

# 深入理解 | CP、Tucker分解

Author: [达布牛学习笔记]

Link: [<https://zhuanlan.zhihu.com/p/302453223>]

本文旨在对张量的CP、Tucker分解进行超详细讲解，希望能够帮助读者对这两张量分解算法有深刻的理解。

本文主要分成三个部分：section1主要介绍Notation and Preliminaries、section2介绍CP分解、section3介绍Tucker分解

## Introduction

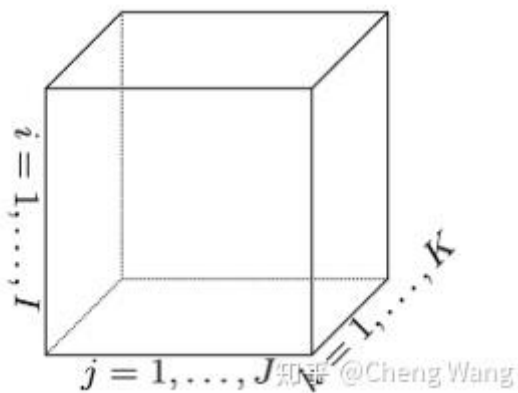
首先，给出本文最关键的一个点：

这两种张量分解算法可以被看做是张量奇异值分解的高阶扩展：CP分解将张量分解为秩一张量之和，Tucker分解是主成分分析（PCA）的高阶形式。

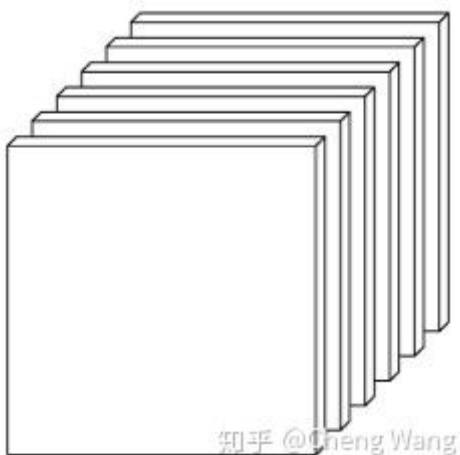
## Notation and Preliminaries

### 1. 何为张量？

向量是一维，矩阵是二维，张量就代表三维及以上，或者说，我们可以将张量理解为高阶矩阵

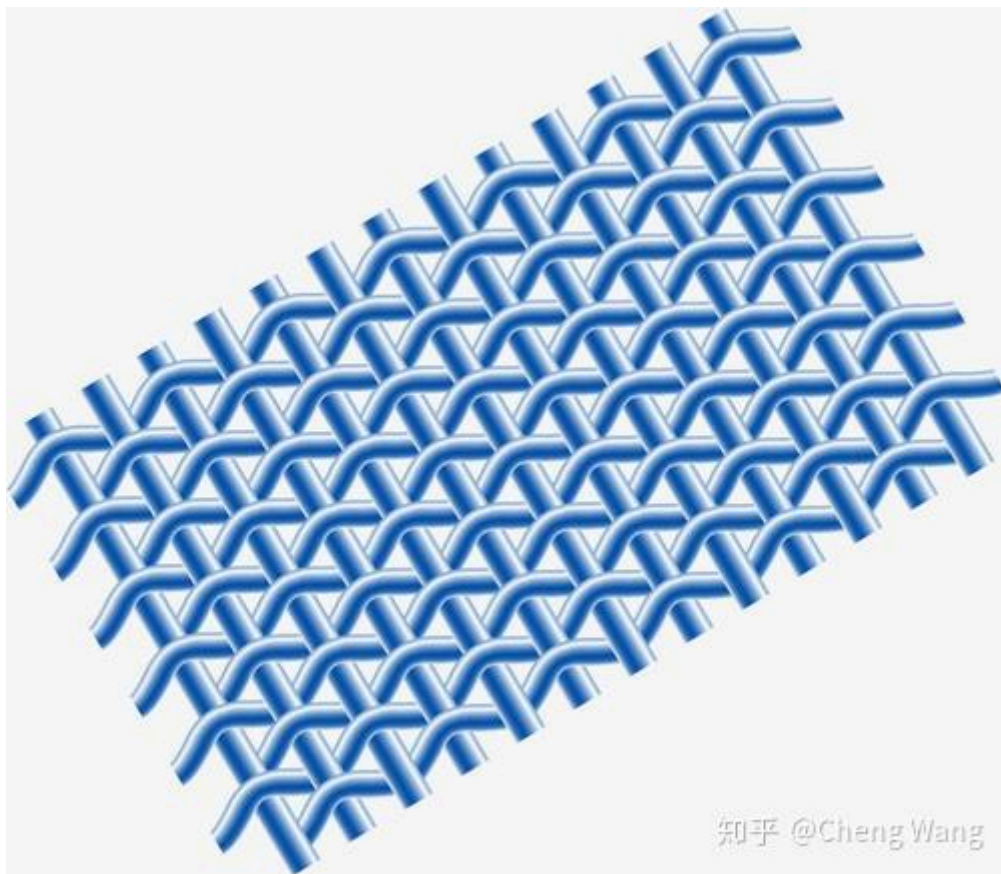


