APPLICATION OF CP DECOMPOSITION

By Ranjan Sarkar
(21MA40022)

Under the supervision
of
Prof. Swanand R. Khare

Date:
May 3, 2023

Department of Mathematics
IIT Kharagpur

CONTENTS

- Introduction
- Tensor Operations
- CP Decomposition
- Alternative Least Square
- Image Compression

Introduction

Tensor

A **Tensor** is a multi-dimensional array, a general form of a Matrix (which is a 2-dim array). More formally, an N-way or Nth-order tensor is an element of the tensor product of N vector spaces, each of which has its own coordinate system.

- A first-order tensor is a vector,
- a second-order tensor is a matrix,
- and tensors of order 3 or higher are called higher-order tensors.

1-dim array - $(1, 2, 3) - \mathbf{a}$ (Vector)

2 -dim array - $\begin{pmatrix} 3 & 3 & 9 \\ 3 & 3 & 2 \end{pmatrix} - \mathbf{A}(2 \times 3 \text{ Matrix })$

3-way array - $\left(\begin{pmatrix} 2 & 8 & 2 & 9 \\ 3 & 0 & 1 & 0 \end{pmatrix}, \begin{pmatrix} 6 & 8 & 1 & 1 \\ 2 & 2 & 3 & 3 \end{pmatrix}\right) - \boldsymbol{\chi}(2 \times 2 \times 4 \text{ Tensor })$

Introduction

Norm

The **norm of a tensor** $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is the square root of the sum of the squares of all its elements, i.e.,

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

Rank 1 Matrix

The *rank* of this matrix is **1**. So we can write it as a product of two vectors or simply if we take a ***product of a column with a row*** we will get ***Rank–One Matrix.***

$$\begin{bmatrix} 4 & 6 \\ -6 & -9 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 2 \\ -3 \\ 1 \end{bmatrix} \begin{bmatrix} 2 & 3 \end{bmatrix}$$

Introduction

Rank 1 Tensor

An N-way tensor $\mathfrak{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ 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 "o" represents the vector outer product. This means that each element of the tensor is the product of the corresponding vector elements:

$$x_{i_1 i_2 \cdots i_N} = a^{(1)}_{i_1} a^{(2)}_{i_2} \cdots a^{(N)}_{i_N} \text{ for all } 1 \leq i_n \leq I_n$$

- ✓ Example of 3rd order Rank-One Tensor -

$$\begin{aligned} X &= \begin{pmatrix} 1 & 2 \end{pmatrix} \circ \begin{pmatrix} -1 & 1 \end{pmatrix} \circ \begin{pmatrix} 3 & 2 \end{pmatrix} \\ &= \left[\begin{bmatrix} -3 & -2 \\ 3 & 2 \end{bmatrix}, \begin{bmatrix} -6 & -4 \\ 6 & 4 \end{bmatrix}\right] \end{aligned}$$

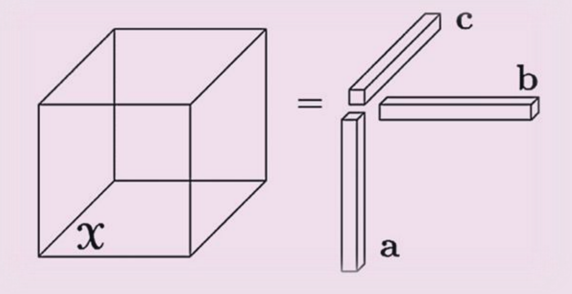


Figure 1. A third-order Rank 1 Tensor (Reprinted)

Tensor Operations

Matricization

Matricization, also known as **unfolding** or **flattening**, is the process of reordering the elements of an N-way array into a matrix.

For instance, a **2 × 3 × 4 tensor** can be arranged as a **6 × 4 matrix** or a **3 × 8 matrix**, and so on. The mode–n matricization of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is denoted by $\mathbf{X}_{(n)}$ and arranges the mode-n fibers to the columns of the resulting matrix.

Tensor element $(i_1, i_2, \ldots, i_N)$ maps to matrix element (i_n, j) where,

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

✓ Example:

$$\mathcal{X} = \left[\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}, \begin{bmatrix} 13 & 14 & 15 & 16 \\ 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 \end{bmatrix} \right]$$

The 3 mode n unfoldings are

$$\mathbf{X}_{(1)} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 \end{bmatrix}$$

$$\mathbf{X}_{(2)} = \begin{bmatrix} 1 & 2 & 3 & 4 & 13 & 14 & 15 & 16 \\ 5 & 6 & 7 & 8 & 17 & 18 & 19 & 20 \\ 9 & 10 & 11 & 12 & 21 & 22 & 23 & 24 \end{bmatrix}$$

$$\mathbf{X}_{(3)} = \begin{bmatrix} 1 & 5 & 9 & 13 & 17 & 21 \\ 2 & 6 & 10 & 14 & 18 & 22 \\ 3 & 7 & 11 & 15 & 19 & 23 \\ 4 & 8 & 12 & 16 & 20 & 24 \end{bmatrix}$$

