Tensor (Multiway arrays) Decomposition and Application

Authors: Tamara G. Kolda and Brett W. Bader

Presented by Yao Zhang

SIAM Review, 2009

Overview

1. Introduction
2. Notation and Preliminaries
3. CP Decomposition
4. Tucker Decomposition
5. Application

Introduction

A tensor is multidimensional array. More formally, an $N-$way or $Nth-$order tensor is a element of the tensor product of N vector space, each of which has its own coordinate system. This notion of tensors is not to be confused with tensors in physics and engineering (such as stress tensors), which are generally referred to as tensor fields in mathematics. A third-order tensor has three indices. A first-order tensor is a vector, a second tensor is a matrix, and tensor of order three or higher are called higher-order tensor.

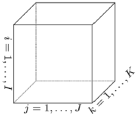

Figure: A third-order tensor:$\mathcal{X} \in \mathbb{R}^{I \times J \times K}$

Introduction

Both the CANDECOMP/PARAFAC (CP) and Tucker tensor decompositions can be considered to be higher-order generation of the matrix singular value decomposition (SVD) and principal component analysis (PCA). In this paper, we discuss the CP decomposition, its connection to tensor rank and tensor border rank, conditions for uniqueness, algorithms and computational issues, and applications, we also discuss the Tucker decomposition, its relationship to compression, the notion of $n-$rank, algorithms and computational issues, and application.

Fibers

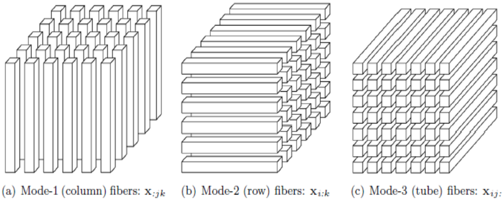

Figure: Fibers of a $3rd-$order tensor

mode-1: fix j, k; mode-2: fix i, k ; mode-3: fix i, j.

Slides

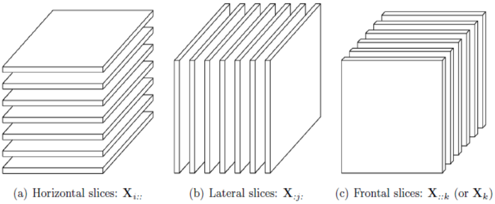

Figure: Slides of a $3rd-$order tensor

Horizontal: fix i; Lateral: fix j ; Frontal: fix k

Norm and Inner Product

Given two tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$.

The norm of $\mathcal{X}$

$$\|\mathcal{X}\| = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} x_{i_1}^2 x_{i_2}^2 \cdots x_{i_N}^N}$$

The inner product of $\mathcal{X}$ and $\mathcal{Y}$

$$\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \cdots i_N} y_{i_1 i_2 \cdots i_N}$$

It is obviously that

$$\langle \mathcal{X}, \mathcal{X} \rangle = \|\mathcal{X}\|^2$$

Rank-One Tensor

An $N-$way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is rank one if it can be written as the outer product of N vectors, i.e.

$$\mathcal{X} = \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \cdots \circ \mathbf{a}^{(N)}$$

The symbol $\circ$ represents the vector outer product. This means that each element of the tensor is product of the corresponding vector elements:

$$x_{i_1\ i_2\ \ldots\ i_N} = a^{(1)}_{i_1} a^{(2)}_{i_2} \cdots a^{(N)}_{i_N} \quad \textit{for all}\ \ 1 \leqslant i_n \leqslant I_n$$

Rank-One Tensor

Does it make sense to a given matrix $X \in \mathbb{R}^{m \times n}$? if $rank(X) = 1$ then we can find two vectors $\mathbf{u} \in \mathbb{R}^{m \times 1}$, $\mathbf{v} \in \mathbb{R}^{n \times 1}$, i.e. $\mathbf{u}$ and $\mathbf{v}$ are all column vector and $rank = 1$.

$$X = \mathbf{u}\mathbf{v}^T = \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} \begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix} = \begin{bmatrix} u_1 v_1 & \cdots & u_1 v_n \\ \vdots & \ddots & \vdots \\ u_m v_1 & \cdots & u_m v_n \end{bmatrix} = \mathbf{u} \circ \mathbf{v}$$

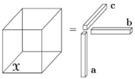

Figure: Rank-one third-order tensor, $\mathcal{X} = \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$. The (i,j,k) element of $\mathcal{X}$ is given by $x_{ijk} = a_i b_j c_k$.

