中图分类号： O24　　　　　　学校代码： 10055
UDC：　　　　　　　　　　　　密级： 公开

南開大學

硕 士 学 位 论 文

基于 Tensor Train 分解的两种低秩张量补全算法

Two methods solving low tensor train rank tensor completion

论文作者 唐云飞　　　　指导教师 杨庆之 教授
申请学位 理学硕士　　　　培养单位 数学科学学院
学科专业 计算数学　　　　研究方向 最优化方法
答辩委员会主席 吴春林　　评 阅 人 张新珍 吴春林

南开大学研究生院

二〇二一年五月

南开大学学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下进行研究工作所取得的研究成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：唐云飞　　　2021 年 6 月 18 日

非公开学位论文标注说明

（本页表中填写内容须打印）

根据南开大学有关规定，非公开学位论文须经指导教师同意、作者本人申请和相关部门批准方能标注。**未经批准的均为公开学位论文，公开学位论文本说明为空白。**

<table>
<tr><td>论文题目</td><td colspan="3"></td></tr>
<tr><td>申请密级</td><td colspan="3">□限制(≤2 年)　　□秘密(≤10 年)　　□机密(≤20 年)</td></tr>
<tr><td>保密期限</td><td colspan="3">20　　年　　月　　日至 20　　年　　月　　日</td></tr>
<tr><td>审批表编号</td><td></td><td>批准日期</td><td>20　　年　　月　　日</td></tr>
</table>

南开大学学位评定委员会办公室盖章(有效)

注：限制★2 年(可少于 2 年)；秘密★10 年(可少于 10 年)；机密★20 年(可少于 20 年)

南开大学学位论文使用授权书

本人完全了解《南开大学关于研究生学位论文收藏和利用管理办法》关于南开大学(简称“学校”)研究生学位论文收藏和利用的管理规定，同意向南开大学提交本人的学位论文电子版及相应的纸质本。

本人了解南开大学拥有在《中华人民共和国著作权法》规定范围内的学位论文使用权，同意在以下几方面向学校授权。即：

1.学校将学位论文编入《南开大学博硕士学位论文全文数据库》，并作为资料在学校图书馆等场所提供阅览，在校园网上提供论文目录检索、文摘及前 16 页的浏览等信息服务；

2.学校可以采用影印、缩印或其他复制手段保存学位论文；学校根据规定向教育部指定的收藏和存档单位提交学位论文；

3.非公开学位论文在解密后的使用权同公开论文。

本人承诺：本人的学位论文是在南开大学学习期间创作完成的作品，并已通过论文答辩；提交的学位论文电子版与纸质本论文的内容一致，如因不同造成不良后果由本人自负。

本人签署本授权书一份（此授权书为论文中一页），交图书馆留存。

学位论文作者暨授权人(亲笔)签字：唐云飞

2021 年 6 月 18 日

南开大学研究生学位论文作者信息

<table>
<tr><td>论 文 题 目</td><td colspan="5">基于 Tensor Train 分解的两种低秩张量补全算法</td></tr>
<tr><td>姓 名</td><td>唐云飞</td><td>学号</td><td>2120180035</td><td>答辩日期</td><td>2021 年 5 月 27 日</td></tr>
<tr><td>论 文 类 别</td><td colspan="5">博士 ☐ 学历硕士 ☑ 专业学位硕士 ☐ 同等学力硕士☐ 划 √ 选 择</td></tr>
<tr><td>学院(单位)</td><td colspan="2">数学科学学院</td><td colspan="2">学科/专业(专业学位)名称</td><td>计算数学</td></tr>
<tr><td>联 系 电 话</td><td colspan="2">18571631118</td><td>电 子 邮 箱</td><td colspan="2">719831040@qq.com</td></tr>
<tr><td colspan="6">通信地址(邮编)： 天津市南开区卫津路 94 号南开大学西区公寓(300071)</td></tr>
<tr><td>非公开论文编号</td><td colspan="2"></td><td>备注</td><td colspan="2"></td></tr>
</table>