这里我举一个具体的应用场景：在网络测量领域，如果我们想要测量  $n$  个网络设备之间的延迟，那么，很自然地，在具体的某一时刻，我们会将这  $n$  个网络设备两两之间通信的延迟表示为一个  $n \times n$  矩阵。接下来，如果我们收集了许多不同时刻的数据，比方说  $n_3$  个时刻，我们将这些数据排列起来，这样就构成了一个  $n \times n \times n_3$  张量，如下图所示



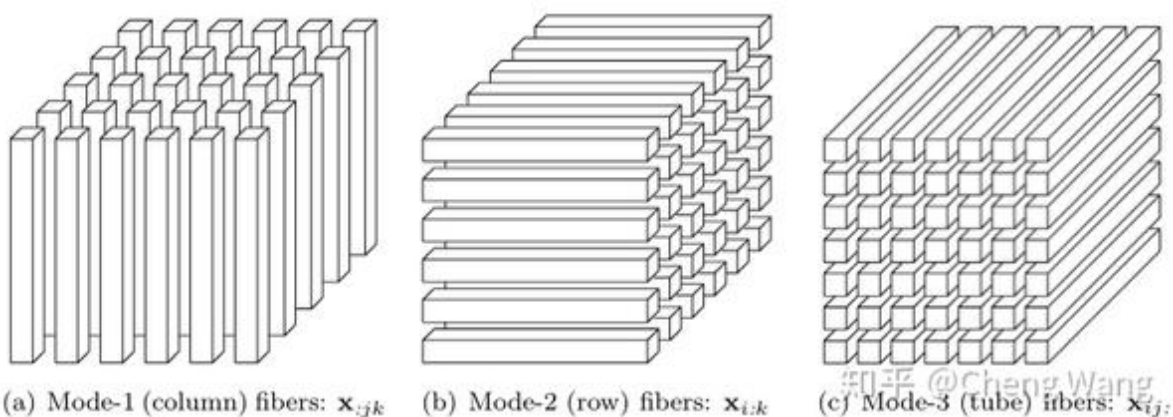
### 2. Fiber?

Fiber: 直译的话就是纤维（感觉这么翻译多少有点奇怪，因此还是直接用Fiber吧）



Fiber是矩阵行和列的高阶类似物，因此，我们不妨先来看看矩阵的Fiber。上图是一张纤维（真的是纤维！）的图片，我们可以看到，在矩阵中，Fiber其实也就是代表着行和列。

那么在张量中呢？话不多说，直接上图。



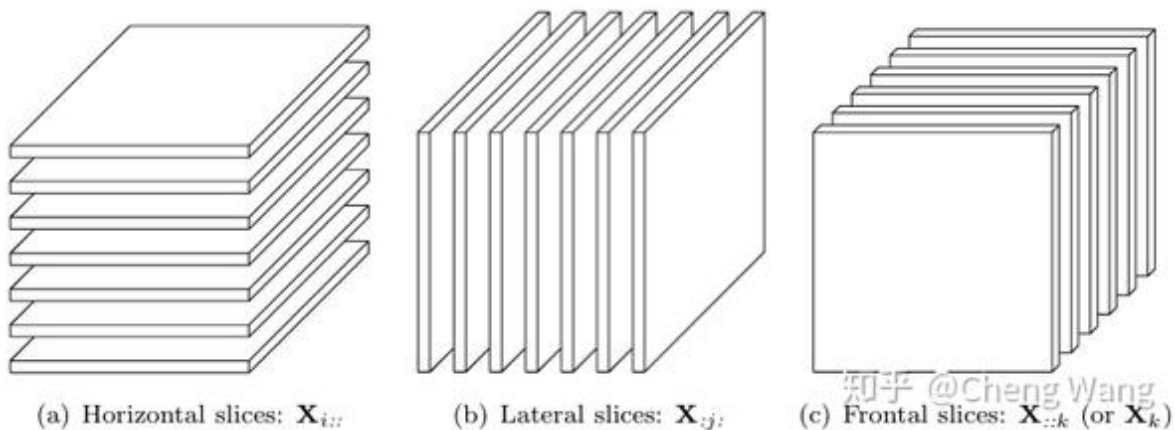
这张图还是很直观的，相信大家也对Fiber有了一个直观的理解。

严谨起见，还是给出三阶张量中不同方向的Fiber的符号表示： $X_{:,j,k}$ 、 $X_{i,:,k}$ 、 $X_{i,j,:}$  可以与上图——对应。

