arXiv:2302.03886v1 [cs.DS] 8 Feb 2023

A张量分解的近似最优核心形状 s

Mehrdad Ghadiri [*1] **Matthew Fahrbach** [*2] **Gang Fu** [2] **Vahab Mirrokni** [2]

摘要

本文研究了组合优化问题：为尺寸约束Tucker分解寻找最优的核心张量形状，也称为多线性秩。我们利用与高阶奇异值的联系，提出一种算法，并为其重构误差给出可证明的近似保证。具体而言，我们引入了一种新颖的*Tucker* 装箱问题，证明该问题为 NP 难问题，并通过将其归约为带有拟阵约束的二维背包问题，给出一种多项式时间近似方案。我们还将这些技术推广到树张量网络分解。我们使用整数规划求解器实现该算法，并表明其解的质量可与采用真实Tucker分解损失进行逐步计算的贪心算法相媲美（有时甚至更好），而运行速度最高可快 1000 倍。

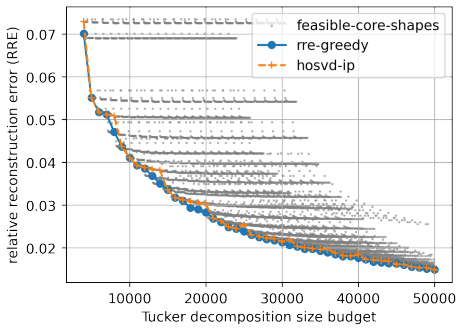

图 *1.* 核心形状的帕累托前沿$\mathbf{r} \in [20]^3$ 对应高光谱张量 $\mathcal{X} \in \mathbb{R}^{1024\times1344\times33}$。绘制了相对重构误差（RRE），即 $L(\mathcal{X}, \mathbf{r})/\|\mathcal{X}\|_F^2$，随压缩率变化的曲线。RRE 贪心算法在每一步计算 Tucker 分解来构建核心形状。HOSVD-IP 是采用整数规划的算法 2，通过对高阶奇异值求解一个替代装箱问题来构建核心形状。

1. 引言

低秩张量分解是现代机器学习工具箱中的强大工具。与低秩矩阵分解一样，它在科学计算、数据挖掘和信号处理领域有着无数应用（Kolda和Bader，2009年；Sidiropoulos 等人，2017），例如数据流中的异常检测（Jang 和 Kang，2021年），以及在移动设备上压缩卷积神经网络，以加快推理速度并降低功耗（Kim 等人，2016年）。

最广泛使用的张量分解是典范多元（CP）分解、Tucker分解和张量列分解（Oseledets，2011）——后两者都是树张量网络（Kr¨amer，2022年）的实例。CP 分解将张量分解为 r 秩一张量之和。Tucker分解则指定每个维度 R_n 的秩 n，并依靠核心张量 $\mathcal{G} \in \mathbb{R}^{R_1\times\cdots\times R_N}$ 来重构分解。多线性秩 $\mathbf{r} = (R_1, \ldots\ldots, R_N)$ 这一概念让实际使用者陷入困境，因为可行核心形状可能呈指数级增长。此外，在这一状态空间中进行搜索的成本可能高得令人难以承受，因为评估核心形状的实际质量需要计算Tucker分解；对于大型张量，这可能耗费数小时，并占用数百GB内存。例如，在MATLAB张量工具箱（Bader和Kolda，2022年）中，我们需要在计算尺寸约束Tucker分解之前，预先指定核心形状参数秩。

在实践中，最常用的 Tucker分解 算法是**r**-截断高阶奇异值分解（高阶奇异值分解）De Lathauwer 等人(2000a)提出的方法、Vannieuwenhoven等人 (2012)提出的顺序截断 ST-HOSVD，以及高阶正交迭代（HOOI），它是一种结构化的交替最小二乘算法。

我们探讨低秩张量分解中一个简单但基础的离散优化问题：

> 如果对 $\mathcal{X}$ 进行 *Tucker*分解，最多可使用 c 个参数，那么哪种核心张量形状能使重构误差最小?

这是最佳秩矩阵近似问题的多线性推广。r 尽管它与低秩矩阵分解有许多相似之处，但张量秩相关问题与其矩阵对应问题可能截然不同，也更具挑战性。例如，计算实值张量的CP秩是NP难的（Hillar和Lim，2013）。

[*]共同贡献 [1]Georgia Tech [2]Google Research。通信作者：Mehrdad Ghadiri <mghadiri3@gatech.edu>。

1.1. 我们的贡献与技术

下面总结本文的主要贡献：

1. 我们形式化了尺寸受限 Tucker 分解的核心张量形状问题，并引入了*Tucker* 装箱问题，并证明该问题是 NP难的。我们开发的近似算法建立在以下关系之上：秩为$\mathbf{r}$ 的 Tucker 分解的最优重构误差与其高阶奇异值的多维尾和之间存在关系（De Lathauwer 等人，2000a；Hackbusch，2019年）。

2. 我们为代理 Tucker 装箱问题（定理 4.6）设计了一种多项式时间近似方案（PTAS），证明只需考虑核心张量成本与因子矩阵成本之间的少量预算划分即可。经过简单变换，每个预算划分子问题都可归约为一个带有分划拟阵约束的二维背包问题。我们使用Grandoni 等人（2014）提出的 PTAS 来求解这些背包问题；在实践中，也可以使用整数线性规划。

3. 我们将我们的方法扩展到树张量网络，该网络推广了Tucker分解、张量列分解和层次 Tucker 分解。在此过程中，我们综合了数学和物理学界关于树张量的多项研究成果，并为计算机科学家提供了简明介绍。

4. 最后，我们在四个真实世界张量上展示了基于 Tucker 打包的核心形状求解器的有效性。我们的 HOSVD-IP 算法与使用真实相对重构误差的贪心算法相比具有竞争力（有时甚至表现更好），同时运行速度最高可快 1000 倍。

1.2. 相关工作

核心形状约束。 De Lathauwer 等人 (2000b年) 提出了计算给定秩-$\mathbf{r}$ 张量近似最佳解的问题，其预先指定的 核心形状 $\mathbf{r}$，并展示了先使用截断 HOSVD 初始化分解、再运行 HOOI 等迭代方法的益处。Eld´en & Savas (2009年); Ishteva 等人(2009年; 2011;2013); Eld´en &Dehghan (2022年) 针对秩-(r_1, r_2, r_3) 分解研究了这一问题，并开发了一系列先进算法：格拉斯曼流形上的牛顿法、黎曼流形上的信赖域方法、用于对称张量的 Jacobi 旋转，以及 Krylov 型迭代方法。然而，所有这些工作都致力于优化固定核心形状下的张量分解，而不是优化核心张量形状本身。

Ehrlacher 等人（2021年）和Xiao 和 Yang （2021年）最近探索了 秩自适应 HOOI 方法，以找到最小的核心形状，使 Tucker分解达到目标重构误差。他们还利用了高阶奇异值分解的性质，但没有对返回的*Tucker*分解的大小施加硬性约束。 Hashemizadeh 等人（2020年）将 图 1 中的 RRE 贪心算法推广到张量网络，同时处理秩和大小约束。

低秩张量分解。Song 等人（2019年）针对多种低秩张量分解（包括 CP 和 Tucker 分解），提出了关于弗罗贝尼乌斯范数的多项式时间（$1+\varepsilon$）近似算法。Frandsen 和 Ge（2022年）证明，如果一个三阶张量具有精确的 Tucker 分解，那么经过适当正则化的损失景观中的所有局部极小值都是全局最优解。近期有多项研究在流式模型（Traor´e 等人，2019年；Sun 等人，2020年）和滑动窗口模型（Jang 和 Kang，2021年）中研究 Tucker 分解。Zhou 等人（2014年）、Cheng 等人（2016年）、Battaglino 等人（2018年）、Malik 和 Becker（2018年）、Che 和 Wei（2019年）、Ma 和 Solomonik（2021年）、Larsen 和 Kolda（2022年）、Fahrbach 等人（2022年）以及 Malik（2022年）提出了基于草图的快速随机低秩张量分解算法。

2. 预备知识

符号约定。 张量的阶 是其维数。标量用普通小写字母表示 $x \in \mathbb{R}$，向量用粗体小写字母表示$\mathbf{x} \in \mathbb{R}^n$，矩阵用粗体大写字母表示$\mathbf{X} \in \mathbb{R}^{m\times n}$，高阶张量用粗体花体字母表示 $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1\times\cdots\times I_N}$。索引集的大小用普通大写字母表示，例如$[N] = \{1, 2, \ldots, N\}$。向量$\mathbf{x}$的第 i个元素记为 x_i，矩阵$\mathbf{X}$ 的第（i，j）个元素记为 x_{ij}，三阶张量 $\boldsymbol{\mathcal{X}}$ 的第（i，j，k）个元素记为 x_{ijk}。

张量积。 张量的纤维是通过固定除一个索引以外的所有索引得到的向量。例如，若 $\boldsymbol{\mathcal{X}} \in \mathbb{R}^3$，则列纤维、行纤维和管纤维分别记为$\mathbf{x}_{:jk}$、$\mathbf{x}_{i:k}$和$\mathbf{x}_{ij:}$。 n $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1\times\cdots\times I_N}$。张量的模展开 是矩阵$\mathbf{X}_{(n)} \in \mathbb{R}^{I_n\times I_1\ldots I_{n-1}I_{n+1}\ldots I_N}$，它将 $\boldsymbol{\mathcal{X}}$ 的模-n 纤维作为$\mathbf{X}_{(n)}$的列，并按索引的字典序排列。