注：本授权书适用我校授予的所有博士、硕士的学位论文。如已批准为非公开学位论文，须向图书馆提供批准通过的《南开大学研究生申请非公开学位论文审批表》复印件和“非公开学位论文标注说明”页原件。

摘要

张量是矩阵和向量的高阶推广，现实世界中的很多数据都有着自然的张量形式。然而在实际应用中，受多种因素的影响，原始的张量数据很可能是不完备的，或者有某些元素是缺失的。所以我们需要根据张量已知的元素和一些对张量结构的先验假设来补全整个张量数据。这其中，一个常用的假设是原始的张量是低秩的，或者是近似低秩的。在这种情况下我们的目的就变成了在给定的线性约束下去优化张量的某种秩。然而，高阶张量的秩的定义并不是唯一的，并且每一种秩都对应着某种特定的张量分解方法。在这些张量分解方法中，CP (CANDECOMP/PARAFAC)分解的复杂度和张量的阶数呈理想的线性关系。但CP 分解有着两个理论上的主要缺点：一是 CP 分解的计算是 NP 困难的，目前依然没有有效的计算方法；二是 CP 秩不是下半连续的函数，在实际计算中是不稳定的。相比于 CP 分解，张量的 Tucker 分解是一种数值稳定的分解方式，但 Tucker 分解的复杂度会随着张量阶数的增长呈指数增长。为了克服两种分解各自的缺点，本文考虑利用最近提出的张量 Tensor Train (TT)分解。TT 分解的计算同样是数值稳定的，并且其复杂度和张量的阶数呈比较理想的线性关系，从而在一定程度上解决了前面两种分解存在的问题。

在假设原始张量是 TT 分解意义下的低秩张量后，我们提出了基于 TT 分解的低秩张量补全模型。我们借鉴了低秩矩阵补全理论中的方法，即用矩阵核范数来逼近矩阵的秩。矩阵核范数是矩阵的秩在单位球上最紧的凸包，所以用矩阵核范数来逼近矩阵的秩是矩阵补全问题中一个非常有效的方法。张量的 TT 核范数是矩阵核范数的一种推广，在本文中我们利用张量的 TT 核范数来逼近张量的 TT 秩，从而将原始的秩优化模型松弛为了一个凸优化模型。我们分别基于 Douglas-Rachford 分裂方法和 ADMM 设计了两种高效的一阶算法。数值算例也证明了我们所提出的算法在实际问题中的有效性。最后，在文章的结尾处我们讨论了一些可以用于加速我们算法的技巧和策略。

关键词：张量补全；低秩张量；Tensor Train (TT)分解；凸优化

Abstract

Tensors are higher order generalizations of matrices and vectors. Many real-world data have a natural tensor form. However, in many applications the tensor of interest may be incomplete or may have corrupted elements. And one often needs to estimate the missing elements based on the partially observed data and some presumed structures of the original tensor. One commonly used assumption is that the tensor is of low rank, or approximately so. In such cases, to complete the partially observed tensor we aim at optimizing the tensor rank under given linear constraints. However, the definitions of tensor rank are not unique for order higher than 3. And each definition corresponds to a certain tensor decomposition. The CP (CANDECOMP/PARAFAC) decomposition, although has the ideally linear complexity with the tensor order, suffers from the theoretical deficits that its computation is NP-hard and it is not a lower semi-continuous function. The Tucker decomposition provides an alternate numerically stable way to decompose a tensor, but its complexity grows exponentially with the tensor order. To tackle these issues, we resort to the recently proposed tensor train decomposition which shares the merits of the CP and the Tucker decompositions, as it is numerically stable and its complexity scales linearly with the tensor order.