Symmetry and Tensors

A tensor is called cubical if every mode is the same size, i.e. $\mathcal{X} = \mathbb{R}^{I \times I \times I \times I \cdots \times I}$. A cubical tensor is called supersymmetric if its elements remain constant under any permutation of the indices. For instance, a three-way tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times I}$ is supersymmetric if

$$x_{ijk} = x_{ikj} = x_{jik} = x_{jki} = x_{kij} = x_{kji} \quad for \;\; all \;\; i, j, k \; = \; 1, \ldots, I.$$

Tensors can be (partially) symmetric in two or more modes as well. For example, a three-way tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times K}$ is symmetric in modes one and two if all its frontal slices are symmetric, i.e.,

$$X_k = X_k^T \quad for \;\; all \; k \; = 1, \ldots, K$$

Diagonal Tensors

A tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times \cdots \times I_N}$ is diagonal if $x_{i_1 i_2 \ldots i_N} \neq 0$ if $i_1 = i_2 = \cdots = i_N$.

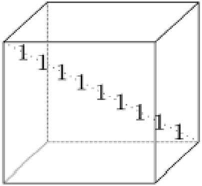

Figure: Three-way tensor of size of size $I \times I \times I$ with ones along the superdiagonal.

Matricization: Transforming a Tensor into a Matrix

The mode-n matricization of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is denote by X_n and arranges the mode-n fibers to be the columns of the resulting matrix. Though conceptually simple, the formal notation is clunky. Tensor element $(i_1, i_2, \ldots, i_N)$ maps to matrix element (i_n, j), where

$$j = 1 + \sum_{k=1, k \neq n}^{N} (i_k - 1) J_K \quad with \quad J_k = \prod_{m=1, m \neq n}^{k-1} I_m \tag{1}$$

Remark

1. *It is very beautiful, rather than clunky*
2. *It is a special case of mode-n matrixcization relevant to CP and Turker decomposition.*

Matricization: Transforming a Tensor into a Matrix

Why formula.1 looks like this?

Consider a map:

$$\begin{bmatrix} i_1 \\ \vdots \\ i_{n-1} \\ i_n \\ i_{n+1} \\ \vdots \\ i_N \end{bmatrix} \rightarrow \begin{bmatrix} i_n \\ i_1 \\ \vdots \\ i_{n-1} \\ i_{n+1} \\ \vdots \\ i_N \end{bmatrix} \rightarrow \begin{bmatrix} i_n \\ i_N \\ \vdots \\ i_{n+1} \\ i_{n-1} \\ \vdots \\ i_1 \end{bmatrix}$$

Matricization: A simple example

The concept is easier to understand using an example. Let the frontal slice of $\mathcal{X} \in \mathbb{R}^{3\times4\times2}$ (i.e. $I_1 = 3, I_2 = 4, I_3 = 2$) be

$$X_1 = \begin{bmatrix} 1 & 4 & 7 & 11 \\ 2 & 5 & 8 & 12 \\ 3 & 6 & 9 & 12 \end{bmatrix} \qquad X_2 = \begin{bmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{bmatrix}$$

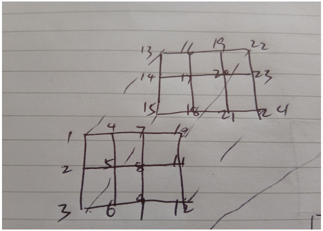

Figure: The third-order tensor: $\mathcal{X} \in \mathbb{R}^{3\times4\times2}$.