Tensor Operations

Kronecker Product

The Kronecker product of matrices $\mathbf{A} \in \mathbb{R}^{I \times J}$ and $\mathbf{B} \in \mathbb{R}^{K \times L}$ is denoted by $\mathbf{A} \otimes \mathbf{B}$. The result is a matrix of size $(IK) \times (JL)$ and defined by

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1J}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2J}\mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}\mathbf{B} & a_{I2}\mathbf{B} & \cdots & a_{IJ}\mathbf{B} \end{bmatrix}$$

✓ Example:

$$A \in \mathbb{R}^{2\times 2}, B \in \mathbb{R}^{2\times 2}, \text{ then } A \otimes B \in \mathbb{R}^{4\times 4}$$

$$A = \begin{bmatrix} 1 & 2 \\ -1 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 1 & -1 \\ -2 & 3 \end{bmatrix}, \text{ then } A \otimes B = \begin{bmatrix} 1 & -1 & 2 & -2 \\ -2 & 3 & -4 & 6 \\ -1 & 1 & 0 & 0 \\ 2 & -3 & 0 & 0 \end{bmatrix}$$

Tensor Operations

The Khatri-Rao product is the **"matching columnwise" Kronecker product**. Given matrices $\mathbf{A} \in \mathbb{R}^{I \times K}$ and $\mathbf{B} \in \mathbb{R}^{J \times K}$, their Khatri-Rao product is denoted by $\mathbf{A} \odot \mathbf{B}$. The result is a matrix of size $(IJ) \times K$ defined by

✓ Example:

$$A \in \mathbb{R}^{2\times 2}, B \in \mathbb{R}^{3\times 2}, \text{ then } A \odot B \in \mathbb{R}^{6\times 2}$$

$$A = \begin{pmatrix} 2 & 1 \\ -1 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & -2 \\ 2 & 1 \\ 3 & 0 \end{pmatrix}, \text{ then } A \odot B = \begin{pmatrix} 2 & -2 \\ 4 & 1 \\ 6 & 0 \\ -1 & 0 \\ -2 & 0 \\ -3 & 0 \end{pmatrix}$$

CP Decomposition

The **CANDECOMP/PARAFAC (CP) decomposition** factorizes a tensor into a sum of component rank-one tensors. For example, given a third-order tensor

$$\boldsymbol{X} \approx \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r, \qquad (1)$$

where $\boldsymbol{R}$ is a positive integer and $\mathbf{a}_r \in \mathbb{R}^I, \mathbf{b}_r \in \mathbb{R}^J$, and $\mathbf{c}_r \in \mathbb{R}^K$ for $r = 1, \ldots, R$.

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

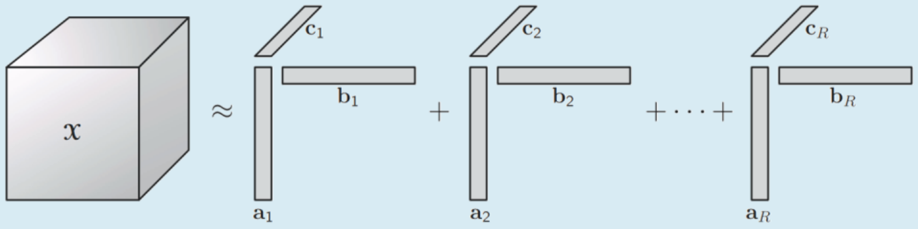


Figure 2: *CP Decomposition of a 3-way tensor* *

The factor matrices refer to the combination of the vectors from the rank-one components, i.e., $\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_R \end{bmatrix}$ and likewise for $\mathbf{B}$ and $\mathbf{C}$. Using these definitions, Eq **(1)** may be written in matricized form:

$$\begin{aligned} \mathbf{X}_{(1)} &\approx \mathbf{A}(\mathbf{B} \odot \mathbf{C})^\top \\ \mathbf{X}_{(2)} &\approx \mathbf{B}(\mathbf{A} \odot \mathbf{C})^\top \\ \mathbf{X}_{(3)} &\approx \mathbf{C}(\mathbf{A} \odot \mathbf{B})^\top \end{aligned} \qquad (2)$$

* Reprinted from given references.