我们将张量 $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1\times\cdots\times I_N}$与矩阵 $\mathbf{A} \in \mathbb{R}^{J\times I_N}$ $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_n \mathbf{A}$ $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1\times\cdots\times I_{n-1}\times J\times I_{n+1}\times\cdots\times I_N}$ 的 n-模乘积 记作 $\mathbf{A}$，其中 。该运算将 $\boldsymbol{\mathcal{X}}$ 的每个 n -模纤维乘以 $\mathbf{A}$，并可按元素表示为

$$(\boldsymbol{\mathcal{X}} \times_n \mathbf{A})_{i_1\ldots i_{n-1}ji_{n+1}\ldots i_N} = \sum_{i_n=1}^{I_n} x_{i_1i_2\ldots i_N} a_{ji_n}.$$

两个张量的内积 $\boldsymbol{\mathcal{X}}$，$\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1\times\cdots\times I_N}$ 是

它们各元素乘积之和：

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

张量 $\mathcal{X}$ 的弗罗贝尼乌斯范数为 $\|\mathcal{X}\|_{\mathrm{F}} = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}$。

Tucker分解。张量 $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ 的*Tucker*分解将 $\mathcal{X}$ 分解为一个核心张量$\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$ 和 N 因子矩阵$\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$。我们将 $\mathbf{r} = (R_1, \ldots, R_N)$ 称为核心形状，它也称为该分解的多线性秩或截断。我们用以下符号表示最优秩$\mathbf{r}$ Tucker分解的损失：

$$L(\mathcal{X}, \mathbf{r}) \stackrel{\text{def}}{=} \min_{\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}} \|\mathcal{X} - \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \cdots \times_N \mathbf{A}^{(N)}\|_{\mathrm{F}}^2.$$

3. 化简为 HOSVD Tucker 打包

3.1. 高阶奇异值分解

我们首先回顾 DeLathauwer、De Moor 和 Vandewalle(2000a)关于高阶奇异值分解（HOSVD）的开创性工作。

定理3.1 (De Lathauwer 等人 (2000a，定理2))。 任意张量 $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ 都可以写成

$$\mathcal{X} = \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \cdots \times_N \mathbf{U}^{(N)},$$

其中，每个$\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times I_n}$ 都是一个正交矩阵，而 $\mathcal{S} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ 是一个张量，其子张量为 $\mathcal{S}_{i_n = \alpha}$，通过固定第 n个索引为 α得到，并具有以下性质：

1. 全正交性： 对于满足 $\alpha \neq \beta$的 n、 α 和β 的所有可能取值，两个子张量 $\mathcal{S}_{i_n=\alpha}$ 和 $\mathcal{S}_{i_n=\beta}$彼此正交，即当 $\alpha \neq \beta$时，； *2.* 排序： 对于 n、 $\|\mathcal{S}_{i_n=1}\|_{\mathrm{F}} \geq \|\mathcal{S}_{i_n=2}\|_{\mathrm{F}} \geq$的所有取值 $\cdots \geq \|\mathcal{S}_{i_n=I_n}\|_{\mathrm{F}} \geq 0.$

此外，值 $\|\mathcal{S}_{i_n=i}\|_{\mathrm{F}}$（记作 $\sigma_i^{(n)}$ ）是模式-n 展开 $\mathbf{X}_{(n)}$ 的奇异值，而 $\mathbf{U}^{(n)}$ 的列是左奇异向量。

接下来，我们介绍`TuckerHOSVD` 算法。在计算秩为$\mathbf{r}$的 Tucker 分解时（Kolda &Bader，2009年），这是一种广泛使用的初始化策略，即核心形状$\mathbf{r}$ 是预先确定的。

算法1`TuckerHOSVD`

输入： $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$，核心形状 $\mathbf{r} = (R_1, \ldots, R_N)$

1: **对于** $n = 1$到 N **执行**
2: $\quad \mathbf{A}^{(n)} \leftarrow R_n$ 张量$\mathbf{X}_{(n)}$的前几个左奇异向量
3: $\mathcal{G} \leftarrow \mathcal{X} \times_1 \mathbf{A}^{(1)} \times_2 \cdots \times_N \mathbf{A}^{(N)}$
4: **返回** $\mathcal{G}$，$\mathbf{A}^{(1)}$，$\mathbf{A}^{(2)}$，……，$\mathbf{A}^{(N)}$

TuckerHOSVD 的输出具有以下误差保证（De Lathauwer 等人，2000a；Hackbusch，2019年）。这些界表明，在满足 Tucker 分解大小约束的前提下优化核心张量形状时，可以改为最小化计算成本更低的代理损失函数。我们在附录 A.1中给出这些结果的完整证明，该证明基于定理3.1。

定理 3.2 (De Lathauwer 等人(2000a，性质10);Hackbusch (2019， 10 2 $\mathcal{X} \in$定理 .)). 对于 $\mathbb{R}^{I_1 \times \cdots \times I_N}$ 任意张量及其核心形状 $\mathbf{r} \in [I_1] \times \cdots \times [I_N]$，设`TuckerHOSVD`($\mathcal{X}$, $\mathbf{r}$) 的输出为 $\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$ 且$\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$，对每个 $n \in [N]$。若令

$$\widehat{\mathcal{X}}_{\mathrm{HOSVD}(\mathbf{r})} \stackrel{\text{def}}{=} \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \cdots \times_N \mathbf{A}^{(N)} \tag{1}$$

表示重构的$\mathbf{r}$截断张量，则

$$\begin{aligned} \|\mathcal{X} - \widehat{\mathcal{X}}_{\mathrm{HOSVD}(\mathbf{r})}\|_{\mathrm{F}}^2 &\leq \sum_{n=1}^{N} \sum_{i_n=R_n+1}^{I_n} \left(\sigma_{i_n}^{(n)}\right)^2 \\ &\leq N \cdot L(\mathcal{X}, \mathbf{r}). \end{aligned}$$

此外，我们有 $L(\mathcal{X}, \mathbf{r}) \leq \|\mathcal{X} - \widehat{\mathcal{X}}_{\mathrm{HOSVD}(\mathbf{r})}\|_{\mathrm{F}}^2$.

定理 3.2表明，以下函数是最优秩-$\mathbf{r}$Tucker分解的重构误差的有意义代理。

定义 3.3。 将核心形状 r 的**代理损失** 定义为

$$\widetilde{L}(\mathcal{X}, \mathbf{r}) \stackrel{\text{def}}{=} \sum_{n=1}^{N} \sum_{i_n=R_n+1}^{I_n} \left(\sigma_{i_n}^{(n)}\right)^2. \tag{2}$$

总结目前的内容，对于任意核心形状 $\mathbf{r} \in [I_1] \times \cdots \times [I_N]$，$\widetilde{L}(\mathcal{X}, \mathbf{r})$我们可以保证 $1/N \cdot L(\mathcal{X}, \mathbf{r}) \leq L(\mathcal{X}, \mathbf{r}) \leq L(\mathcal{X}, \mathbf{r})$。完整细节请参阅 附录 A.2 。

3.2. Tucker 装箱问题

接下来，注意到 $\mathcal{X}$ 的所有模-n展开的奇异值平方和为

$$\sum_{n=1}^{N} \sum_{i_n=1}^{I_n} \left(\sigma_{i_n}^{(n)}\right)^2 = \sum_{n=1}^{N} \|\mathbf{X}_{(n)}\|_{\mathrm{F}}^2 = N \|\mathcal{X}\|_{\mathrm{F}}^2.$$

这意味着，我们可以通过考虑代理损失的补来求解奇异值打包问题。以下引理概括了 定理 3.2 中关于截断HOSVD误差保证的内容。

引理 3.4。 对于任意张量 $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ 以及 *Tucker*分解大小的预算 $c \geq 1 + \sum_{n=1}^{N} I_n$，令可行核心形状集合为

$$F = \left\{ \mathbf{r} \in [I_1] \times \cdots \times [I_N] : \prod_{n=1}^{N} R_n + \sum_{n=1}^{N} I_n R_n \leq c \right\}.$$

则有

$$\widetilde{\mathbf{r}}^* \overset{\text{def}}{=} \arg\min_{\mathbf{r}\in F} \widetilde{L}(\mathcal{X},\mathbf{r}) = \arg\max_{\mathbf{r}\in F} \sum_{n=1}^{N}\sum_{i_n=1}^{R_n}\left(\sigma_{i_n}^{(n)}\right)^2.$$

此外，如果$\mathbf{r}^* \overset{\text{def}}{=} \arg\min_{\mathbf{r}\in F} L(\mathcal{X},\mathbf{r})$是一个最优的预算约束核心形状，那么 $L(\mathcal{X},\widetilde{\mathbf{r}}^*) \le N \cdot L(\mathcal{X},\mathbf{r}^*)$。

为了找到一种核心形状，使其最优 Tucker 分解能够逼近满足大小约束的最优损失 $L(\mathcal{X},\mathbf{r}^*)$，我们求解引理 3.4中的最大化问题。优化这一代理目标的开销，远低于将秩r Tucker 分解求解器作为子程序的方法。我们通过引入下面更一般的问题，将这一思路形式化。