Matricization: A simple example

$$X_{(1)} = \left[\begin{matrix} \\ (\textit{row } 1) \\ (\textit{row } 2) \\ (\textit{row } 3) \end{matrix} \underbrace{\begin{matrix} \textit{col } 1 & \textit{col } 2 & \textit{col } 3 & \textit{col } 4 \\ 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{matrix}}_{\textit{block}1} \underbrace{\begin{matrix} \textit{col } 1 & \textit{col } 2 & \textit{col } 3 & \textit{col } 4 \\ 13 & 16 & 19 & 1 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{matrix}}_{\textit{block}2} \right]$$

$$X_{(2)} = \left[\begin{matrix} \\ (\textit{row } 1) \\ (\textit{row } 2) \\ (\textit{row } 3) \\ (\textit{row } 4) \end{matrix} \underbrace{\begin{matrix} \textit{col } 1 & \textit{col } 2 & \textit{col } 3 \\ 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{matrix}}_{\textit{block } 1} \underbrace{\begin{matrix} \textit{col } 1 & \textit{col } 2 & \textit{col } 3 \\ 13 & 14 & 15 \\ 16 & 17 & 18 \\ 19 & 20 & 21 \\ 22 & 23 & 24 \end{matrix}}_{\textit{block } 2} \right]$$

$$X_{(3)} = \left[\begin{matrix} \\ (\textit{row } 1) \\ (\textit{row } 2) \end{matrix} \underbrace{\begin{matrix} \textit{col } 1 & \textit{col } 2 & \textit{col}3 \\ 1 & 2 & 3 \\ 13 & 14 & 15 \end{matrix}}_{\textit{block } 1} \underbrace{\begin{matrix} \textit{col } 1 & \textit{col } 2 & \textit{col } 3 \\ 4 & 5 & 6 \\ 16 & 17 & 18 \end{matrix}}_{\textit{block } 2} \underbrace{\begin{matrix} \textit{col } 1 & \textit{col } 2 & \textit{col } 3 \\ 7 & 8 & 9 \\ 19 & 20 & 21 \end{matrix}}_{\textit{block } 3} \underbrace{\begin{matrix} \textit{col } 1 & \textit{col } 2 & \textit{col } 3 \\ 10 & 11 & 12 \\ 22 & 23 & 24 \end{matrix}}_{\textit{block } 4} \right]$$

Matricization: the jth column

Table: Some patterns when we do the transforming a tensor into a matrix

numbers	rows	blocks	columns
$X_{(1)}$	I_1	I_3	I_2
$X_{(2)}$	I_2	I_3	I_1
$X_{(3)}$	I_3	I_2	I_1

$$
\begin{aligned}
j = \quad & (i_N - 1)\, I_{N-1} I_{N-2} \cdots I_{n+1} I_{n-1} \cdots I_2 I_1 + \\
& (i_{N-1} - 1)\, I_{N-2} \cdots I_{n+1} I_{n-1} \cdots I_2 I_1 + \\
& \cdots + \\
& (i_{n+1} - 1)\, I_{n-1} \cdots I_2 I_1 + \\
& (i_{n-1} - 1)\, I_{n-2} \cdots I_2 I_1 + \\
& \cdots + \\
& (i_2 - 1)\, I_1 + \\
& i_1
\end{aligned}
$$

Matricization: different maps

Backward cyclic:

$$\begin{bmatrix} i_1 \\ \vdots \\ i_{n-1} \\ i_n \\ i_{n+1} \\ \vdots \\ i_N \end{bmatrix} \to \begin{bmatrix} i_n \\ i_1 \\ \vdots \\ i_{n-1} \\ i_{n+1} \\ \vdots \\ i_N \end{bmatrix} \to \begin{bmatrix} i_n \\ i_n - 1 \\ \vdots \\ i_1 \\ i_N \\ \vdots \\ i_(n+1) \end{bmatrix}$$

Forward cyclic:

$$\begin{bmatrix} i_1 \\ \vdots \\ i_{n-1} \\ i_n \\ i_{n+1} \\ \vdots \\ i_N \end{bmatrix} \to \begin{bmatrix} i_n \\ i_1 \\ \vdots \\ i_{n-1} \\ i_{n+1} \\ \vdots \\ i_N \end{bmatrix} \to \begin{bmatrix} i_n \\ i_n + 1 \\ \vdots \\ i_N \\ i_1 \\ \vdots \\ i_(n-1) \end{bmatrix}$$

The $n-$mode Product (Dimensionality Reduction)

The $n-$mode matrix product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a matrix $U \in \mathbb{R}^{J \times I_n}$ is denoted by $\mathcal{X} \times_n U$ and is of size $I_1 \times \cdots I_{n-1} \times J \times I_{n+1} \times \cdots I_N$.