By assuming a low rank prior of the original tensor under the Tensor Train decomposition, we propose a low rank tensor completion model and derive two algorithms to solve this model. We borrow the idea from the low rank matrix completion theory in which the matrix rank is approximated by matrix nuclear norm, which is the tightest convex envelope for the matrix rank on the unit ball. This relaxation is also proved to be a power approach for matrix completion. As an extension of matrix nuclear norm, the tensor train nuclear norm is used to approximate the tensor train rank and the original model in thus converted into a convex program. We then design two algorithms that are based on the Douglas-Rachford splitting technique and the classical ADMM to efficiently solve this convex program. Numerical experiments validated the effectiveness of the proposed algorithms. And in the final part we end our discussion with some possible speed-up strategies to further improve algorithm performance.

Key Words: tensor completion; low rank tensors; tensor train decomposition; convex optimization.

目录

摘要 I

Abstract.. II

目录 IV

第一章 引言 1

第二章 张量的预备知识和 Tensor Train 分解 3

第一节 张量预备知识 3

第二节 张量的 Tensor Train 分解 5

第三章 基于 TT 秩的低秩张量恢复模型 7

第一节 相关研究介绍 7

第二节 低秩张量恢复的 TT 秩模型 10

第四章 两种基于 TT 分解的低秩张量恢复算法 12

第一节 基于 Douglas-Rachford 分裂方法的 DR-TT 算法 12

4.1.1 模型的等价转化 12

4.1.2 函数 $\boldsymbol{f}$ 的邻近算子 13

4.1.3 函数 $\boldsymbol{g}$ 的邻近算子 14

4.1.4 DR-TT 算法的收敛性分析 17

第二节 基于 ADMM 的 ADMM-TT 算法 17

4.2.1 模型的等价转化 18

4.2.2 变量$\boldsymbol{Y}$的迭代式 19

4.2.3 变量$\boldsymbol{X}$的迭代式 19

4.2.4 ADMM-TT 算法的收敛性分析 20
第五章 数值实验 22
第一节 算法细节和参数选取 22
第二节 随机张量恢复实验 25
5.2.1 低 TT 秩随机张量的恢复 25
5.2.2 低 Tucker 秩随机张量的恢复 27
第三节 图像数据恢复实验 28
第六章 结论 31
参考文献 32
致谢 34
个人简历 35

第一章 引言

张量是矩阵和向量的高阶推广，在数值计算领域，一个 d 阶张量通常被视为有着 d 个索引的 d 维数组。张量分解的目的是用尽可能简洁的表达式来表示给定的张量。有关张量和张量分解的深入介绍可以参考文献[1]。现实世界中的很多数据都有着自然的张量形式，例如 RGB 彩色图片是 3 阶张量，而由彩色图片组成的视频是 4 阶张量等。在实际应用中，受系统损坏，环境干扰和人为操作失误等因素的影响，原始的张量数据可能是不完整的，甚至是随机的。例如在视频网站的评价系统中，可以用一个矩阵 M 来表示用户对视频的评分，即矩阵元素 $M(i,j)$ 代表了用户 i 对视频 j 的评分值。然而单个用户几乎不可能看完所有的视频，所以M的很多元素都是缺失的。又比如在卫星遥感领域，由于云层的遮盖或者系统本身的故障等，卫星拍摄到的高光谱图像往往也会有一部分是缺失的。在这些情况下我们就需要将张量数据缺失的部分恢复出来。这一问题被称为张量补全或者张量恢复，在这一问题中，我们总假定原始的张量是存在且确定的。然而如果不对原始张量的结构加以约束，那么张量补全问题实际上是无解的，即我们可以令张量的未知元素取任意的值。所以在张量补全问题中，我们往往会根据已有经验来假定原始的张量具有某些先验的结构。在很多领域，尽管张量空间的维度很高，但我们感兴趣的张量却通常是低秩的，或者近似为低秩的[1]，所以张量的低秩结构是一个合理且应用广泛的假设。