### 3. Slice?

Slice: 翻译成切片

和Fiber类似，张量的不同维度也是对应着不同的切片方式：**水平切片、侧面切片、正面切片**



正面切片也就对应着我们文章开头提到的网络延迟测量的例子。

三阶张量的切片的符号表示:  $X_{i,:,:}$ 、 $X_{:,j,:}$ 、 $X_{::,k}$

我们可以对比一下Fiber以及切片的符号表示, 可以看出来, **Fiber是固定三阶张量的两个维度, 切片是固定三阶张量的一个维度。如果扩展到更  $n$  阶的张量, 那么Fiber是固定张量的  $n - 1$  个维度, 切片是固定张量的  $n - 2$  个维度。**

#### 4. norm

假设现在有张量  $\chi \in R^{I_1 \times I_2 \times \dots \times I_N}$ , 张量的范数定义为张量的各个元素  $\chi_{ijk}$  的平方和开根号, 用公式表示如下:

$$\|\chi\| = \sqrt{\sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \dots i_N}^2}$$

可以看到, 这里张量范数的定义和矩阵的 *Frobenius* 范数定义是类似的, **更多关于矩阵范数的整理在我的这篇文章:**

[Cheng Wang: 矩阵基础 | 向量范数与矩阵范数](<https://zhuanlan.zhihu.com/p/265713608>)**5. 张量内积**

张量的内积? 在此之前我们不妨先回忆一下向量的内积与外积, 向量的内积就是两个向量对应元素乘积之和, 结果是个标量, 两个向量外积结果是一个矩阵, 三个向量外积是一个三阶张量。

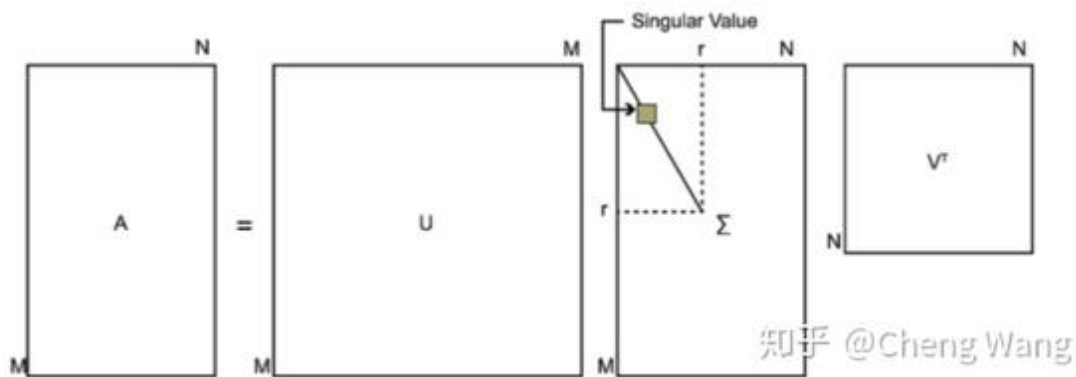
此处张量内积的定义和向量内积是类似的, 即张量对应元素乘积之和:

$$\langle \chi, \mathbf{y} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \dots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \dots i_N} y_{i_1 i_2 \dots i_N}$$

#### 6. 秩一张量?

顺着我们刚才提到过的三个向量外积是一个三阶张量的概念, 我们引出秩一张量的概念。

为了大家更好的理解, 我们可以先来回忆一下矩阵SVD的概念:



上图是矩阵SVD分解的示意图，一个矩阵可以分解为左右奇异向量矩阵以及一个奇异值矩阵的乘积形式。我们现在不妨以另外一种角度来看待矩阵的SVD分解：

我们如果拿出第一个左奇异向量以及第一个右奇异向量，这两个向量做外积，我们就可以得到一个矩阵，同时这两个奇异向量对应同一个奇异值，我们尝试将奇异值理解为这两个向量外积得到的这个矩阵的在原始矩阵中所占的权重，以此类推我们就可以得到所有奇异值对应的左右奇异向量外积的结果矩阵，然后将这些矩阵加起来就得到了原始矩阵。

现在回到我们的问题，秩一张量。在此，我先给出秩一张量的定义，相信大家对照上面我们对矩阵SVD分解的理解，理解秩一张量也就不是难事儿了。

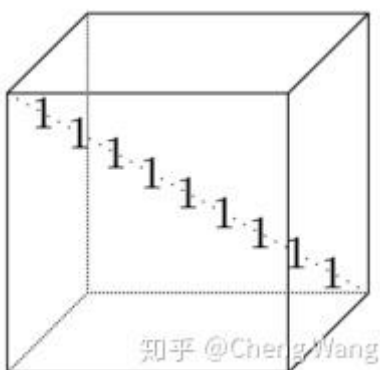
$$\mathcal{X} = \mathbf{a}^{(1)} \circ \mathbf{a}^{(2)} \circ \dots \circ \mathbf{a}^{(N)}$$