Remark

If $j < I_n$, then it is the dimensionality reduction.

Elementwise, we have

$$(\mathcal{X} \times_n U)_{i_1 \ldots i_{n-1} j i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_N} u_{j i_n}$$

The $n-$mode Product (Dimensionality Reduction)

The $n-$mode vector product of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a matrix $\mathbf{v} \in \mathbb{R}^{I_n}$ is denoted by $\mathcal{X} \bar{\times}_n \mathbf{v}$ and is of size $I_1 \times \cdots I_{n-1} \times I_{n+1} \times \cdots I_N$., the order is $N-1$. Elementwise, we have

$$(\mathcal{X} \bar{\times}_n \mathbf{v})_{i_1 \ldots i_{n-1} i_{n+1} \ldots i_N} = \sum_{i_n=1}^{I_n} x_{i_1 i_2 \ldots i_N} v_{i_n}$$

Some properties of the $n-$mode product

$$\mathcal{Y} = \mathcal{X} \times_n U \Leftrightarrow Y_{(n)} = UX_{(n)}$$

$$\mathcal{X} \times_m A \times_n B = \mathcal{X} \times_n B \times_m A \quad (m \neq n)$$

$$\mathcal{X} \times_n A \times_n B = \mathcal{X} \times_n (BA) \quad (m = n)$$

$$\mathcal{X} \,\overline{\times}_m\, \mathbf{a} \,\overline{\times}_n\, \mathbf{b} = (\mathcal{X} \,\overline{\times}_m\, \mathbf{a}) \;\; \overline{\times}_{n-1} \; \mathbf{b} = \; \mathcal{X} \;\; \overline{\times}_n\, \mathbf{b} \,\overline{\times}_m\, \mathbf{a} \quad m < n$$

Kronecker Product

The Kronecker product of matrices $A \in \mathbb{R}^{I\times J}$ and $B \in \mathbb{R}^{K\times L}$ is denoted by $A \otimes B$, and define as flowing:

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

$$= \underbrace{\begin{bmatrix} \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1L} \\ a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{K1} & a_{11}b_{K2} & \cdots & a_{11}b_{KL} \end{bmatrix} & a_{12}B & \cdots & a_{2J}B \\ a_{21}B & a_{22}B & \cdots & a_{2J}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}B & a_{I2}B & \cdots & a_{IJ}B \end{bmatrix}}_{Column:\quad L\times J} \quad row:\quad K\times I$$

$$= \begin{bmatrix} \underbrace{\mathbf{a}_1 \otimes \mathbf{b}_1 \;\; \mathbf{a}_1 \otimes \mathbf{b}_2 \cdots \mathbf{a}_1 \otimes \mathbf{b}_L}_{\in \;\; \mathbb{R}^{IK\times L}} \cdots\cdots \underbrace{\mathbf{a}_J \otimes \mathbf{b}_1 \;\; \mathbf{a}_J \otimes \mathbf{b}_2 \cdots \mathbf{a}_J \otimes \mathbf{b}_L}_{\in \;\; \mathbb{R}^{IK\times L}} \end{bmatrix} \in \mathbb{R}^{IK\times JL}$$

It is obviously that $A \otimes B \neq B \otimes A$.

Kronecker-Rao Product

The Kronecker-Rao product of matrices $A \in \mathbb{R}^{I \times K}$ and $B \in \mathbb{R}^{J \times K}$ (A, B matching columnwise) is denoted by $A \odot B$, and define as flowing:

$$A \odot B = \left[\left(\underbrace{\mathbf{a}_1 \otimes \mathbf{b}_1}_{\in \mathbb{R}^{IJ \times 1}} \right), (\mathbf{a}_2 \otimes \mathbf{b}_2) \cdots, (\mathbf{a}_k \otimes \mathbf{b}_k) \right] \in R^{IJ \times K}$$

It is obviously that $A \odot B \neq B \odot A$.

