : //iksinc.online/2018/05/02/understanding − tensors − and − tensor − decompositions − part − 3/*

Example (Reduce Storage Bottleneck via Multiway Tucker-based Compression)

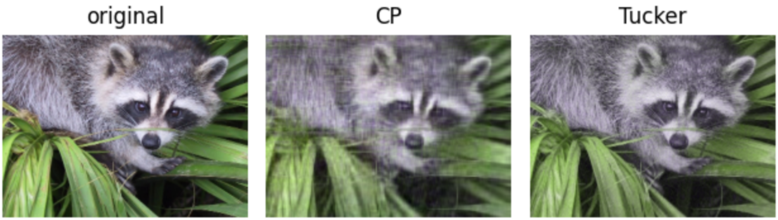


11

[11] Credit: Kolda, Michigan Institute for Data Science (MIDAS) Seminar

Example (Tensor Decomposition in Image Compression)

Source:*http* :
//tensorly.org/stable/auto$_e$xamples/decomposition/plot$_i$mage$_c$ompression.html



Example (Tensor completion in Image Compression)

Source: $https://www.mdpi.com/2076-3417/10/3/797/htm$

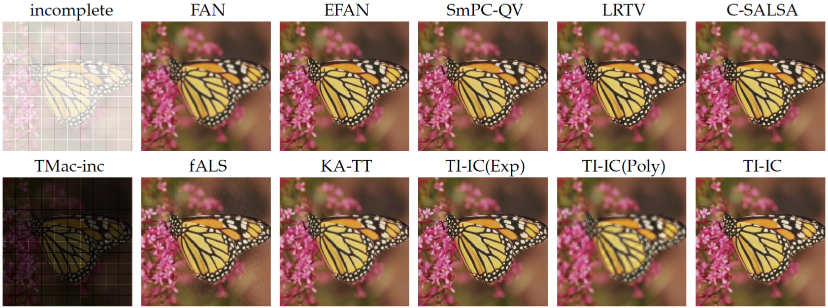


Figure 5. Test D: resolution up-scaling for the image "Monarch".

Example (Tucker completion for a cine cardiac MRI series)

Source: *https : //arxiv.org/abs/*1911.10454

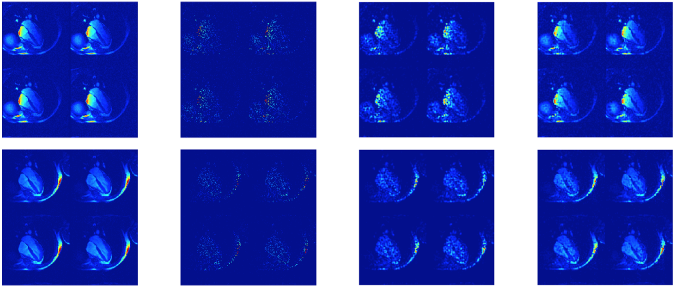
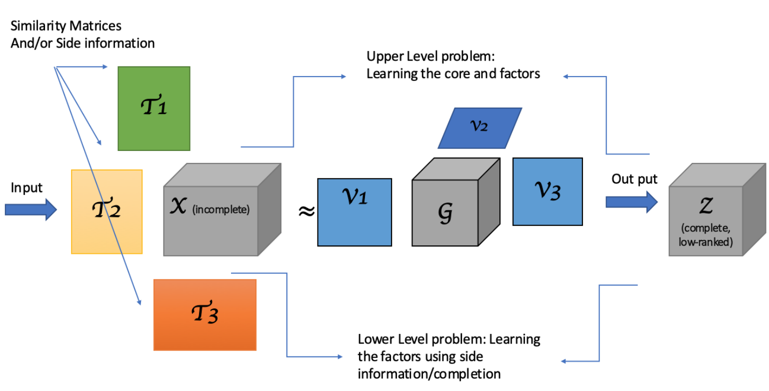


Figure: Tucker and Dual Core completion results (third and fourth colums) for a cine cardiac MRI series $192 \times 192 \times 8 \times 19$ with 85 percent missing rate.

Pitch to the "Demo"

TR-MLC

Simultaneous tensor recovery and reconstruction using metric learning constraints (TR-MLC): more details during the "demo".



References

References: I

0 . Conrad, Keith. "Tensor products." Notes of course, available on-line (2018).

1 . Kolda, Tamara G., and Brett W. Bader. "Tensor decompositions and applications." SIAM review 51, no. 3 (2009): 455-500. (highly recommended)

2 . Merris, Russell. “Multilinear algebra”. Crc Press, 1997.

3 . Brett W. Bader, Tamara G. Kolda and others. MATLAB Tensor Toolbox Version 2.6, Available online, February 2015

4 . Ji, Yuwang, Qiang Wang, Xuan Li, and Jie Liu. "A survey on tensor techniques and applications in machine learning." IEEE Access 7 (2019): 162950-162990.

5 . Lu, Haiping, Konstantinos N. Plataniotis, and Anastasios Venetsanopoulos. Multilinear subspace learning: dimensionality reduction of multidimensional data. CRC press, 2013.

6 .Song, Qingquan, Hancheng Ge, James Caverlee, and Xia Hu. "Tensor completion algorithms in big data analytics." ACM Transactions on Knowledge Discovery from Data (TKDD) 13, no. 1 (2019): 1-48.

7 . Hale, Elizabeth, and Ashley Prater-Bennette. "Comparison of CP and Tucker tensor decomposition algorithms." In Big Data III: Learning, Analytics, and Applications, vol. 11730, p. 117300D. International Society for Optics and Photonics, 2021.

8 . Bagherian, Maryam, Renaid B. Kim, Cheng Jiang, Maureen A. Sartor, Harm Derksen, and Kayvan Najarian. "Coupled matrix–matrix and coupled tensor–matrix completion methods for predicting drug–target interactions." Briefings in bioinformatics 22, no. 2 (2021): 2161-2171.

9 . Suárez-Díaz, Juan Luis, Salvador García, and Francisco Herrera. "A tutorial on distance metric learning: Mathematical foundations, algorithms, experimental analysis, prospects and challenges (with appendices on mathematical background and detailed algorithms explanation)." arXiv preprint arXiv:1812.05944 (2018).

Thank you!