再次回到矩阵SVD分解的场景，上式的定义放在矩阵情况下，似乎应该就是这个样子：矩阵做SVD分解以后，奇异值矩阵中第一个奇异值为1，剩下的奇异值都为0。因此，奇异值为1对应的左右奇异向量的外积所得到的矩阵就是原始矩阵。

如果你对这一段内容有所理解的话，那么在section2要介绍的CP分解对你来说一定不是难点

## 7. 对角张量

类似于对角矩阵，对角张量很好理解，因此直接上图。



## 8. 张量矩阵化与向量化

在张量的分析与计算中，经常希望用一个矩阵代表一个三阶张量。此时，就需要有一种运算，能够将一个三阶张量（三路阵列）经过重新组织或者排列，变成一个矩阵（二路阵列）。将一个三路或N路阵列重新组织成一个矩阵形式的变换称为张量的矩阵化。张量的矩阵化有时也称张量的展开或张量的扁平化。

除了高阶张量的唯一矩阵表示外，一个高阶张量的唯一向量表示也是很多场合感兴趣的。高阶向量的向量化是一种将张量排列成唯一一个向量的变换。

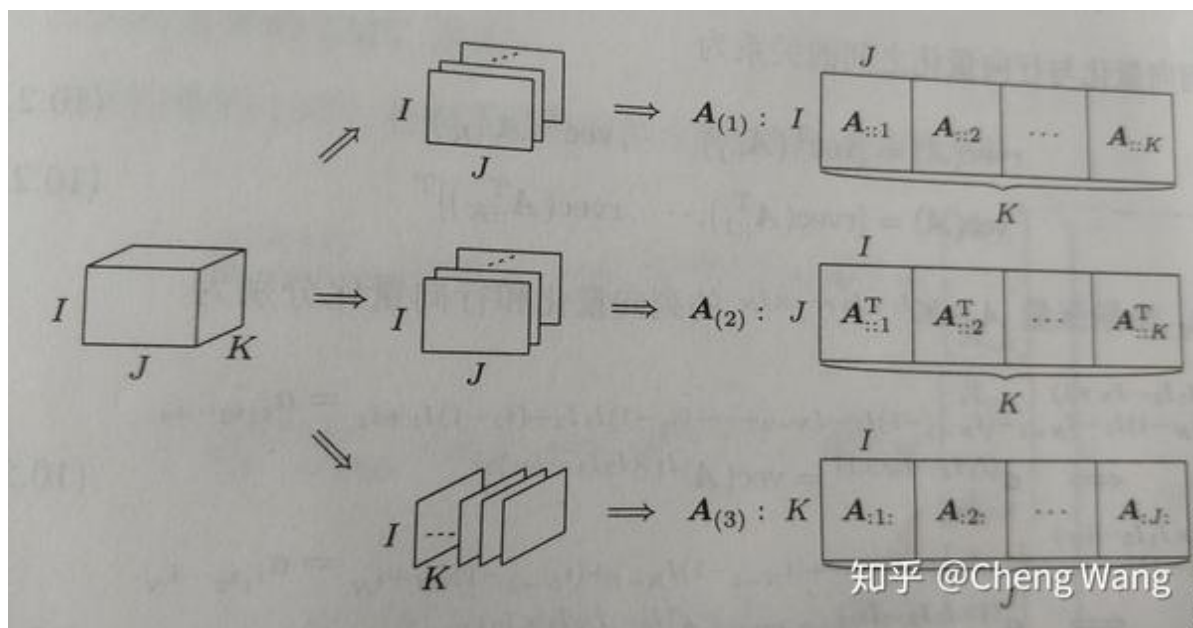
在张量的矩阵化过程中，首先可以分为水平展开与纵向展开，然后可以进一步细分为Kiers展开、LMV展开、Kolda展开。在此，我们仅介绍Kolda水平展开。

Kolda于2006年提出该方法，将  $N$  阶张量元素  $a_{i_1, i_2, \dots, i_N}$  映射为模式- $n$ 矩阵  $A_{(n)}$  的元素  $a_{i_n, j}^{I_n \times I_1 \times \dots \times I_{n-1} \times I_{n+1} \times \dots \times I_N}$ ，其中

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

知乎 @Cheng Wang

光看上式可能很难理解，因此可以借助下图来理解：



我们再来举个例子：

$$\mathbf{X}_1 = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix}, \quad \mathbf{X}_2 = \begin{bmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{bmatrix}.$$

知乎 @Cheng Wang

$\mathbf{X}_1$ 、 $\mathbf{X}_2$  代表张量  $\chi$  的两个正面切片矩阵。该张量的模式- $n$ 展开如下所示：

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

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

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

知乎 @Cheng Wang

对于该张量的向量化形式，因为很好理解，因此直接给出该例的向量化结果：



$$\text{vec}(\mathbf{X}) = \begin{bmatrix} 1 \\ 2 \\ \vdots \\ 24 \end{bmatrix}.$$