定义 3.5（Tucker 打包问题）。给定一个形状 $(I_1,\dots,I_N) \in \mathbb{Z}_{\ge 1}^N$, N非递增序列 $\mathbf{a}^{(n)} \in \mathbb{R}_{\ge 0}^{I_n}$以及一个预算 $c \ge 1$, *Tucker* 装箱问题 要求找出一个核心形状 $\mathbf{r} \in [I_1]\times\cdots\times[I_N]$，使其满足：

$$\text{maximize} \quad \sum_{n=1}^{N}\sum_{i_n=1}^{R_n} a_{i_n}^{(n)} \tag{3}$$

$$\text{subject to} \quad \prod_{n=1}^{N} R_n + \sum_{n=1}^{N} I_n R_n \le c \tag{4}$$

我们还将目标函数记为 $f(\mathbf{r}) \overset{\text{def}}{=} \sum_{n=1}^{N}\sum_{i_n=1}^{R_n} a_{i_n}^{(n)}$.**定理 3.6.** *Tucker* 装箱问题是NP难。

我们在附录A.3 中通过从等分问题进行一次复杂的归约来证明这一结果（例如，参见Garey和Johnson（1979，SP12）。

NP难性促使我们需要高效的近似算法。在第4节中，我们提出一种多项式时间近似方案（PTAS），用于 Tucker 装箱问题。全多项式时间近似方案（FPTAS）是否存在，仍是留待未来研究解决的一项富有挑战性的开放问题。

总之，由于Tucker 打包是代理损失最小化的补问题，我们必须量化针对打包问题的（$1-\varepsilon$）-近似会如何影响代理损失中产生的误差。我们在附录A.4 中详细说明这一点，并在下文给出主要思路。

引理3.7。设$\mathbf{r} \in [I_1]\times\cdots\times[I_N]$是任意一个能使 *Tucker* 装箱问题达到$(1-\varepsilon/N)$近似比的核心形状。那么，我们有相对重构误差$(\mathcal{X},\mathbf{r}) \le N\cdot$相对重构误差$(\mathcal{X},\mathbf{r}^*)+\varepsilon$，其中相对重构误差$(\mathcal{X},\mathbf{r}) := L(\mathcal{X},\mathbf{r})/\|\mathcal{X}\|_F^2$。

备注 3.8。我们可以通过（1）寻找近似最优的核心形状，（2）运行 `TuckerHOSVD` 来初始化Tucker分解，并（3）使用交替最小二乘法（ALS）改进张量分解。这类似于 k均值++改进Lloyd算法的方式（Arthur和Vassilvitskii，2006年）。

4. 算法

4.1. 预热：与多重选择背包的联系

首先，考虑 Tucker 装箱问题的一个简化版本，该版本只考虑核心张量的大小，即因子矩阵不占用任何预算。我们将证明，经过两次简单变换后，这个新问题可归约为多重选择背包问题，[1] 该问题是 NP难（Pisinger，1995），但存在全多项式时间近似方案（Lawler，1977）。

具体而言，该优化问题为

$$\text{maximize} \quad \sum_{n=1}^{N}\sum_{i_n=1}^{R_n} a_{i_n}^{(n)} \tag{5}$$

$$\text{subject to} \quad \prod_{n=1}^{N} R_n \le c \tag{6}$$

前缀和变换。为了更接近 0-1 背包问题，对每个 $n\in[N]$ 和 $i_n \in [I_n]$，通过对 $a_{i_n}^{(n)}$ 求前缀和来定义新的系数：

$$p_{i_n}^{(n)} \overset{\text{def}}{=} \sum_{j_n=1}^{i_n} a_{j_n}^{(n)}.$$

这个仅考虑核心大小的 Tucker 打包问题可以重新表述为以下整数规划：

$$\begin{aligned}
\text{maximize} \quad & \sum_{n=1}^{N}\sum_{i_n=1}^{I_n} p_{i_n}^{(n)} x_{i_n}^{(n)} \\
\text{subject to} \quad & \prod_{n=1}^{N}\sum_{i_n=1}^{I_n} i_n x_{i_n}^{(n)} \le c \qquad (7)\\
& \sum_{i_n=1}^{I_n} x_{i_n}^{(n)} = 1 \qquad \forall n\in[N] \\
& x_{i_n}^{(n)} \in \{0,1\} \qquad \forall n\in[N], i_n\in[I_n]
\end{aligned}$$

为简洁起见，我们对 i_n 进行优化，而不是对 R_n 进行优化。

对数变换。接下来，将约束条件（7）替换为线性不等式

$$\sum_{n=1}^{N}\sum_{i_n=1}^{I_n} \log(i_n) x_{i_n}^{(n)} \le \log(c).$$

[1]多重选择背包问题是一种 0-1 背包问题，其中物品被划分为 N 类别，并且必须从每个类别中恰好选取一个物品（Sinha与Zoltners，1979）。

这种替换是有效的，因为在任何可行解中，对于每个 $n \in [N]$，$x_1^{(n)}$、$x_2^{(n)}$、……、$x_{I_n}^{(n)}$ 中恰有一个等于 1，其余均为 0。综合以上内容，这个仅考虑核心大小的 Tucker 装箱问题就是以下多重选择背包问题：

$$\begin{aligned} \text{maximize} \quad & \sum_{n=1}^{N} \sum_{i_n=1}^{I_n} p_{i_n}^{(n)} x_{i_n}^{(n)} && (8) \\ \text{subject to} \quad & \sum_{n=1}^{N} \sum_{i_n=1}^{I_n} \log(i_n) x_{i_n}^{(n)} \le \log(c) \\ & \sum_{i_n=1}^{I_n} x_{i_n}^{(n)} = 1 \qquad \forall n \in [N] \\ & x_{i_n}^{(n)} \in \{0,1\} \qquad \forall n \in [N], i_n \in [I_n] \end{aligned}$$

定理 4.1 (Lawler1977)。 存在一种算法，可在时间和空间 $1-\varepsilon$- 近似求解问题 (8)，其复杂度为 $O($ $N^2 \varepsilon^{-1} \sum_{n=1}^{N} I_n)$。

定理 4.1 中的 多重选择背包全多项式时间近似方案会对目标函数中的所有系数 $p_{i_n}^{(n)}$ 统一缩放并取整，然后使用动态规划。

4.2. Tucker 装箱问题的多项式时间近似方案

现在我们考虑 Tucker分解的实际成本，即核心张量和因子矩阵的大小。首先，我们介绍一种简单的网格搜索算法，用于求解一大类可行解（即向下闭集）的一般情形下的近似 Tucker 打包问题。这涵盖了 Tucker 打包问题，并有助于我们在第 5 节中将结果推广到树张量网络。

定义 4.2。 对于任意 $N \ge 1$ 和 $(I_1, \ldots\ldots, I_N) \in \mathbb{Z}_{\ge 1}^N$，令 $F \subseteq [I_1] \times \cdots \times [I_N]$。集合 F是向下封闭的，如果对于任意一对 $(R_1, \ldots\ldots, R_N), (R'_1, \ldots\ldots, R'_N) \in [I_1] \times \cdots \times [I_N]$，只要满足 $R'_n \le R_n$对所有 $n \in [N]$均成立，则 $(R_1, \ldots\ldots, R_N) \in F$蕴含 $(R'_1, \ldots\ldots, R'_N) \in F$。

引理 4.3。 设 $0 < \varepsilon \le 1$和 $F \subseteq [I_1] \times \cdots \times [I_N]$ 是向下封闭的。对于每个 $n \in [N]$，定义

$$S_n^{(\varepsilon)} = \left\{ \lceil (1+\varepsilon)^k \rceil : k \in \mathbb{Z}_{\ge 0}, \lceil (1+\varepsilon)^k \rceil \le I_n \right\}.$$

设 $\mathbf{r}^*$ 是广义问题的一个最优解

$$\begin{aligned} \text{maximize} \quad & \sum_{n=1}^{N} \sum_{i_n=1}^{R_n} a_{i_n}^{(n)} && (9) \\ \text{subject to} \quad & (R_1, \ldots, R_N) \in F \end{aligned}$$

并令$\mathbf{r}^{(\varepsilon)} = (R_1^{(\varepsilon)}$、……、$R_N^{(\varepsilon)})$ 为以下问题的最优解

$$\begin{aligned} \text{maximize} \quad & \sum_{n=1}^{N} \sum_{i_n=1}^{R_n} a_{i_n}^{(n)} && (10) \\ \text{subject to} \quad & (R_1, \ldots, R_N) \in (S_1^{(\varepsilon)} \times \cdots \times S_N^{(\varepsilon)}) \cap F \end{aligned}$$

那么，$f(\mathbf{r}^{(\varepsilon)}) \ge (1+\varepsilon)^{-1} f(\mathbf{r}^*)$。此外，存在一种算法，可以在(10) 的运行时间内求得 $O(\sum_{n=1}^{N} I_n + \varepsilon^{-N} \prod_{n=1}^{N} (1 + \log_2(I_n)))$的最优解。

引理 4.3 中的时间复杂度 关于张量的阶数呈指数级，因此我们现在着手设计主`TuckerPackingSolver`算法，其运行时间为 $O($多项式$(N,$ 对数 $c,$ $I_n))$ 其中 $\varepsilon > 0$为常数。