在假定原始张量具有某种意义下的低秩结构之后，通常我们的目标是去寻找一个满足约束的秩最低的张量。我们称这个问题为张量的低秩补全(LRTC)问题，或者低秩恢复问题。张量的低秩补全问题在很多领域都有着应用，例如高光谱数据恢复[2][3]、图像及视频修复[4][5]、文本分析[6]和多任务学习[7]等。如果所需要恢复的张量是 2 阶的，那么问题简化成了大家熟知的低秩矩阵补全(LRMC)问题。低秩矩阵补全问题的理论研究和实际应用都已经比较成熟，目前也已经发展了出许多有效的算法，如 SVT [8]、LMaFit [9]和 APGL [10]等。

然而高阶张量有多种定义秩的方式，并且不同的秩对应着不同的张量分解。在这些分解中，应用最为广泛的两种是 CP 分解和 Tucker 分解。鉴于张量的 CP

秩的计算是 NP 困难的[11]，所以有关 CP 秩的优化问题在实际中比较难以操作。与 CP 秩相比，张量的 Tucker 分解和 Tucker 秩的计算是数值稳定的且在多项式时间内可行，这使得低 Tucker 秩结构成为了张量补全问题中一个合适的先验假设。文献[12]从大数据的角度总结了目前流行的基于 CP 秩和 Tucker 秩的张量补全算法。尽管基于 Tucker 秩的张量补全研究已经有了很多进展，在应用中，基于 Tucker 秩的张量补全算法在某些问题上的表现却不太令人满意。其中重要的两点原因是：

(1) 基于 Tucker 秩的低秩张量恢复模型依赖于张量的模 i 展开，这种展开方式将张量的元素按一定规则重新排列后得到一个矩阵，但这个矩阵是高度不均衡的，即矩阵的列数远远大于行数。这使得基于 Tucker 秩的低秩恢复模型不能很好的反映张量元素间的整体联系[20]；

(2) Tucker 分解的存储量和计算量与张量的阶数呈指数关系，所以对大规模问题而言，其计算量非常大。

为了解决上面提出来的两点问题，在本文中我们将考虑一种新的张量分解方式，即张量的 Tensor Train (TT)分解[14]。TT 分解将一个 d 阶张量分解为 d 个至多 3 阶的张量的 TT 乘积。对于低 TT 秩的张量，这种分解的存储量和计算量与张量阶数呈线性关系，从而在一定程度上解决了问题(2)。同时，基于 TT 秩的低秩恢复模型依赖的是张量的 i 阶展开，与模 i 展开相比，由这种展开方式得到的矩阵比较均衡，所以能更好的反映张量元素之间的关系。

本文接下来的内容中按照如下方式来组织：在第二章我们简要的介绍张量的相关知识和记号；在第三章我们回顾低秩张量恢复的相关研究，同时提出基于 TT 秩的低秩张量恢复模型；在第四章我们分别利用 Douglas-Rachford 分裂方法和 ADMM 来求解所提出的模型；在第五章我们用数值实验来展示提出的算法的有效性；最后在第六章我们总结本文的内容并提出一些算法加速的策略。

第二章 张量的预备知识和 Tensor Train 分解

第一节 张量预备知识

我们从数值计算的角度出发，将 d 阶张量定义为有着 d 个索引的 d 维数组，用大写英文字母表示张量和矩阵，例如X、Y。用 $X(x_1,\dots,x_d)$ 表示张量 X 的索引为 $(x_1,\dots,x_d)$ 的元素。对于张量$X\in\mathbb{R}^{n_1\times\cdots\times n_d}$以及$i=1,\dots,d$，我们称 n_i 为张量 X第 i 个维度的维数。如果固定张量 X 的部分索引而允许其他索引自由变化，则我们得到了 X 的一个子张量，我们采用 Matlab 的记法来表示子张量。以 4 阶张量 $Y\in\mathbb{R}^{n_1\times n_2\times n_3\times n_4}$ 为例，如果固定 Y 的第一个索引为$y_1\in[n_1]$，其中 $[n_1]$ 表示集合 $\{1,2,\dots,n_1\}$，则我们得到了一个 3 阶张量 $X=Y(y_1,:,:,:)\in\mathbb{R}^{n_2\times n_3\times n_4}$，其中 X 的元素为：