知乎 @Cheng Wang

## 9. mode-n product

在此我们仅考虑三阶张量与矩阵的乘积，由于该方法是Tucker定义的，现在常称为Tucker积。

**三阶张量的Tucker积：**考虑三阶张量  $\chi \in K^{I_1 \times I_2 \times I_3}$  和矩阵  $A \in K^{J_1 \times I_1}$ ， $B \in K^{J_2 \times I_2}$ ， $C \in K^{J_3 \times I_3}$  的乘积。三阶张量的Tucker模式-1积  $\chi \times_1 A$ ，模式-2积  $\chi \times_2 B$ ，模式-3积  $\chi \times_3 C$  分别定义为：

$$\begin{aligned} (\chi \times_1 A)_{j_1 i_2 i_3} &= \sum_{i_1=1}^{I_1} x_{i_1 i_2 i_3} a_{j_1 i_1}, \quad \forall j_1, i_2, i_3 \\ (\chi \times_2 B)_{i_1 j_2 i_3} &= \sum_{i_2=1}^{I_2} x_{i_1 i_2 i_3} b_{j_2 i_2}, \quad \forall i_1, j_2, i_3 \\ (\chi \times_3 C)_{i_1 i_2 j_3} &= \sum_{i_3=1}^{I_3} x_{i_1 i_2 i_3} c_{j_3 i_3}, \quad \forall i_1, i_2, j_3 \end{aligned}$$

知乎 @Cheng Wang

我们先来考虑一下模式-1积， $\chi \times_1 A$ 。我们先将  $\chi$  进行模式-1展开，所得矩阵的维度为  $I_1 \times I_2 I_3$ 。如果你有点忘了模式-n展开，可以再翻回去看一下上一节。

也就是说，我们将张量展开成了一个矩阵，那么模式-n乘积也就变成了两个矩阵相乘，这个大家肯定都不陌生。这两个矩阵的维度分别是  $I_1 \times I_2 I_3$  以及  $A \in J_1 \times I_1$ ，因此，很明显的一点是，如果按照原来的顺序，这两个矩阵是无法直接相乘的，因此我们需要把  $A$  放到张量展开矩阵的左边。所得结果是一个  $J_1 \times I_2 I_3$  的矩阵，注意，此处我们还需要将这个矩阵还原成张量，所以最终的结果是一个  $J_1 \times I_2 \times I_3$  的张量。

模式-2积与模式-3积就不再赘述了。

## 10. 特殊矩阵乘积

关于特殊矩阵乘积，如矩阵内积、Hadamard积、Kronecker积与Khatrri-Rao积等，我在另外一篇文章中介绍过，感兴趣的同学可以去看一下。

[Cheng Wang：矩阵基础 | 矩阵特殊乘积运算](<https://zhuanlan.zhihu.com/p/280898025>)至此，基本知识就介绍完毕了，由于本文的主要目的是讲解CP分解与Tucker分解，因此对于以上知识的讲解可能并不详细，如果有不清楚的同学可以去看一下其他博客。

# CP分解

CP分解是将张量分解为一系列的秩一张量之和。还记得文章开头我写的那句话吗？我们可以将CP分解看作是矩阵SVD分解的高阶扩展，在这一部分，我们就要一起来看看为什么是这样？

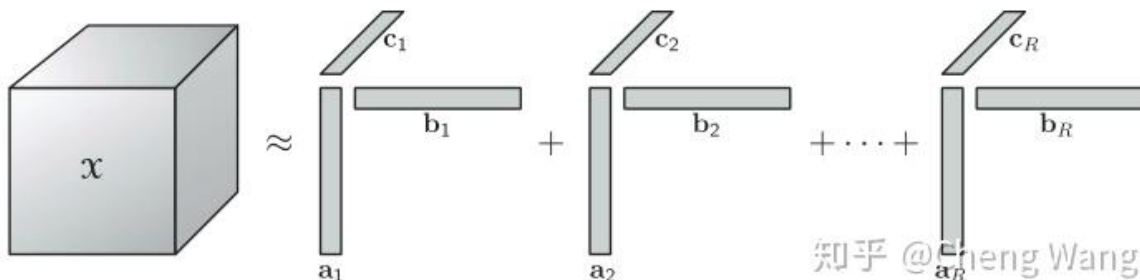
在开始之前，我要再次强调一遍 Introduction部分第六节秩一张量的内容，如果你看到这里已经有所遗忘的话，希望你能再回头好好看看，尤其是黑体加粗的那段话，关于矩阵SVD分解的理解方式。

在Introduction6中，我们已经讲到过，矩阵的SVD分解可以看作 **左右奇异向量外积所得矩阵+对应的权重（也就是对应奇异值）** 所得的一系列矩阵之和。我们不妨将这个权重放到左右奇异向量外积所得的矩阵中，然后将这种带有权重的矩阵称作因子矩阵，那么**矩阵SVD分解就是将矩阵分解为许多因子矩阵之和**。