算法描述。 该算法分为两个阶段：**(1)** 当因子矩阵成本较低时，尝试所有预算分配划分，穷举搜索所有“小型”核心形状；**(2)** 在核心张量大小与因子矩阵成本之间尝试更粗略的划分。在大型阶段，我们证明只需考虑 $O($对数$_{1+\varepsilon} c)$ 种此类划分。每种预算划分都会导出一个形式为（13）的问题，应用前缀和与对数变换后，它会变成熟悉的带分划拟阵约束的二维背包问题：

$$\begin{aligned} \text{maximize} \quad & \sum_{n=1}^{N} \sum_{i_n=1}^{I_n} p_{i_n}^{(n)} x_{i_n}^{(n)} && (11) \\ \text{subject to} \quad & \sum_{n=1}^{N} \sum_{i_n=1}^{I_n} \log(i_n) x_{i_n}^{(n)} \le \log(c_{\text{core}}) \\ & \sum_{n=1}^{N} \sum_{i_n=1}^{I_n} I_n i_n x_{i_n}^{(n)} \le c - c_{\text{core}} \\ & \sum_{i_n=1}^{I_n} x_{i_n}^{(n)} = 1 \qquad \forall n \in [N] \\ & x_{i_n}^{(n)} \in \{0,1\} \qquad \forall n \in [N], i_n \in [I_n] \end{aligned}$$

更一般地，(11) 是一个 d预算约束的拟阵独立集问题（例如参见 Grandoni 等人 (2014），其目标是在满足 d 个背包约束和一个拟阵约束的条件下最大化线性目标函数。回想一下，多维背包问题（即使没有任何拟阵约束）也不存在全多项式时间近似方案，除非 P = = NP（Gens 和 Levner，1979；Korte 和 Schrader，1981）。不过，正如下一个定理所示，它存在多项式时间近似方案。

定理 4.4 (Grandoni 等人 2014，推论 4.4)。对于d预算约束的拟阵独立集问题，存在一个多项式时间近似方案（即一个$(1-\varepsilon)$-近似算法），其运行时间为 $O(m^{O(d^2/\varepsilon)})$，其中 m 是物品的数量。

(11) 中的物品数量为 $m = \sum_{n=1}^{N} I_n$，每个物品对应一种核心形状维度选择。因此，由于 $d = 2$，运行时间为 $O((\sum_{n=1}^{N} I_n)^{O(1/\varepsilon)})$。进而，我们可以据此界定 算法 2的总体运行时间。

注 4.5。 我们可以使用整数线性规划求解器来求解 (11)，而不必使用 Grandoni 等人 (2014)的方法，但只有在我们将核心形状成本与因子矩阵成本分离的情况下才能做到这一点，也就是说，只有在采用 c_{factor} 和 c_{core} 预算划分时才能做到。

算法 2 TuckerPackingSolver

输入： 形状 $(I_1, \dots, I_N) \in \mathbb{Z}^N_{\geq 1}$，$N$ 非递增序列 $\mathbf{a}^{(n)} \in \mathbb{R}^{I_n}_{\geq 0}$，预算 $c \geq 1 + \sum_{n=1}^N I_n$，误差 $\varepsilon > 0$

初始化 $S \leftarrow \varnothing$
对于 $c_{\text{factor}} \in [\lceil 1/\varepsilon \rceil \sum_{n=1}^N I_n]$ **do**
  设 $\mathbf{r}'$ 是以下问题的 $(1-\varepsilon)$ 近似解：

$$\begin{aligned} &\text{最大化} && f(\mathbf{r}) \\ &\text{约束条件为} && \textstyle\prod_{n=}^N R_n \leq c - c_{\text{factor}} \\ & && \textstyle\sum_{n=1}^N I_n R_n \leq c_{\text{factor}} \\ & && R_n \in [\min(\lceil 1/\varepsilon \rceil, I_n)] \qquad \forall n \in [N] \end{aligned} \tag{12}$$

  更新 $S \leftarrow S \cup \{\mathbf{r}'\}$
针对 $k = 0$至 $\lfloor \log_{1\ +\varepsilon} c \rfloor$ **执行**
  设定 $c_{\text{core}} \leftarrow (1+\varepsilon)^k$
  令 $\mathbf{r}^{(k)}$ 为以下问题的 $(1-\varepsilon)$-近似解：

$$\begin{aligned} &\text{最大化} && f(\mathbf{r}) \\ &\text{约束条件为} && \textstyle\prod_{n=}^N R_n \leq c_{\text{core}} \\ & && \textstyle\sum_{n=1}^N I_n R_n \leq c - c_{\text{core}} \\ & && R_n \in [I_n] \qquad \forall n \in [N] \end{aligned} \tag{13}$$

  更新 $S \leftarrow S \cup \{\mathbf{r}^{(k)}\}$
返回 $\operatorname{argmax}_{\mathbf{r}\ \in S} f(\mathbf{r})$

定理 4.6. 若 $0 < \varepsilon < 1/3$，则算法 2 会在时间 $O((\log_{1+\varepsilon}(c) + \lceil 1/\varepsilon \rceil \sum_{n=1}^N I_n)(\sum_{n=1}^N I_n)^{O(1/\varepsilon)})$ 内返回问题 (9) 的 $(1-3\varepsilon)$-近似解。

证明。TuckerPackingSolver 求解两类形状：（1）“小型”解，其中每个 $R_n \leq \lceil 1/\varepsilon \rceil$；（2）“大型”解，其中 $R_n > \lceil 1/\varepsilon \rceil$ 对某个 $n \in [N]$ 成立。

在小型阶段，注意到由于 $R_n \leq \min(\lceil 1/\varepsilon \rceil, I_n)$，因子矩阵成本为 $\sum_{n=1}^N I_n R_n \leq \lceil 1/\varepsilon \rceil \sum_{n=1}^N I_n$。因此，我们可以穷举检查所有形如 $c_{\text{factor}} \in [\lceil 1/\varepsilon \rceil \sum_{n=1}^N I_n]$ 的小型预算划分。每种划分都会导出一个带有划分拟阵的二维背包问题（但物品集合更小），因此使用定理 4.4 为每个子问题求得一个 $(1-\varepsilon)$-近似解。如果 Tucker 装箱问题的一个最优解 $\mathbf{r}^*$ 规模较小，则算法 2 可以恢复一个近似最优目标值。

对于大型阶段，假设最优核心形状 $\mathbf{r}^*$ 有一个维度 $m \in [N]$ 满足 $R^*_m > \lceil 1/\varepsilon \rceil$。算法通过将预算 c 拆分为核心张量的大小和因子矩阵的总成本，间接搜索大形状。一个关键观察是，我们只需检查 $O(\log_{1+\varepsilon} C)$ 种不同的拆分，因为大维度中有足够的松弛量 R^*_m。

接下来，令 $c^*_{\text{core}} = \prod_{n=1}^N R^*_n$，，并令 k^* 为最大的整数，使得 $(1+\varepsilon)^k \leq c^*_{\text{core}}$。定义 $\widehat{c}_{\text{core}} = (1+\varepsilon)^{k^*}$，$\widehat{\ }$ $\widehat{\ }$，并令 $\widehat{\ }\mathbf{r} = (R_1, \dots, R_N)$，其中

$$\widehat{R}_n = \begin{cases} R^*_n & \text{if } n \neq m, \\ \lfloor R^*_n/(1+\varepsilon) \rfloor & \text{if } n = m. \end{cases}$$

$\widehat{\ }$由于我们假设 $R^*_m \geq \lceil 1/\varepsilon \rceil$ 是一个大维度，$R_m = \lfloor R^*_m/(1+\varepsilon) \rfloor \geq \lfloor \lceil 1/\varepsilon \rceil/(1+\varepsilon) \rfloor$. 接下来，注意到

$$\begin{aligned} \prod_{n=1}^N \widehat{R}_n &\leq \frac{1}{1+\varepsilon} \prod_{n=1}^N R^*_n = \frac{c_{\text{core}}}{1+\varepsilon} \\ &< \frac{(1+\varepsilon)^{k^*+1}}{1+\varepsilon} = (1+\varepsilon)^{k^*} = \widehat{c}_{\text{core}}, \end{aligned}$$

且 $\sum_{n=1}^N I_n \widehat{R}_n \leq \sum_{n=1}^N I_n R^*_n \leq c - c^*_{\text{core}} \leq c - \widehat{c}_{\text{core}}$.因此，$\widehat{\ }\mathbf{r}$ 是针对 $k = k^*$ 的（13）的一个可行解。此外，$f(\mathbf{r}^{(k^*)}) \geq (1-\varepsilon) f(\widehat{\ }\mathbf{r})$。

接下来，我们证明 $f(\widehat{\ }\mathbf{r}) \geq (1-2\varepsilon) f(\mathbf{r}^*)$。对于任意 $n \neq m$，$\widehat{\ }$，我们有 $R_n = R^*_n$，因此可得

$$\sum_{i_n=1}^{\widehat{R}_n} a^{(n)}_{i_n} = \sum_{i_n=1}^{R^*_n} a^{(n)}_{i_n}. \tag{14}$$

对于大维度 m，我们有

$$\widehat{R}_m = \left\lfloor \frac{R^*_m}{1+\varepsilon} \right\rfloor \geq \frac{R^*_m}{1+\varepsilon} - 1 = \frac{R^*_m - (1+\varepsilon)}{1+\varepsilon} \cdot \frac{R^*_m}{R^*_m}.$$