Hadamard Product

The Hadamard product is denoted by $A * B$, $\left(A, B \in \mathbb{R}^{I \times J}\right)$ and define as flowing:

$$A * B = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1J}b_{1J} \\ a_{21}b_{21} & a_{22}b_{22} & \cdots & a_{2J}b_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}b_{I1} & a_{I2}b_{I2} & \cdots & a_{IJ}b_{IJ} \end{bmatrix} \in \mathbb{R}^{I \times J}$$

Some properties

Some useful properties:

$$(A \otimes B)(C \otimes D) = AC \otimes BD$$

$$(A \otimes B)^+ = A^+ \otimes B^+$$

$$A \odot B \odot C = (A \odot B) \odot C = A \odot (B \odot C)$$

$$(A \odot B)^T (A \odot B) = A^T A * B^T B$$

$$(A \odot B)^+ = \left((A^T A) * (B^T B)\right)^+ (A \odot B)^T$$

where A^+ denotes the Moore-Penrose pseudoinverse of A.

Some properties

Let $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \cdots I_N}$ and $A^{(n)} \in \mathbb{R}^{J_n \times I_n}$ for all $n \in \{1, .., N\}$ then

$$\mathcal{Y} = \ \mathcal{X} \times_1 A^{(1)} \times_2 A^{(2)} \cdots \times_N A^{(\mathrm{N})} \Leftrightarrow$$

$$Y_{(n)} = A^{(n)} X_{(n)} \left(A^{(N)} \otimes \cdots \otimes A^{(n+1)} \otimes A^{(n-1)} \otimes \cdots \otimes A^{(1)} \right)^T$$

CANDECOMP/PARAFAC Decomposition

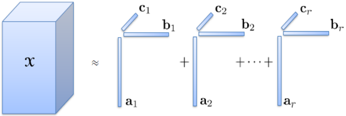

Figure: CP Decomposition

$$\min_{A,B,C} \|\mathcal{X} - \mathcal{M}\|^2 \; s.t. \;\; \mathcal{M} = [A, B, C] \Leftrightarrow \min_{A,B,C} \sum_{i,j,k} \left(x_{ijk} - \sum_{l} a_{il} b_{jl} c_{kl} \right)^2$$

unfortunately, it is nonconvex, but its subproblems are convex.

CP Decomposition

Repeat until convergence

1. $$\min_{A} \sum_{i,j,k} \left(x_{ijk} - \sum_{l} a_{il} b_{jl} c_{kl} \right)^2$$

2. $$\min_{B} \sum_{i,j,k} \left(x_{ijk} - \sum_{l} a_{il} b_{jl} c_{kl} \right)^2$$

3. $$\min_{C} \sum_{i,j,k} \left(x_{ijk} - \sum_{l} a_{il} b_{jl} c_{kl} \right)^2$$

What we do in the Iteration

$$\min_A \sum_{i,j,k} \left(x_{ijk} - \sum_l a_{il} b_{jl} c_{kl} \right)^2 \Leftrightarrow \min_A \left\| X_{(1)} - A(C \odot B)^T \right\|_F^2$$

$$\min_B \sum_{i,j,k} \left(x_{ijk} - \sum_l a_{il} b_{jl} c_{kl} \right)^2 \Leftrightarrow \min_B \left\| X_{(2)} - B(C \odot A)^T \right\|_F^2$$

$$\min_C \sum_{i,j,k} \left(x_{ijk} - \sum_l a_{il} b_{jl} c_{kl} \right)^2 \Leftrightarrow \min_C \left\| X_{(3)} - C(B \odot A)^T \right\|_F^2$$

Tucker Decomposition

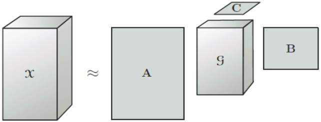

Figure: Tucker Decomposition

$$\min_{\mathcal{G},A,B,C} \|\mathcal{X} - \mathcal{M}\|^2 \ s.t. \ \ \mathcal{M} = [\![\mathcal{G}, A, B, C]\!]$$

$$\text{where } m_{ijk} = \sum_{r_1} \sum_{r_2} \sum_{r_3} g_{r_1 r_2 r_3} a_{ir_1} b_{jr_2} c_{kr_3}$$