将这种思想扩展到张量的CP分解，也就是说**张量的CP分解是将张量分解为许多因子张量之和**。从而引出张量CP分解的定义：对于一个  $\chi \in R^{I \times J \times K}$ ，其CP分解公式如下：

$$\chi \approx \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r,$$

注意  $\mathbf{a}_r, \mathbf{b}_r, \mathbf{c}_r$  的维度分别是  $\mathbf{a}_r \in R^I, \mathbf{b}_r \in R^J, \mathbf{c}_r \in R^K$ ， $R$  就是我们所说的CP秩。更直观的理解看下图：



在我学习CP分解的时候，我认为最有效的一招是去理解张量中的每一个元素是由哪些元素组成的。因此，接下来，我们来对上图进行更深入的探究，同时也是为了引出CP分解的另外一种定义形式。

$\chi_{111}$  元素就是张量左上角的元素，我们知道，这个元素一定是各个因子张量左上角的元素之和组成的。我们不妨来考虑一下上图的第一个因子张量。

在此之前，我们先定义一些新的符号，在上图中  $\mathbf{a}_1, \mathbf{b}_1, \mathbf{c}_1$  代表三个向量，现在定义  $a_{1i}, b_{1j}, c_{1k}$  代表这三个向量中不同位置的标量值。为了书写方便，我们将第一个下标1去掉，现在就变成了  $a_i, b_j, c_k$ 。假设  $I = J = K = 3$ ，那么  $\mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$  如下图所示：

$$a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \quad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad c = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

$$a \circ b = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \times [b_1 \ b_2 \ b_3] = \begin{bmatrix} a_1 b_1 & a_1 b_2 & a_1 b_3 \\ a_2 b_1 & a_2 b_2 & a_2 b_3 \\ a_3 b_1 & a_3 b_2 & a_3 b_3 \end{bmatrix}$$

$a \circ b \circ c =$  一个张量.

知乎 @Cheng Wang

该张量的第一个正面切片:

$$\begin{bmatrix} a_1 b_1 c_1 & a_1 b_2 c_1 & a_1 b_3 c_1 \\ a_2 b_1 c_1 & a_2 b_2 c_1 & a_2 b_3 c_1 \\ a_3 b_1 c_1 & a_3 b_2 c_1 & a_3 b_3 c_1 \end{bmatrix}$$

第二个正面切片:

$$\begin{bmatrix} a_1 b_1 c_2 & a_1 b_2 c_2 & a_1 b_3 c_2 \\ a_2 b_1 c_2 & a_2 b_2 c_2 & a_2 b_3 c_2 \\ a_3 b_1 c_2 & a_3 b_2 c_2 & a_3 b_3 c_2 \end{bmatrix}$$

第三个正面切片:

$$\begin{bmatrix} a_1 b_1 c_3 & a_1 b_2 c_3 & a_1 b_3 c_3 \\ a_2 b_1 c_3 & a_2 b_2 c_3 & a_2 b_3 c_3 \\ a_3 b_1 c_3 & a_3 b_2 c_3 & a_3 b_3 c_3 \end{bmatrix}$$

知乎 @Cheng Wang

也就是说, 第一个因子张量的左上角位置元素为  $a_{11}b_{11}c_{11}$ , 第二个因子张量左上角位置元素为  $a_{21}b_{21}c_{21}$ , 以此类推, 我们就可以知道  $\chi_{111} = \sum_{r=1}^R a_{r1}b_{r1}c_{r1}$ , 张量  $\chi$  其他位置的元素也可以用同样的方法推导。

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

我们将CP分解中最小的单元看作了不同的向量, 向量的外积组成了不同的张量, 因子张量之和就是原始张量。接下来, 我们介绍CP分解的另一种表示形式, 在这种表示方式中, 最小的单元是矩阵。



我们令  $A = [a_1, a_2, \dots, a_R], B = [b_1, b_2, \dots, b_R], C = [c_1, c_2, \dots, c_R]$ ,  $X_{(1)}, X_{(2)}, X_{(3)}$  代表  $\chi$  张量的模式-1, 模式-2, 模式-3分解,  $\odot$  代表Khatri-Rao积, 在我的这篇博客中有介绍:

[Cheng Wang: 矩阵基础 | 矩阵特殊乘积运算](<https://zhuanlan.zhihu.com/p/280898025>)那么, 我们就可以定义:

$$X_{(1)} \approx A(C \odot B)^T,$$

$$X_{(2)} \approx B(C \odot A)^T,$$

$$X_{(3)} \approx C(B \odot A)^T.$$

此处, 我仅对  $X_{(1)}$  进行简单推导, 主要关注于等式右边计算过程中的矩阵维数变化:

Handwritten derivation showing the dimensions of matrices  $A, B, C$  and their Khatri-Rao product  $C \odot B$ , and the resulting matrix  $A(C \odot B)^T$ .

$$A = \begin{bmatrix} a_1 & a_2 & \dots & a_R \end{bmatrix} \in \mathbb{R}^{I \times R}$$

$$B = \begin{bmatrix} b_1 & b_2 & \dots & b_R \end{bmatrix} \in \mathbb{R}^{J \times R}$$

$$C = \begin{bmatrix} c_1 & c_2 & \dots & c_R \end{bmatrix} \in \mathbb{R}^{K \times R}$$

$$C \odot B = \begin{bmatrix} b_1 \odot c_1 & b_2 \odot c_2 & \dots & b_R \odot c_R \end{bmatrix} \in \mathbb{R}^{JK \times R}$$

$$(C \odot B)^T \in \mathbb{R}^{R \times JK}$$

$$A(C \odot B)^T = \begin{bmatrix} a_1 & a_2 & \dots & a_R \end{bmatrix} \begin{bmatrix} b_1 \odot c_1 \\ b_2 \odot c_2 \\ \vdots \\ b_R \odot c_R \end{bmatrix} \in \mathbb{R}^{I \times JK}$$

知乎 @Cheng Wang

更进一步, 我们尝试用因子矩阵表达张量  $\chi$  中不同的正面切片, 令  $X_k$  代表第  $k$  个正面切片矩阵, 那么:

$$X_k \approx A D^{(k)} B^T, \quad \text{where } D^{(k)} \equiv \text{diag}(\mathbf{c}_{k:}) \quad \text{for } k = 1, \dots, K.$$

综上所述, CP模型就有了另一种表示方式:

$$\chi \approx [A, B, C] \equiv \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r.$$

知乎 @Cheng Wang

此外, 如果我们将  $A, B, C$  的列向量进行归一化, 那么意味着我们将不同向量代表的权重都分离出来, 因此, CP模型也可以有如下表示:

$$\mathcal{X} \approx [\lambda; \mathbf{A}, \mathbf{B}, \mathbf{C}] \equiv \sum_{r=1}^R \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r.$$

知乎 @Cheng Wang

至此，我们的CP分解讲解就告一段落了，接下来开始介绍Tucker分解。

## Tucker分解

Tucker分解与CP分解可谓是学习张量的同学们的两个拦路虎，通常我们认为，CP分解是张量分解的一种特殊情况，接下来就让我们开始攻克Tucker分解吧！

如果你完整地看到了现在，并且对CP分解部分地内容也理解了，那么恭喜你，Tucker分解对你来说也一定不是问题！

与CP分解的讲解过程类似，我们还是再次回顾一下文章开头的那句话：**Tucker分解是主成分分析 (PCA) 的高阶形式。**

我在令一篇文章中也有对PCA方法进行过简要的介绍，如果你对PCA有些遗忘的话，可以先去看看这篇文章：

[Cheng Wang: 深入理解 | PCA降维](<https://zhuanlan.zhihu.com/p/269432722>)其实，通俗来讲，PCA的目的就是**用  $k$  个主分量概括表达统计相关的  $n$  个特征**，因此PCA算法是一种降维方法。那么，我们之前说Tucker分解是PCA的高阶形式，那么我们能否尝试用这种降维的思想去理解Tucker分解呢？

总结一下，我们现在提出了两个问题：

1. Tucker分解为何是PCA降维的高阶形式？
2. CP分解为何是Tucker分解的特殊情况？

接下来，我们会在逐步的讲解当中解决这两个问题。

首先，我们需要确立一种思维，我们知道一个矩阵可以理解作为一种线性变换，那么对于  $A \times B$ ，我们可以理解为在矩阵  $A$  上施加一种线性变换，如果同时考虑矩阵维数变化的话， $A \in R^{I_1 \times I_2}$ ,  $B \in R^{I_2 \times I_3}$ ，那么  $(A \times B) \in R^{I_1 \times I_3}$ ，我们尝试这样去理解： $A \times B$  相当于沿着矩阵的第二个维度施加了线性变化，使得第二个维度由  $I_2$  变成了  $I_3$ 。

还是怀揣着这种想法，我们再来看看张量的模式-n积。

与Introduction9中的例子相同，我们现在来考虑  $\chi \times_1 A$ ，其中张量  $\chi \in R^{I_1 \times I_2 \times I_3}$ ，矩阵  $A \in R^{J_1 \times I_1}$ ，我们知道这个模式-1积的结果是一个  $J_1 \times I_2 \times I_3$  的张量，那么我们不妨认为模式-1积是沿着张量  $\chi$  的第一个维度施加了线性变换，使得第一个维度从  $I_1$  变化到了  $J_1$ 。如果  $J_1 < I_1$  的话，就相当于沿着张量的第一维度进行了一种降维操作。