假设 $R^*_m \geq \lceil 1/\varepsilon \rceil$于是给出

$$\frac{R^*_m - (1+\varepsilon)}{R^*_m} \geq 1 - \frac{1+\varepsilon}{\lceil 1/\varepsilon \rceil} \geq 1 - \varepsilon - \varepsilon^2.$$

因此，对于任意 $\varepsilon \geq 0$，都有

$$\widehat{R}_m \geq \frac{1 - \varepsilon - \varepsilon^2}{1+\varepsilon} R^*_m \geq (1-2\varepsilon) R^*_m.$$

由于 $a^{(m)}_{i_1} \geq \cdots \geq a^{(m)}_{I_m}$ 是单调不增的，我们有

$$\sum_{i_m=1}^{\widehat{R}_m} a^{(m)}_{i_m} \geq (1-2\varepsilon) \sum_{i_m=1}^{R^*_m} a^{(m)}_{i_m}. \tag{15}$$

结合 (14) 和 (15) 可得 $f(\widehat{\ }\mathbf{r}) \geq (1-2\varepsilon) f(\mathbf{r}^*)$，于是 $f(\mathbf{r}^{(k^*)}) \geq (1-\varepsilon)(1-2\varepsilon) f(\mathbf{r}^*) \geq (1-3\varepsilon) f(\mathbf{r}^*)$，这就证明了近似保证。[2]

最后，运行时间由我们将问题归约为带有划分拟阵约束的二维背包问题，并对每种预算划分应用定理 4.4 得出。□

[2] 要获得一个多项式时间近似方案，只需考虑预算划分 $c_{\text{factor}} \in [\sum_{n=1}^N I^2_n]$，即 `TuckerPackingSolver` 的第 2 行，但正如附录 B 中所解释的，这会导致更长的运行时间。

5. 树张量网络分解

这里我们考虑一种称为树张量网络的通用分解（Oseledets 和 Tyrtyshnikov，2009年；Kr¨amer，2020年），它将 Tucker分解、张量列分解和层次 Tucker 分解作为特例。

定义 5.1（树张量网络）。设 $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ 为任意张量。设 $G = (V, E)$ 为一棵有 N 个叶节点的有根树，其中每个节点 v 对应于一个子集 $S_v \subseteq [N]$。叶节点是 $[N]$ 的 N 个单元素集，而内部节点 v 由 $S_v = \cup_{u \in C_v} S_u$ 递归定义，其中 C_v 是 v 的子节点集合。

每条边 $e \in E$ 都赋有一个整数 $R_e \geq 1$。于是，张量 $\mathcal{X}$ 关于树 G 的（截断）树张量网络分解是如下张量集合，其中每个张量对应一个 $v \in V$。对于每个叶节点，其张量为 $\mathbf{A}^{(v)} \in \mathbb{R}^{I_n \times R_e}$，其中 e 是连接 v 与其父节点的边。对于每个内部节点 v，其张量为 $\mathcal{T}^{(v)} \in \mathbb{R}^{R_{e_1} \times \cdots \times R_{e_k}}$，其中 $E_v = \{e_1, \ldots, e_k\}$ 是与 v 相连的边的集合。

$\widehat{\ }$输出张量 $\mathcal{X}$ 是通过对 G 中的所有“节点张量”沿其对应边进行逐模乘积构造而成的。这些乘积满足交换律和结合律（例如，参见 Kr¨amer (2020年) 中的命题2.17）。

备注 5.2. Tucker分解对应的树张量网络是一棵深度为一的树，而对于分层Tucker分解，它是一棵（近乎）平衡的二叉树（Oseledets 和 Tyrtyshnikov，2009年；Grasedyck，2010年）。

我们在附录C中给出了一个带图示的例子。现在我们将张量展开的定义推广。

定义5.3。对于任意 $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ 和 $S \subseteq [N]$，矩阵化 $\mathbf{X}_{(S)} \in \mathbb{R}^{P \times Q}$，其中 $P = \prod_{n \in S} I_n$ 和 $Q = \prod_{n \in [N] \setminus S} I_n$，是一个矩阵，其元素按照原始索引元组的字典序排列。

下一个定理给出了一个多项式时间算法，用于寻找满足给定 R_e 值且重构误差有界的树张量网络分解。

定理 5 4 2010 ¨ 2020 $\mathcal{X} \in$ 。 $\mathbb{R}^{I_1 \times \cdots \times I_N}$ $G = V E$ R_e $e \in E$ (Grasedyck (); Kramer ())。设 (,)为树张量网络参数，如定义 *5.1*所述。存在一个多项式时间算法，可以找到 $\mathcal{T}^{(v)}$ 使得 $v \in V$(对于叶节点，这些张量是矩阵$\mathbf{A}^{(v)}$)，并满足

$$\|\mathcal{X} - \widehat{\mathcal{X}}\|_{\mathrm{F}}^2 \leq \sum_{v \in V \setminus \{r\}} \sum_{i_v = R_v + 1}^{P_v} \left(\sigma_{i_v}^{(v)}\right)^2 \leq (|V| - 1) \cdot \|\mathcal{X} - \mathcal{X}_{\text{best}}\|_{\mathrm{F}}^2,$$

其中 $\mathcal{X}_{\text{best}}$ 是针对 G 的最佳树张量网络分解，数值 R_e、r 是根节点，$P_v = \prod_{n \in S_v} I_n$，而 $\sigma_i^{(v)}$ 是$\mathbf{X}_{(S_v)}$的第 i个奇异值。

此外，树张量网络分解的大小为 $\sum_{v \in L} I_v R_v + \sum_{v \in J} \prod_{e \in E_v} R_e$，其中 L 是叶节点集合，J 是内部节点集合，R_v 是连接 v 与其父节点的边上的值。

由此可定义 NP 难树张量网络装箱问题，它是 Tucker 打包问题的推广。

定义 5.5（树张量网络装箱）。给定一个形状 $(I_1, \ldots, I_N) \in \mathbb{Z}_{\geq 1}^N$、树 $G = (V, E)$，其叶节点为 $L = [N]$，内部节点为 J，以及 $|V| - 1$ 非递增序列（对应非根节点）$\mathbf{a}^{(v)} \in \mathbb{R}_{\geq 0}^{P_v}$，其中 $P_v = \prod_{n \in S_v} I_n$，并给定预算 $c \geq 1$，树张量网络装箱问题要求找到 $R_v \in \mathbb{Z}_{\geq 1}$，使得 $v \in V \setminus \{r\}$，其中 r 是根节点，N_v 是节点 v的邻居集合，并满足：

$$\text{maximize} \quad \sum_{v \in V \setminus \{r\}} \sum_{i=1}^{R_v} a_i^{(v)} \tag{16}$$

$$\text{subject to} \quad \sum_{v \in J} \prod_{u \in N_v} R_u + \sum_{v \in L} I_v R_v \leq c \tag{17}$$

定理 5.6。存在一种用于树张量网络装箱问题的$(1 - \varepsilon)$-近似算法，其运行时间为$O(\sum_{v \in V \setminus \{t\}} P_v + \varepsilon^{-(|V|-1)} \prod_{v \in V \setminus \{t\}} (1 + \log_2(P_v)))$。

6. 实验

我们比较了几种算法，用于计算四个真实世界张量的尺寸约束 Tucker 分解的核心形状（见表 1）。这些实验展示了~使用代理损失 $L(\mathcal{X}, \mathbf{r})$ 替代真实相对重构误差（RRE）的有效性，无论是在解的质量还是运行时间方面。所有实验均使用 NumPy（Harris 等人，2020年），并在配备 Intel Xeon W-1235 处理器（3.7 GHz, 8.25MB 缓存）和 128GB 内存的设备上进行。

表 *1*。实验中使用的张量数据集统计信息。

张量	形状	SIZE
心脏 MRI	$256 \times 256 \times 14 \times 20$	$18,350,080$
高光谱	$1024 \times 1344 \times 33$	45，416，488
VicRoads	$1084 \times 2033 \times 96$	$211,562,112$
COIL-100	$7200 \times 128 \times 128 \times 3$	$353,894,400$

6.1. 算法

我们考虑的前四种算法基于 HOSVD Tucker 打包：它们计算模式-n 奇异值$\sigma_{i_n}^{(n)}$，并将$\mathbf{a}^{(n)} = \{(\sigma_{i_n}^{(n)})^2\}_{i_n=1}^{I_n}$ 作为其 Tucker 打包实例的输入。第五种是常用的贪心算法，在每一步计算真实损失 $L(\mathcal{X}, \mathbf{r})$。

HOSVD-IP 是`TuckerPackingSolver` 算法，$\varepsilon = 0.25$，但我们使用`scipy.optimize.mlip` 中的整数规划求解器来求解每个预算划分子问题，而非使用Grandoni 等人(2014)提出的多项式时间近似方案。