$$X(x_1,x_2,x_3)=Y(y_1,x_1,x_2,x_3) \tag{2.1}$$

其余的情况与(2.1)类似可得。为简便记，在不引起混淆的情况下有时我们用符号 $\mathcal{T}$ 来表示一般的 d 阶实张量空间 $\mathbb{R}^{n_1\times\cdots\times n_d}$ 。接下来我们给出一些相关的定义。

定义 2.1（张量内积）：两个张量 $X,Y\in\mathcal{T}$ 的内积定义为：

$$\langle X,Y\rangle=\sum_{i_1,\cdots,i_d=1}^{n_1,\cdots,n_d}X(i_1,\dots,i_d)Y(i_1,\dots,i_d) \tag{2.2}$$

张量 X 的 Frobenius 范数为$||X||_F=\langle X,X\rangle^{1/2}$。

定义 2.2（张量外积）：设 $X\in\mathbb{R}^{n_1\times\cdots\times n_d},Y\in\mathbb{R}^{n_{d+1}\times\cdots\times n_{d+s}}$，张量 X 和 Y 的外积是一个$d+s$ 阶张量 Z，记为 $Z=X\otimes Y$， Z 的元素定义为：

$$Z(z_1,\dots,z_d,z_{d+1},\dots,z_{d+s})=X(z_1,\dots,z_d)Y(z_{d+1},\dots,z_{d+s}) \tag{2.3}$$

定义 2.3（秩 1 张量）：如果一个张量 $X\in\mathcal{T}$ 可以写成 d 个向量的外积，则称张量 X 是秩 1 的，即：

$$X=a_1\otimes a_2\otimes\cdots\otimes a_d,\ \ a_i\in\mathbb{R}^{n_i} \tag{2.4}$$

任意 $X \in \mathcal{T}$ 都可以分解成有限个秩 1 张量的和。令 e_i^n 为 $\mathbb{R}^n$ 中第 i 个分量为 1，其余分量均为 0 的单位向量，则有平凡的秩 1 分解：

$$X = \sum_{i_1,\cdots,i_d=1}^{n_1,\cdots,n_d} X(x_1, \dots, x_d) e_{x_1}^{n_1} \otimes e_{x_2}^{n_2} \otimes \cdots \otimes e_{x_d}^{n_d} \tag{2.5}$$

张量 X 的更一般的秩 1 分解可以记为：

$$X = \sum_{i=1}^{r} \lambda_i a_{1,i} \otimes a_{2,i} \otimes \cdots \otimes a_{d,i} \tag{2.6}$$

在张量 X 所有的秩 1 分解中，使得这些秩 1 项的数目 r 达到最小的分解称为张量的 CP 分解，这个最小的正整数 r 即为张量的 CP 秩。

定义 2.4（张量的 CP 秩）：对$X \in \mathcal{T}$， X 的 CP 秩定义为：

$$rank_{CP}(X) := min\{r \in \mathbb{N}: X = \sum_{i=1}^{r} \lambda_i a_{1,i} \otimes a_{2,i} \otimes \cdots \otimes a_{d,i}\} \tag{2.7}$$

对于 2 阶张量即矩阵，CP 秩等价于通常意义下的矩阵的秩。但高阶张量 CP 秩的计算在大多数情况下是 NP 困难的[11]，且高阶张量的 CP 秩不是下半连续的函数，即一个 CP 秩小于等于 r 的张量序列有可能收敛到 CP 秩大于 r 的张量[24]，这使得高阶张量的最佳低 CP 秩逼近问题通常是病态的。