SVD: Singular Value Decomposition

For matrices (2 way tensor), Eckart and Young showed that a best rank-k approximation is given by the leading k factors of the **SVD (Singular Value Decomposition)**. In other words, let R be the rank of a matrix **A** and assume its SVD is given by

$$\mathbf{A} = \sum_{r=1}^{R} \sigma_r \mathbf{u}_r \circ \mathbf{v}_r \quad \text{with } \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_R > 0.$$

Then a ***rank-k approximation*** that minimizes $\|\mathbf{A} - \mathbf{B}\|$ is given by

$$\mathbf{B} = \sum_{r=1}^{k} \sigma_r \mathbf{u}_r \circ \mathbf{v}_r.$$

This type of result does not hold true for higher-order tensors. For instance, consider a third-order tensor of rank R with the following CP decomposition:

$$\boldsymbol{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r.$$

It is possible that the best rank-k approximation may not even exist. The problem is referred to as one of degeneracy. A tensor is degenerate if it may be approximated arbitrarily well by a factorization of lower rank

ALS

OLS: Ordinary Least Square

Suppose that the $\mathbf{m} \times \mathbf{n}$ matrix $\mathbf{A}$ is tall, i.e., there are more equations $(\boldsymbol{m})$ than variables $(\boldsymbol{n})$.
So the system of linear equations,

$$\boldsymbol{Ax} = \boldsymbol{b}$$

These equations have a solution only if $\boldsymbol{b}$ is a linear combination of the columns of $\mathbf{A}$.

- For the approximate solution,
  we would find such $\boldsymbol{x}$ so that the residual $\boldsymbol{r} = (\boldsymbol{Ax} - \boldsymbol{b})$ as small as possible.
- Similar to, Minimize the norm of $\boldsymbol{r}$ i.e. $\boldsymbol{Minimize} \parallel \boldsymbol{Ax} - \boldsymbol{b} \parallel$
- Same as, $\boldsymbol{Minimize} \parallel \boldsymbol{Ax} - \boldsymbol{b} \parallel^2$

Solution of Least Square

From calculus, any minimizer $\hat{x}$ of the function $f(x) = \|Ax - b\|^2$ must satisfy,

$$\nabla f(\hat{x}) = 2A^T(A\hat{x} - b) = 0,$$

$$\boxed{\begin{array}{ll} \text{Solution:} & \hat{x} = \left(A^T A\right)^{-1} A^T b \\ & \hat{x} = A^{\dagger} b \end{array}}$$

$A^{\dagger}$ is called ***Pseudo-inverse*** of $\boldsymbol{A}$

ALS

Matrix Least Square

Suppose that the $\mathbf{m} \times \mathbf{n}$ matrix $\mathbf{A}$ is tall, i.e., there are more equations (m) than variables (n).
So the system,

$$\boldsymbol{AX} = \boldsymbol{B}$$

These equations have a solution only if the columns of $\boldsymbol{B}$ is a linear combination of the columns of $\mathbf{A}$.

- For the approximate solution,
  we would find such $\boldsymbol{x}$ so that the residual $\boldsymbol{R} = (\boldsymbol{AX} - \boldsymbol{B})$ as small as possible.
- Similar to, Minimize the norm of $\boldsymbol{R}$ i.e. $\boldsymbol{Minimize} \parallel \boldsymbol{AX} - \boldsymbol{B} \parallel$
- Same as, $\boldsymbol{Minimize} \parallel \boldsymbol{AX} - \boldsymbol{B} \parallel^2$

Solution of Matrix Least Square

The matrix least squares problem is in fact nothing but a set of k ordinary least squares problems.

$$\boxed{\text{Solution:} \quad \hat{X} = A^{\dagger} B}$$

$A^{\dagger}$ is called ***Pseudo-inverse*** of $\boldsymbol{A}$

ALS

Method to find factors of CP Decomposition

For a 3-way data (say, $\mathbf{X}$), we have 3 types of matricization techniques and thus we will get 3 matrix $\boldsymbol{X}_{(1)}, \boldsymbol{X}_{(2)}$ and $\boldsymbol{X}_{(3)}$. suppose our approximated Tensor is going to be $\mathbf{A}$ (we will start with a random initial $\mathbf{A}$)

$$\begin{aligned}
\mathbf{X}_{(1)} &\approx \mathbf{A_1}(\mathbf{A_2} \odot \mathbf{A_3})^\top \\
\mathbf{X}_{(2)} &\approx \mathbf{A_2}(\mathbf{A_1} \odot \mathbf{A_3})^\top \\
\mathbf{X}_{(3)} &\approx \mathbf{A_3}(\mathbf{A_1} \odot \mathbf{A_2})^\top
\end{aligned} \tag{3}$$