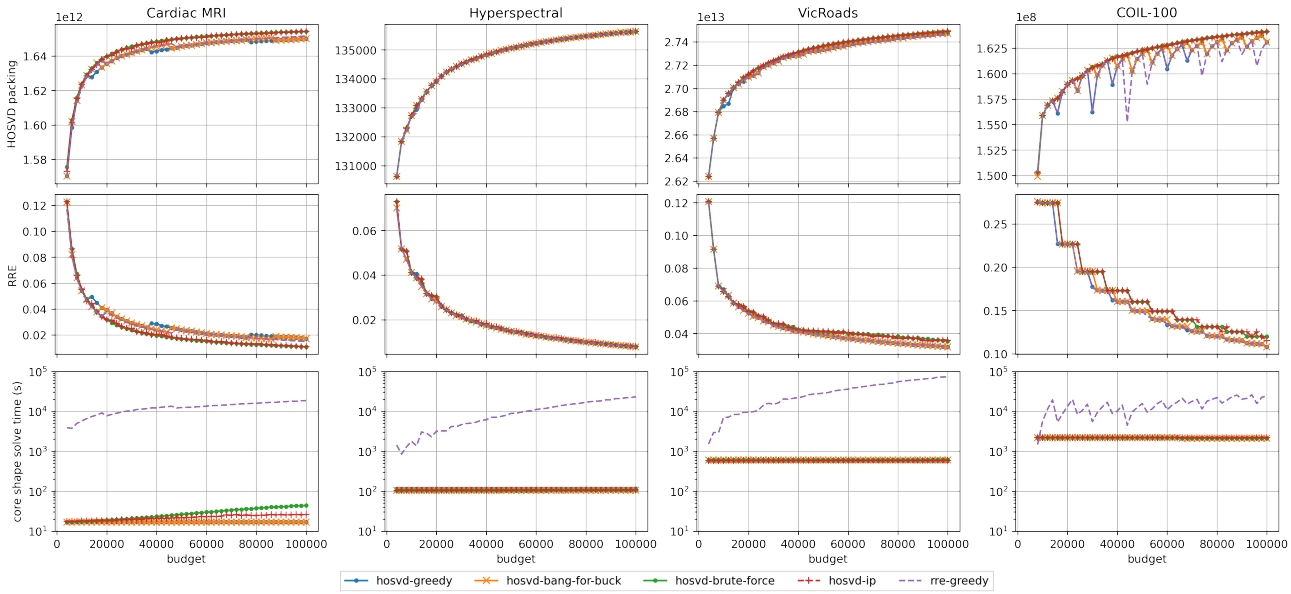

图 2。 在 Tucker 分解大小预算递增至 $c \leq 100,000$ 的情况下，比较四个真实世界张量（列）上的五种核心形状求解器。顶行图显示核心形状解 $\mathbf{r}$ 的 HOSVD Tucker 打包目标值 $f(\mathbf{r})$，中间行显示相对重构误差，底行显示各算法的运行时间（以秒计）。

HOSVD 贪心算法通过反复计算$\mathbf{r} \leftarrow \arg\max_{\mathbf{r}' \in N(\mathbf{r})} f(\mathbf{r}')$来最大化相同的 $N(\mathbf{r})$ 可行核心形状的相邻集合 $\mathbf{r}' = \mathbf{r} + \mathbf{e}_n$ 和$\mathbf{e}_n \in \{0,1\}^N$ 是标准单位向量。这是对预算约束进行扩展后的算法3.1，见Ehrlacher 等人 (2021年)。

HOSVD 性价比算法 与 HOSVD 贪心算法类似，但它在每一步中递增使 $(f(\mathbf{r}') - f(\mathbf{r}))/(\text{cost}(\mathbf{r}') - \text{cost}(\mathbf{r}))$ 最大的维度，其中 $\text{cost}(\mathbf{r})$ 是秩为 $\mathbf{r}$ 的 Tucker分解的大小。

HOSVD 穷举算法 穷举检查所有可行核心形状，并输出最大的 Tucker 打包目标值。

RRE 贪心算法 通过在每一步计算 $O(N)$秩为 $\mathbf{r}'$ 的 Tucker 分解来构造核心形状，并递增使相对重构误差改善最多的维度。具体而言，更新为 $\mathbf{r} \leftarrow \arg\min_{\mathbf{r}' \in N(\mathbf{r})} L(\mathcal{X}, \mathbf{r}')$，类似于 Hashemizadeh 等人(2020年)提出的 Greedy-TL算法。

6.2. 结果

我们考虑所有张量数据集的预算 $c \leq 100,000$。对于每个 c，我们运行每种算法以获得核心形状$\mathbf{r}$。然后在图 2中，我们绘制打包目标 $f(\mathbf{r})$、相对重构误差（RRE），即$L(\mathcal{X}, \mathbf{r})/\|\mathcal{X}\|_F^2$，以及算法运行时间（包括模式-$n$ 奇异值计算）随 c 变化的情况。每次 $L(\mathcal{X}, \mathbf{r})$ 计算均使用 20 次 HOOI 迭代。

心脏 MRI 表明，最大化 HOSVD Tucker 打包目标 $f(\mathbf{r})$ 能够使 RRE 明显优于 RRE 贪心算法，同时运行速度快 1000 倍。仔细观察这些算法输出的核心形状可以发现，HOSVD-{暴力搜索、IP} 总是返回以下形式的核心形状：形式 $(x, y, z, 1)$，而贪心算法会随着 c的增加，将预算分配给时间维度，例如 $(x, y, z, 3)$。第四维度的增加对应着 $f(\mathbf{r})$ 和 RRE 在 图 2中的退化点。

这个张量也足够小，因此可以看出 HOSVD 打包算法运行时间的差异。具体而言，我们看到：**(1)** 计算 $\sigma_{i_n}^{(n)}$ 存在固定开销；**(2)** HOSVD-{贪心算法、物有所值} 比 HOSVD-IP 快，而 HOSVD-IP 又比 HOSVD 穷举算法快。所有算法都明显快于 RRE 贪心算法。

高光谱 表明，代理损失 $L(\mathcal{X}, \mathbf{r})$ 和相对重构误差可以引导贪心算法得到相同的核心形状，并且 HOSVD 贪心算法可以实现最大的 Tucker 打包目标。我们发现，对于 HOSVD 求解器而言，计算高阶奇异值已成为瓶颈，而非求解打包实例本身。

VicRoads 表明，相对重构误差与代理损失之间存在明显差距。尽管 HOSVD-{贪心算法、性价比优先}在打包目标上并非最优，但它们达到的相对重构误差与速度慢 100 倍的 RRE 贪心算法相同。这些数据揭示了代理损失的一个缺陷，同时也表明高阶奇异值仍然有效。

COIL-100 可能是最有趣的张量，因为它展现了 HOSVD Tucker 贪心打包算法的非单调行为。与心脏 MRI 类似，每当贪心核心形状求解器增大第一个索引的维度（对应物体的数量）时，打包目标 $f(\mathbf{r})$。这种现象也出现在 RRE 图中，但方向相反。

致谢

我们感谢早期版本的这项工作中的匿名审稿人，他们提醒我们关注多重选择背包文献。我们还要感谢 Mohit Singh 和 Anupam Gupta，他们就我们的主要算法进行了富有成效的讨论。

参考文献

Arthur, D. 和 Vassilvitskii, S. k-均值++：谨慎选择初始中心的优势。技术报告，斯坦福大学，2006年。

Bader, B. W. 和 Kolda, T. G. MATLAB 张量工具箱，版本 3.4。https://www.tensortoolbox.org/，2022年。

Battaglino, C.、Ballard, G. 和 Kolda, T. G. 一种实用的随机CP张量分解。*SIAM* 矩阵分析与应用杂志，39(2):876–901，2018年。Che, M. 和 Wei, Y. 用于近似 Tucker 分解和张量列车分解的随机化算法。计算数学进展，45(1):395–428，2019年。Cheng, D.、Peng, R.、Liu, Y. 和 Perros, I. Spals：通过隐式杠杆分数采样实现快速交替最小二乘法。神经信息处理系统进展，29，2016年。De Lathauwer, L.、De Moor, B. 和 Vandewalle, J. 多线性奇异值分解。*SIAM* 矩阵分析与应用杂志，21(4):1253–1278，2000a。De Lathauwer, L.、De Moor, B. 和 Vandewalle, J. 关于高阶张量的最佳秩-1和秩-($r_1, r_2, \ldots, r_n$) 近似。*SIAM* 矩阵分析与应用杂志，21(4):1324–1342，2000b年。Eckart, C. 和 Young, G. 用低秩矩阵近似矩阵。*Psychometrika*，1(3):211–218，1936年。Ehrlacher, V.、Grigori, L.、Lombardi, D. 和 Song, H. 用于张量压缩的自适应分层子张量划分。《*SIAM*科学计算杂志》，43(1):A139–A163，2021年。Eld´en, L. 和 Dehghan, M. 一种用于计算大型稀疏张量最佳秩-(r_1, r_2, r_3)近似的 Krylov–Schur类方法。《数值算法》，第1–33页，2022年。Eld´en, L. 和 Savas, B. 一种用于计算张量最佳多线性秩-(r_1, r_2, r_3)近似的 Newton–Grassmann方法。*SIAM* 矩阵分析与应用杂志，31(2):248–271，2009年。