与 CP 分解不同，张量的 Tucker 分解将张量 $X \in \mathcal{T}$ 分解为一个核张量与 d 个矩阵的模 i 乘积。我们先给出张量模 i 乘积和模 i 展开的定义。

定义 2.5（模 $\boldsymbol{i}$ 乘积）：设 $X \in \mathbb{R}^{n_1\times\cdots\times n_i\times\cdots\times n_d}$，$A_i \in \mathbb{R}^{m_i\times n_i}$。张量 X 与矩阵A_i 的模 i 乘积是一个张量 $B \in \mathbb{R}^{n_1\times\cdots\times m_i\times\cdots\times n_d}$，其元素为：

$$B(x_1, \dots, x_i, \dots, x_d) = \sum_{a=1}^{n_i} A_i(x_i, a) X(x_1, \dots, a, \dots, x_d) \tag{2.8}$$

记为$B = X \times_i A_i$。

定义 2.6（模 $\boldsymbol{i}$ 展开）：张量 $X \in \mathcal{T}$的模 i 展开是一个大小为 $n_i \times \prod_{j\neq i} n_j$ 的矩阵，记为 $X_{(i)}$，其元素为$X_{(i)}\big(x_i, \pi(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d)\big) = X(x_1, \dots, x_d)$，其中 π 将指标$(x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_d)$一对一的映射到集合 $\{j \in \mathbb{N}: 0 < j \leq \prod_{j\neq i} n_j\}$ 之中。

张量的 Tucker 分解将 d 阶张量 $X \in \mathcal{T}$ 分解为一个核张量 $S \in \mathbb{R}^{r_1\times\cdots\times r_d}$ 与 d 个列正交矩阵 $\{U_i \in \mathbb{R}^{n_i\times r_i}: i = 1, \dots, d\}$ 的模 i 乘积，即：

$$X = S \times_1 U_1 \times_2 \cdots \times_d U_d \tag{2.9}$$

式(2.9)也称为张量的高阶 SVD (HOSVD)，有关 HOSVD 的计算方法和其他性质可以参考文献[22]的第 12 章以及文献[23]。

我们称由核张量 S 的维数组成的向量 $(r_1, r_2, \dots, r_d)$ 为张量 X 的 Tucker 秩，记为：

$$rank_T(X) = (r_1, r_2, \dots, r_d) \tag{2.10}$$

与 CP 秩不同的是，张量的 Tucker 秩是一个 d 维整数向量，且容易验证：

$$r_i = rank\left(X_{(i)}\right), i = 1, \dots, d \tag{2.11}$$

即张量的 Tucker 秩的第 i 个分量等于对张量做模 i 展开后得到的矩阵的秩，所以 Tucker 秩和 Tucker 分解的计算是数值稳定的。当 $d = 2$ 时，Tucker 秩同样等价于矩阵的秩。

第二节 张量的 Tensor Train 分解

在 2.1 节中我们简要的介绍了张量的 CP 分解和 Tucker 分解。从存储量的角度来看，CP 分解的存储量与张量的阶数呈理想的线性关系。假设$X \in \mathbb{R}^{n\times\cdots\times n}$，则 CP 分解的存储量为 $O(ndr)$，这里 r 是 X 的 CP 秩。但由于大多数情况下 CP 秩的计算是 NP 困难的，且 CP 秩不是下半连续的，所以在实际应用中 CP 分解的计算往往很不稳定。与 CP 分解相比，Tucker 分解给出了一种数值稳定的分解方式，但如果假设X 的 Tucker 秩为$(r, r, \dots, r)$，那么 Tucker 分解所需要的存储量为 $O(r^d + dnr)$，与张量的阶数依然呈指数关系。