(其实在这里，同学们就可以尝试和PCA降维做联系了。)

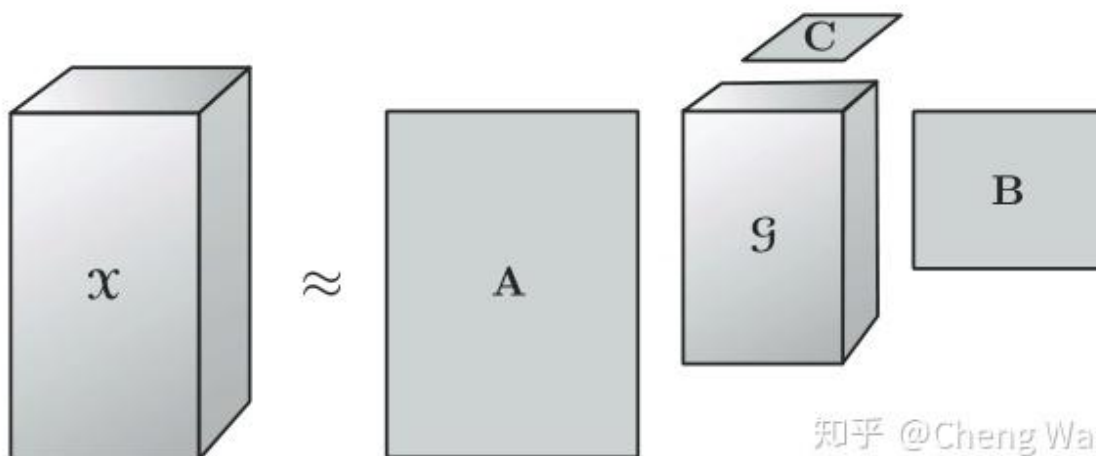
现在，我们给出Tucker分解的定义，对于张量  $\chi \in R^{I \times J \times K}$ ，其Tucker分解形式如下：

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

其中， $A \in R^{I \times P}$ ,  $B \in R^{J \times Q}$ ,  $C \in R^{K \times R}$  是Tucker分解得到的因子矩阵，

$$\mathcal{G} \in \mathbb{R}^{P \times Q \times R}$$

代表核心张量，它的元素代表不同因子矩阵之间相互作用的水平。Tucker分解示意图如下：



类似于CP分解的第二种表示形式，Tucker分解也可以表示如下：

$$[\mathcal{G}; \mathbf{A}, \mathbf{B}, \mathbf{C}]$$

我们接下来再来看看张量中每一个元素的表示形式：

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

(具体推导过程就就不再赘述了，其实和CP分解中元素的推导过程是大同小异的)

介绍了这么多公式，我们现在一起来梳理一下，（对照下面Tucker分解的公式）**张量的Tucker分解本质就是将原始张量分解为一个核心张量以及对应不同维度的因子矩阵（三阶张量分解中就是三个因子矩阵），每个因子矩阵我们可以看作对应于不同维度的线性变换操作**

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

如果  $P, Q, R$  小于  $I, J, K$  的话，**核心张量就相当于原始张量的压缩形式，即对原始张量进行降维操作的结果。因此，Tucker分解的确是可以看作PCA降维的高阶形式。**

这样我们就回答了第一个问题，那么对于第二个问题：为什么CP分解可以看作Tucker分解的特殊情况呢？

对于这个问题，其实思维敏捷的同学应该已经稍微有一点思路了，不过对于这个问题，上面的内容可能还是比较模糊。因此，接下来我们引入Tucker分解的另外一种形式，该表示形式同样在CP分解中介绍过，如下：

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

CP分解

对应地，Tucker的表示如下：

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

Tucker分解

此处  $G_{(1)}$  代表核心张量的模式-1展开， $\otimes$  是Kronecker积， $\odot$  Khatri-Rao积其实是Kronecker积的扩展形式，如果你不理解这两种乘积，还是去推荐看一下我的这篇文章：

[Cheng Wang：矩阵基础 | 矩阵特殊乘积运算](<https://zhuanlan.zhihu.com/p/280898025>)因此，**如果我们的核心张量是一个超对角张量的话（即，在Introduction7介绍的对角张量的基础上约束  $P = Q = R$ ），那么Tucker分解就变成了CP分解。**

（长出一口气...）至此，Tucker分解也基本介绍完毕了，其实在Tucker分解与CP分解中，它们的秩以及求解过程也是十分值得讨论的问题，但是在这篇文章中并不打算介绍，在未来我会找时间好好写写这两个方面的内容。

如果这篇文章对你有帮助的话，点个赞吧~

## 欢迎关注我的公众号【达布牛学习笔记】

#### 参考

1. Kolda, Tamara G., and Brett W. Bader. "Tensor Decompositions and Applications." 51 (2009): 455-500.
2. 《矩阵分析与应用第二版》张贤达著