Fahrbach, M., Fu, G., and Ghadiri, M. 具有张量分解应用的次二次 Kronecker 回归。载于神经信息处理系统进展，2022年。Frandsen, A. and Ge, R. Tucker分解的优化景观。数学规划，193(2):687–712，2022年。Garey, M. R. and Johnson, D. S. 计算机与难解性。W. H. Freeman，1979年。Gens, G. V. and Levner, E. V. 组合问题近似算法的计算复杂性。载于国际数学计算机科学基础研讨会，第292–300页。Springer，1979年。Grandoni, F., Ravi, R., Singh, M., and Zenklusen, R. 多目标优化的新方法。数学规划，146(1):525–554，2014年。Grasedyck, L. 张量的层次奇异值分解。*SIAM* 矩阵分析与应用杂志，31(4):2029–2054，2010年。Hackbusch, W. 张量空间与数值张量演算，第56卷。Springer，第2版，2019年。Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., 等。使用 NumPy 进行数组编程。《自然》，585(7825):357–362，2020年。Hashemizadeh, M., Liu, M., Miller, J., and Rabusseau, G. 基于张量网络的自适应张量学习。机器学习中的量子张量网络研讨会，第三十四届神经信息处理系统大会，2020年。Hillar, C. J. and Lim, L.-H. 大多数张量问题都是 NP 难问题。《*ACM* 杂志》（*JACM*），60(6):1–39，2013年。Ishteva, M., De Lathauwer, L., Absil, P.-A., and Van Huffel, S. 求张量最佳秩-(r_1，r_2，r_3) 近似的微分几何牛顿法。《数值算法》，51(2):179–194，2009年。Ishteva, M., Absil, P.-A., Van Huffel, S., and De Lath- auwer, L. 基于黎曼信赖域方案的高阶张量最佳低多线性秩近似。*SIAM* 矩阵分析与应用杂志，32(1):115–135，2011年。Ishteva, M., Absil, P.-A., and Van Dooren, P. 求对称张量最佳低多线性秩近似的 Jacobi 算法。*SIAM* 矩阵分析与应用杂志，34(2):651–672，2013年。

Jang, J.-G. and Kang, U. 用于回答各类时间范围查询的快速且内存高效的 Tucker分解。载于第*27*届*ACM SIGKDD*知识发现与数据挖掘大会论文集，第725–735页，2021年。Kim, Y.-D., Park, E., Yoo, S., Choi, T., Yang, L., and Shin, D. 深度卷积神经网络压缩：面向快速、低功耗移动应用。载于*ICLR*，2016年。Kolda, T. G. and Bader, B. W. 张量分解及其应用。《*SIAM*评论》，51(3):455–500，2009年。Korte, B. and Schrader, R. 关于快速近似方案的存在性。载于《非线性规划*4*》，第415–437页。爱思唯尔，1981年。Kr¨amer, S. 树张量网络、相关奇异值与高维近似。博士论文，亚琛工业大学，2020年。Larsen, B. W. and Kolda, T. G. 用于低秩张量分解的实用杠杆值采样。*SIAM* 矩阵分析与应用杂志，43(3):1488–1517，2022年。Lawler, E. L. 背包问题的快速近似算法。载于第*18*届计算机科学基础年度研讨会，第206–213页。IEEE，1977年。Ma, L. and Solomonik, E. 低秩张量分解的快速、精确随机化算法。神经信息处理系统进展，34:24299–24312，2021年。Malik, O. A. 具有最坏情况保证的张量分解高效采样。载于国际机器学习会议，第14887–14917页。PMLR，2022年。Malik, O. A. and Becker, S. 使用 TensorSketch 对大型张量进行低秩Tucker分解。神经信息处理系统进展，31:10096–10106，2018年。Nascimento, S. M., Amano, K., and Foster, D. H. 自然场景中局部照明颜色的空间分布。《视觉研究》，120:39–44，2016年。Nene, S. A., Nayar, S. K., Murase, H., 等。哥伦比亚物体图像库（COIL-20）。1996年。Oseledets, I. V. 张量列分解。《*SIAM*科学计算杂志》，33(5):2295–2317，2011年。Oseledets, I. V. and Tyrtyshnikov, E. E. 打破维数灾难，或如何在高维情形下使用奇异值分解。《*SIAM*科学计算杂志》，31(5):3744–3759，2009年。Pisinger, D. 多重选择背包问题的一个最简算法。《欧洲运筹学杂志》，83(2):394–410，1995。Schimbinschi, F., Nguyen, X. V., Bailey, J., Leckie, C., Vu, H., and Kotagiri, R. 复杂城市网络中的交通流预测：利用大数据和机器学习。载于*2015*年*IEEE*大数据国际会议，第1019–1024页。IEEE，2015。Sidiropoulos, N. D., De Lathauwer, L., Fu, X., Huang, K., Papalexakis, E. E., and Faloutsos, C. 面向信号处理和机器学习的张量分解。《*IEEE*信号处理汇刊》，65(13):3551–3582，2017。Sinha, P. and Zoltners, A. A. 多重选择背包问题。《运筹学》，27(3):503–515，1979。Song, Z., Woodruff, D. P., and Zhong, P. 相对误差张量低秩近似。载于第三十届*ACM-SIAM*离散算法年度研讨会论文集，第2772–2789页。SIAM，2019。Sun, Y., Guo, Y., Luo, C., Tropp, J., and Udell. 基于流式数据的张量低秩Tucker近似。《*SIAM*数据科学数学杂志》，2(4):1123–1150，2020。Traor´e, A., Berar, M., and Rakotomamonjy, A. Singleshot：一种可扩展的Tucker张量分解。神经信息处理系统进展，32，2019。Vannieuwenhoven, N., Vandebril, R., and Meerbergen, K. 高阶奇异值分解的一种新截断策略。《*SIAM*科学计算杂志》，34(2):A1027–A1052，2012。Xiao, C. and Yang, C. 一种用于截断Tucker分解的秩自适应高阶正交迭代算法。*arXiv*预印本*arXiv:2110.12564*，2021。Zhou, G., Cichocki, A., and Xie, S. 低多线性秩大规模张量的分解。*arXiv*预印本*arXiv:1412.1885*，2014。

A. 第3节的缺失分析

A.1. 定理 3.2的证明

A.1.1. 重构误差上界

对于任意 Tucker分解 $\mathcal{X} = \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \cdots \times_N \mathbf{A}^{(N)} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$， $\mathcal{X}$ 的模-n 展开可写为 as

$$\mathbf{X}_{(n)} = \mathbf{A}^{(n)} \mathbf{G}_{(n)} \left(\mathbf{A}^{(1)} \otimes \cdots \mathbf{A}^{(n-1)} \otimes \mathbf{A}^{(n+1)} \otimes \cdots \otimes \mathbf{A}^{(N)} \right)^{\intercal}. \tag{18}$$

这是对Kolda和Bader （2009年）中公式（4.2）的修正版。

引理 A.1 （De Lathauwer 等人 （2000a，性质10））。 对于任意 $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ 和 $n \in [N]$，令 $\sigma_1^{(n)} \geq \sigma_2^{(n)} \cdots \geq \sigma_{I_n}^{(n)}$ 表示 $\mathbf{X}_{(n)}$的奇异值。则对于任意核心形状$\mathbf{r} = (R_1, R_2, \ldots, R_N) \in [I_1] \times [I_2] \times \cdots \times [I_N]$，有

$$\begin{aligned} L(\mathcal{X}, \mathbf{r}) &\leq \|\mathcal{X} - \widehat{\mathcal{X}}_{\text{HOSVD}(\mathbf{r})}\|_{\text{F}}^2 \\ &\leq \sum_{i_1=R_1+1}^{I_1} \left(\sigma_{i_1}^{(1)}\right)^2 + \sum_{i_2=R_2+1}^{I_2} \left(\sigma_{i_2}^{(2)}\right)^2 + \cdots + \sum_{i_N=R_N+1}^{I_N} \left(\sigma_{i_N}^{(N)}\right)^2. \end{aligned}$$

证明。 设 $\mathcal{X} = \mathcal{S} \times_1 \mathbf{U}^{(1)} \times_2 \cdots \times_N \mathbf{U}^{(N)}$ 的高阶奇异值分解定理3.1中所示。设 $\mathcal{S}$ 表示 $\mathcal{S}$相对于核心形状$\mathbf{r}$ 的截断版本，满足

$$\overline{s}_{i_1 \ldots i_N} = \begin{cases} s_{i_1 \ldots i_N} & \text{if } i_n \leq R_n \text{ for all } n \in [N], \\ 0 & \text{otherwise.} \end{cases}$$

设 $\overline{\mathcal{X}} = \overline{\mathcal{S}} \times_1 \mathbf{U}^{(1)} \times_2 \cdots \times_N \mathbf{U}^{(N)}$。对所有维度 $n \in [N]$ 求和，并利用高阶奇异值分解可得定理3.1定理3.1，

$$\begin{aligned} \|\mathcal{S} - \overline{\mathcal{S}}\|_{\text{F}}^2 &\leq \sum_{n=1}^{N} \sum_{j=R_n+1}^{I_n} \|(\mathcal{S} - \overline{\mathcal{S}})_{i_n=j}\|_{\text{F}}^2 \\ &= \sum_{n=1}^{N} \sum_{j=R_n+1}^{I_n} \|\mathcal{S}_{i_n=j}\|_{\text{F}}^2 \\ &= \sum_{n=1}^{N} \sum_{i_n=R_n+1}^{I_n} \left(\sigma_{i_n}^{(n)}\right)^2. \end{aligned}$$