Tucker Decomposition VS CP Decomposition

Given a third-order tensor $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$.
$A = [\mathbf{a}_1, ..., \mathbf{a}_P], \mathbf{a}_i \in \mathbb{R}^{I\times 1}, A \in \mathbb{R}^{I\times P}, B = [\mathbf{b}_1, ..., \mathbf{b}_Q], \mathbf{b}_j \in \mathbb{R}^{J\times 1}, B \in \mathbb{R}^{J\times Q}; C = [\mathbf{c}_1, ..., \mathbf{c}_R], \mathbf{c}_k \in \mathbb{R}^{K\times 1}, C \in \mathbb{R}^{K\times R}, \mathcal{G} \in \mathbb{R}^{P\times Q\times R}$. $\mathbf{a}_r \in \mathbb{R}^I, \mathbf{b}_r \in \mathbb{R}^J$ and $\mathbf{c}_r \in \mathbb{R}^K$.

Tucker Decomposition:

$$\mathcal{X} \approx \mathcal{G} \times_1 A \times_2 B \times_3 C = \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} g_{pqr}\, \mathbf{a}_p \circ \mathbf{b}_q \circ \mathbf{c}_r$$

$$x_{ijk} = \sum_{p=1}^{P}\sum_{q=1}^{Q}\sum_{r=1}^{R} g_{pqr} a_{ip} b_{jq} c_{kr} \ \ \textit{for}\ i = 1, ..., I;\ j = 1, ..., J;\ k = 1, ..., K$$

CP decomposition:

$$\mathcal{X} \approx \sum_{i=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$$

$$x_{ijk} \approx \sum_{r=1}^{R} a_{ir} b_{jr} c_{jr} \quad \textit{for}\ i\ = 1, ..., I;\ j\ = 1, ..., J, ;\ k\ = 1, ..., K$$

Orthogonal Constraint Problem

Consider the problem of projecting a given matrix $M \in \mathbb{R}^{n \times n}$ onto the space of orthogonal matrices $O(n) \left\{ U \in \mathbb{R}^{n \times n} : U^T U = I \right\}$. That is, we want to find a matrix $U \in \mathbb{R}^{n \times n}$ that minimizes

$$\min_{U} \|M - U\|_F^2 \quad s.t. \;\; U^T U = I \tag{2}$$

Notice that there are n^2 constraints in $U^T U = I$. This suggests using n^2 Lagrange multipliers, which can be conveniently represented as the entries of a matrix $\Lambda \in \mathbb{R}^{n \times n}$. However, since the matrix $U^T U$ is symmetric, there are only $\dfrac{n(n+1)}{2}$ independent constrains. Therefore, the matrix Λ needs to be chosen to be be symmetric.

Orthogonal Constraint Problem

And the Lagrangian function for Eq.2 can be written as

$$\mathcal{L}(U, \Lambda) = \|M - U\|_F^2 + \langle \Lambda, U^T U - I \rangle$$

Then, we can get

$$\frac{\partial \mathcal{L}}{\partial U} = 0 \Rightarrow (U - M) + U\Lambda = 0 \Rightarrow \Lambda = U^T M - I$$

Since Λ is symmetric, so is $U^T M$. Let $M = W \sum V^T$ be the singular value decomposition of M. Both W, V are orthogonal matrices. If the singular values of M are all different, then in oder for $U^T M = U^T W \Sigma V^T$ to be symmetric, we must have $U^T W = V$, hence $U = WV^T$.

HOSVD

Algorithm 1 HOSVD

procedure $\mathrm{HOSVD}(\mathcal{X}, R_1, R_2, ..., R_N)$
  for $n = 1, ..., N$ **do**
    $A^{(n)} \leftarrow R_n$ *leading left sigular vectors of* $X_{(n)}$
  end for
  $\mathcal{G} \leftarrow \mathcal{X} \times_1 (A^{(1)})^T \times_2 (A^{(2)})^T ... \times_N (A^{(N)})^T$
  Return $\mathcal{G}, A^{(1)}, A^{(2)}, ..., A^{(N)}$
end procedure