Now, we have 3 Least Square Minimization Problem,

$$\begin{aligned}
&\text{Min } \|\mathbf{A_1}(\mathbf{A_2} \odot \mathbf{A_3})^\top - \mathbf{X}_{(1)}\| \\
&\text{Min } \|\mathbf{A_2}(\mathbf{A_1} \odot \mathbf{A_3})^\top - \mathbf{X}_{(2)}\| \\
&\text{Min } \|\mathbf{A_3}(\mathbf{A_1} \odot \mathbf{A_2})^\top - \mathbf{X}_{(3)}\|
\end{aligned} \tag{4}$$

The first one of these minimization problem can be written,

$$\text{Min } \|(\mathbf{A_2} \odot \mathbf{A_3})\mathbf{A_1}^\top - \mathbf{X}_{(1)}^\top\| \tag{5}$$

ALS

Method to find factors of CP Decomposition

Here, we will fix the co-efficient of $A_1^\top$ i.e. $(\mathbf{A_2} \odot \mathbf{A_3}) = \mathbf{B}_1$ (say).

$$\text{Min } \|\mathbf{B}_1\mathbf{A_1}^\top - \mathbf{X}_{(1)}^\top\| \tag{6}$$

So, here A_1 is the unknown and the Matrix Least Square solution is,

$$\hat{\mathbf{A}}_1^\top = \left(\mathbf{B}_1^\top\mathbf{B}_1\right)^{-1}\mathbf{B}_1^\top\mathbf{X}_{(1)}^\top$$

i.e. $$\hat{\mathbf{A}}_1 = \mathbf{X}_{(1)}\mathbf{B}_1\left(\mathbf{B}_1^\top\mathbf{B}_1\right)^{-1}$$

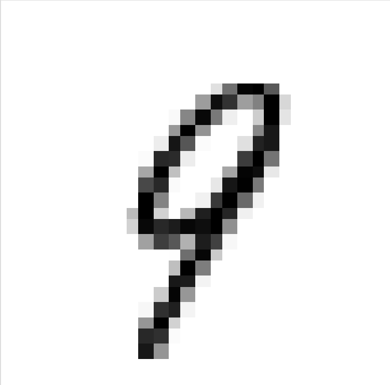
Main Updating Formula

$$\hat{\mathbf{A}}_1 = \mathbf{X}_{(1)}\mathbf{B}_1\left(\mathbf{B}_1^\top\mathbf{B}_1\right)^{-1}$$
$$\hat{\mathbf{A}}_2 = \mathbf{X}_{(2)}\mathbf{B}_2\left(\mathbf{B}_2^\top\mathbf{B}_2\right)^{-1}$$
$$\hat{\mathbf{A}}_3 = \mathbf{X}_{(3)}\mathbf{B}_3\left(\mathbf{B}_3^\top\mathbf{B}_3\right)^{-1}$$

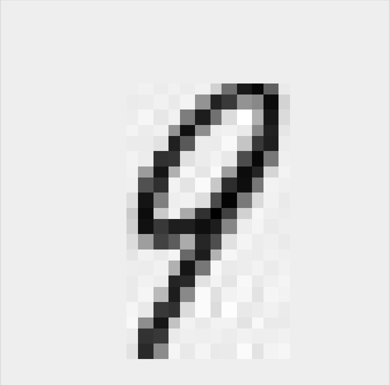
Image Compression

Example of 8 rank approximation

Actual



Compressed



MNIST Dataset

Actual Image

Resolution = 28 x 28
No. of Pixels = 784

Compressed Image

No. of Pixels = 8 * (28 + 28)
= 448

* Compression has been done by Python implementation of this theory

Image Compression

Example of 8 rank approximation

Actual Compressed

IIT Kgp Logo

Actual Image

Resolution = 896 * 800 * 3
No. of Pixels =2150400

Compressed Image

No. of Pixels = 8 * (896 + 800 + 3)
= 13592

So, **Computational Complexity** $= \boldsymbol{o(iter * nmk^3)}$

$\boldsymbol{iter}$ = no. of iteration
shape of tensor $= (i_1, i_2, \cdots, i_n)$
Rank approximation $= \boldsymbol{k}$

order of tensor $= \boldsymbol{n}$
$\boldsymbol{m} = \max(i_1, i_2, \cdots, i_n)$

Reference

[1] Kolda, Tamara G., and Brett W. Bader. Tensor decompositions and applications SIAM review 51.3 (2009): 455-500.

[2] Stephen Boyd, Lieven Vandenberghe. Introduction to Applied Linear Algebra: Vectors, Matrices, and Least Squares Cambridge University Press 2018, pp. 225 – 239

[3] J. B. Kruskal, Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics, Linear Algebra Appl., 18 (1977), pp. 95–138.

[4] J. B. Kruskal, Rank, decomposition, and uniqueness for 3-way and N-way arrays, in Multiway Data Analysis, R. Coppi and S. Bolasco, eds., North-Holland, Amsterdam, 1989, pp. 7–18.

Thank You!