CP 分解的不稳定性和 Tucker 分解存储量的指数增长限制了二者在实际问题中的应用，而最近文献[14]提出的 Tensor Train 分解则在一定程度上解决了这些问题。在介绍 TT 分解之前，我们先给出两个相关的定义。

定义 2.7（张量的 i 阶展开）：设张量$X \in \mathcal{T}$，定义 X 的 i 阶展开是一个大小为$(n_1 \cdots n_i) \times (n_{i+1} \cdots n_d)$的矩阵，记为$X_{[i]}$，其元素为：

$$X_{[i]}\left(\pi_1(x_1, \dots, x_i), \pi_2(x_{i+1}, \dots, x_d)\right) = X(x_1, \dots, x_d) \tag{2.12}$$

其中映射π_1和π_2同样将多个索引一一的映射为单个索引。

定义 2.8（张量的 TT 乘积）：对于$X \in \mathbb{R}^{n_1\times\cdots\times n_N}, Y \in \mathbb{R}^{n_N\times\cdots\times n_{N+s}}$，定义 X 和 Y 的 TT 乘积为一个 $N+s-1$ 阶张量，记为 $Z = X \odot Y$，其元素为：

$$Z(x_1,\dots,x_{N-1},x_{N+1},x_{N+s}) = \sum_{a=1}^{n_N} X(x_1,\dots,x_{N-1},a)Y(a,x_{N+1},\dots,x_{N+s}) \tag{2.13}$$

张量的 TT 分解将 d 阶张量 X 表示为 d 个阶数不超过 3 的张量的 TT 乘积，即：

$$X = G_1 \odot G_2 \odot \cdots \odot G_d \tag{2.14}$$

其中 $G_i \in \mathbb{R}^{r_{i-1}\times n_i\times r_i}, i = 1,\dots,d$ 且 $r_0 = r_d = 1$。

TT 分解也可用矩阵表示为：

$$X(x_1,x_2,\dots,x_d) = G_1(x_1)G_2(x_2)\cdots G_d(x_d) \tag{2.15}$$

其中$G_i(x_i) = G_i(:,x_i,:) \in \mathbb{R}^{r_{i-1}\times r_i}$是张量 G_i固定第二个索引为x_i得到的矩阵。

下文中我们沿用更紧凑的写法(2.14)。在 TT 分解中，我们称$(r_1,\dots,r_{d-1})$为给定分解 $G_1 \odot G_2 \odot \cdots \odot G_d$ 的 TT 秩。通过对(2.14)式等号两边做 i 阶展开，从等式：

$$X_{[i]} = (G_1 \odot \cdots \odot G_i)_{[i]}(G_{i+1} \odot \cdots \odot G_d)_{[1]} \tag{2.16}$$

中可以看出：

$$r_i \geq rank\left(X_{[i]}\right), i = 1,\dots,d-1 \tag{2.17}$$

如果对所有的 $i = 1,\dots,d-1$ 都有 $r_i = rank\left(X_{[i]}\right)$，则此时该分解的 TT 秩达到了最小，我们就称分解 $X = G_1 \odot G_2 \odot \cdots \odot G_d$ 是最优的。我们定义这个最小的 TT 秩即为张量 X 的 TT 秩，也就是：

$$rank_{TT}(X) = \left(rank\left(X_{[1]}\right),\dots,rank\left(X_{[d-1]}\right)\right) \tag{2.18}$$

张量的最优 TT 分解总是存在的，且可以由文献[14]中的 TT-SVD 算法来求解。

TT 分解主要的优点在于 TT 分解的计算是数值稳定的。假设$X \in \mathbb{R}^{n\times\cdots\times n}$的 TT 秩为 $rank_{TT}(X) = (r,\dots,r)$，则 TT 分解需要的存储量为 $O(dnr^2)$，与阶数 d 呈比较理想的线性关系。除此之外 TT 秩意义下的低秩逼近问题总是有解的，且相比于 Tucker 秩，其计算量会大大减少。