HOOI

Algorithm 2 HOOI

procedure $\mathrm{HOOI}(\mathcal{X}, R_1, R_2, ..., R_N)$

  initialize $A^n \in \mathbb{R}^{I_n \times R}$ for $n = 1, ..., N$ using HOSVD

  Repeat

  for $n = 1, ..., N$ **do**

    $\mathcal{Y} \leftarrow \mathcal{X} \times_1 (A^{(1)})^T ... \times_n -1 (A^{(n-1)})^T \times_n +1 (A^{(n+1)})^T ... \times_N (A^{(N)})^T$

    $A^{(n)} \leftarrow R_n$ *leading left sigular vectors of* $Y_{(n)}$

  end for

  until fit crease to improve or maximum iteration exhausted

  $\mathcal{G} \leftarrow \mathcal{X} \times_1 (A^{(1)})^T \times_2 (A^{(2)})^T ... \times_N (A^{(N)})^T$

  Return $\mathcal{G}, A^{(1)}, A^{(2)}, ..., A^{(N)}$

end procedure

Tensor RPCA [Cao 2016]

$$\min_{\mathcal{X},\mathcal{S},\mathcal{E},\mathcal{G},U_1,U_2,U_3} \lambda\|\mathcal{S}\|_{3D-TV} + \frac{1}{2}\|\mathcal{E}\|_F^2$$

$$\begin{aligned} s.t. \quad & \mathcal{X} = \mathcal{L} + \mathcal{S} + \mathcal{E} \\ & \mathcal{L} = \mathcal{G} \times_1 U_1 \times_2 U_2 \times U_3 \\ & y = \mathcal{A}(\mathcal{X}) \end{aligned}$$

where the factor matrix U_1 and U_2 are orthogonal in columns for two spatial modes, the factor matrix U_3 is orthogonal in columns for the temporal mode, the core tensor $\mathcal{G}$ interacts with these factors and 3D-TV term $\|\cdot\|_{3D-TV}$ is defined as

$$\|\mathcal{X}\|_{3D-TV} = \|\mathcal{X}_h(i,j,k)\|_1 + \|\mathcal{X}_v(i,j,k)\|_1 + \|\mathcal{X}_t(i,j,k)\|_1$$

where

$$\begin{aligned} \|X_h(i,j,k)\|_1 &= X(i,j+1,k) - X(i,j,k) \\ \|X_v(i,j,k)\|_1 &= X(i+1,j,k) - X(i,j,k) \\ \|X_t(i,j,k)\|_1 &= X(i,j,k+1) - X(i,j,k) \end{aligned}$$

References

T. Kolda, B. Bader (2009)
Tensor Decompositions and Applications
SIAM Review.

Wenfei Cao, Yao Wang, Jian Sun, Deyu Meng, Can Yang, Andrzej Cichocki, Zongben Xu (2016)
Total Variation Regularized Tensor RPCA for Background Subtraction from Compressive Measurements
IEEE Transactions on Image Processing.

Recent publications of Kolda

- E. Phipps, T. Kolda (2019)
  Software for Sparse Tensor Decomposition on Emerging Computing Architectures
  SIAM Journal on Scientific Computing.

- D. Hong, T. Kolda, J. Duersch (2019)
  Generalized Canonical Polyadic Tensor Decomposition
  SIAM Review.

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

- C. Anderson-Bergman, T. Kolda, K. Kincher-Winoto (2018)
  TuckerMPI: XPCA: Extending PCA for a Combination of Discrete and Continuous Variables
  arXiv.

Recent publications of Kolda

C. Battaglino, G. Ballard, T. Kolda (2018)
A Practical Randomized CP Tensor Decomposition
SIAM Journal on Matrix Analysis and Applications.

A. Williams, T. Kim, F. Wang, S. Vyas, S. Ryu, K. Shenoy, M. Schnitzer, T. Kolda, S. Ganguli (2018)
Unsupervised Discovery of Demixed, Low-dimensional Neural Dynamics across Multiple Timescales through Tensor Components Analysis
Neuron.

S. Mohammadi, D. Gleich, T. Kolda, A. Grama (2017)
Triangular Alignment (TAME): A Tensor-based Approach for Higher-order Network Alignment
IEEE/ACM Transactions on Computational Biology and Bioinformatics.

Thank you!