我们有 $\|\mathcal{S} - \overline{\mathcal{S}}\|_{\text{F}}^2 = \|\mathbf{S}^{(n)} - \mathbf{S}_{(n)}^{-}\|_{\text{F}}^2$对 n的所有取值均成立。此外，由于两个正交矩阵的克罗内克积也是正交矩阵，且乘以正交矩阵不会影响弗罗贝尼乌斯范数，公式（18）表明

$$\begin{aligned} \|\mathbf{S}_{(n)} - \overline{\mathbf{S}}_{(n)}\|_{\text{F}}^2 &= \|\mathbf{U}_{(n)}(\mathbf{S}_{(n)} - \overline{\mathbf{S}}_{(n)})(\mathbf{U}_{(1)} \otimes \ldots \otimes \mathbf{U}_{(n-1)} \otimes \mathbf{U}_{(n+1)} \otimes \cdots \otimes \mathbf{U}_{(N)})^{\intercal}\|_{\text{F}}^2 \\ &= \|\mathcal{X} - \overline{\mathcal{X}}\|_{\text{F}}^2. \end{aligned}$$

注意到 $\overline{\mathcal{X}} = \mathcal{X}_{\text{HOSVD}(\mathbf{r})}$ 根据 `TuckerHOSVD` 在 算法1 的定义，并综合以上结果，

$$L(\mathcal{X}, \mathbf{r}) \leq \|\mathcal{X} - \widehat{\mathcal{X}}_{\text{HOSVD}(\mathbf{r})}\|_{\text{F}}^2 = \|\mathcal{X} - \overline{\mathcal{X}}\|_{\text{F}}^2 = \|\mathcal{S} - \overline{\mathcal{S}}\|_{\text{F}}^2 \leq \sum_{n=1}^{N} \sum_{i_n=R_n+1}^{I_n} \left(\sigma_{i_n}^{(n)}\right)^2,$$

至此，证明完成。 □

A.1.2. 重构误差的下界

定理 A.2 (Eckart 与 Young (1936年))。 设 $\mathbf{A} \in \mathbb{R}^{n \times d}$满足 $n \geq d$，且奇异值为 $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_d \geq 0$。设 $\mathbf{A}_k$是 k 阶的最佳近似$\mathbf{A}$ （按弗罗贝尼乌斯范数衡量）。则

$$\|\mathbf{A} - \mathbf{A}_k\|_{\text{F}}^2 = \sum_{i=k+1}^{d} \sigma_i^2.$$

引理 A.3。 对于任意 $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ 和 $n \in [N]$，令 $\sigma_1^{(n)} \geq \cdots \geq \sigma_{I_n}^{(n)}$ 表示$\mathbf{X}_{(n)}$的奇异值。对于每种核心形状 $\mathbf{r} = (R_1, R_2, \ldots, R_N) \in [I_1] \times [I_2] \times \cdots \times [I_N]$，有

$$L(\mathcal{X}, \mathbf{r}) \geq \max_{n \in [N]} \sum_{i_n = R_n + 1}^{I_n} \left(\sigma_{i_n}^{(n)}\right)^2.$$

证明。 设使 $L(\mathcal{X}, \mathbf{r})$ 最小的核心张量和因子分别为 $\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$ 和 $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$，即

$$L(\mathcal{X}, \mathbf{r}) = \|\mathcal{X} - \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \cdots \times_N \mathbf{A}^{(N)}\|_{\mathrm{F}}^2.$$

Let $\widehat{\mathcal{X}} = \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \cdots \times_N \mathbf{A}^{(N)}$。由方程（18）以及 $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ 的维度可知 at

$$\operatorname{rank}\left(\widehat{\mathbf{X}}_{(n)}\right) \leq \operatorname{rank}\left(\mathbf{A}^{(n)}\right) \leq R_n,$$

因为 $R_n \leq I_n$。根据 Eckart–Young–Mirsky 定理（定理 A.2）以及 $\sigma_i^{(n)}$ 在 定理3.1 中的刻画，可得

$$\|\mathcal{X} - \widehat{\mathcal{X}}\|_{\mathrm{F}}^2 = \|\mathbf{X}_{(n)} - \widehat{\mathbf{X}}_{(n)}\|_{\mathrm{F}}^2 \geq \sum_{i_n = R_n + 1}^{I_n} \left(\sigma_{i_n}^{(n)}\right)^2. \tag{19}$$

E方程（19）对 n的所有取值都成立，因此取使右侧最大化的方程。 □

A.1.3. 结果结合

现在结合引理 A.1和引理 A.3，得到一个对所有核心形状都成立的近似不等式。

引理 A.4。 对于任意 $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ 以及任意核心形状$\mathbf{r} = (R_1, R_2, \ldots, R_N) \in [I_1] \times [I_2] \times \cdots \times [I_N]$，有

$$\frac{1}{N} \cdot \widetilde{L}(\mathcal{X}, \mathbf{r}) \leq L(\mathcal{X}, \mathbf{r}) \leq \widetilde{L}(\mathcal{X}, \mathbf{r}),$$

~其中 $L(\mathcal{X}, \mathbf{r})$ 定义见公式（2）。

证明。 将引理 A.3的证明中的公式（19）对 $n \in [N]$ 的所有取值求和，得到

$$\sum_{n=1}^{N} \sum_{i_n = R_n + 1}^{I_n} \left(\sigma_{i_n}^{(n)}\right)^2 \leq N \cdot L(\mathcal{X}, \mathbf{r}) \implies \frac{1}{N} \cdot \widetilde{L}(\mathcal{X}, \mathbf{r}) \leq L(\mathcal{X}, \mathbf{r}). \tag{20}$$

~上界 $L(\mathcal{X}, \mathbf{r}) \leq L(\mathcal{X}, \mathbf{r})$ 是引理 A.1的另一种表述。 □

定理 3.2 （De Lathauwer 等人（2000a，性质10）；Hackbusch （2019年，定理10.2））。对于任意张量 $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ 和核心形状$\mathbf{r} \in [I_1] \times \cdots \times [I_N]$，令$\texttt{TuckerHOSVD}(\mathcal{X}, \mathbf{r})$ 的输出为 $\mathcal{G} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$ 和$\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}$，对于每个 $n \in [N]$。若令

$$\widehat{\mathcal{X}}_{\mathrm{HOSVD}(\mathbf{r})} \stackrel{\mathrm{def}}{=} \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \cdots \times_N \mathbf{A}^{(N)} \tag{1}$$

表示重构的$\mathbf{r}$截断张量，则

$$\begin{aligned} \|\mathcal{X} - \widehat{\mathcal{X}}_{\mathrm{HOSVD}(\mathbf{r})}\|_{\mathrm{F}}^2 &\leq \sum_{n=1}^{N} \sum_{i_n = R_n + 1}^{I_n} \left(\sigma_{i_n}^{(n)}\right)^2 \\ &\leq N \cdot L(\mathcal{X}, \mathbf{r}). \end{aligned}$$

此外，我们有 $L(\mathcal{X}, \mathbf{r}) \leq \|\mathcal{X} - \widehat{\mathcal{X}}_{\mathrm{HOSVD}(\mathbf{r})}\|_{\mathrm{F}}^2$.

证明。 证明结合引理 A.1 和公式（20）。 □

A.2. 引理 3.4 的证明

引理 A.5。 对于任意张量 $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ 以及 *Tucker*分解大小的预算 $c \geq 1 + \sum_{n=1}^{N} I_n$，令可行核心形状集合为

$$F = \left\{ \mathbf{r} \in [I_1] \times \cdots \times [I_N] : \prod_{n=1}^{N} R_n + \sum_{n=1}^{N} I_n R_n \leq c \right\}.$$

于是，我们有

$$\widetilde{\mathbf{r}}^* \overset{\text{def}}{=} \arg\min_{\mathbf{r} \in F} \widetilde{L}(\mathcal{X}, \mathbf{r}) = \arg\max_{\mathbf{r} \in F} \sum_{n=1}^{N} \sum_{i_n=1}^{R_n} \left(\sigma_{i_n}^{(n)} \right)^2.$$

此外，如果 $\mathbf{r}^* \overset{\text{def}}{=} \arg\min_{\mathbf{r} \in F} L(\mathcal{X}, \mathbf{r})$ 是一个最优预算约束核心形状，那么 $L(\mathcal{X}, \widetilde{\mathbf{r}}^*) \leq N \cdot L(\mathcal{X}, \mathbf{r}^*)$。

证明。 对于任意 n，我们有

$$\sum_{i_n=1}^{I_n} \left(\sigma_{i_n}^{(n)} \right)^2 = \|\mathbf{X}_{(n)}\|_{\mathrm{F}}^2 = \|\mathcal{X}\|_{\mathrm{F}}^2.$$

因此，对于 $\mathbf{r} = (R_1, R_2, \ldots, R_N)$ 的任意取值，我们有

$$\left[\sum_{n=1}^{N} \sum_{i_n=1}^{R_n} \left(\sigma_{i_n}^{(n)} \right)^2 \right] + \left[\sum_{n=1}^{N} \sum_{i_n=R_n+1}^{I_n} \left(\sigma_{i_n}^{(n)} \right)^2 \right] = N \|\mathcal{X}\|_{\mathrm{F}}^2.$$

这是一个只取决于 $\mathcal{X}$ 的常数值，因此最小化 $L(\mathcal{X}, \mathbf{r})$ 等价于最大化装箱版本，因为两个问题都在同一个集合 F 上进行优化。

最后，我们有

$$\widetilde{L}(\mathcal{X}, \widetilde{\mathbf{r}}^*) \leq \widetilde{L}(\mathcal{X}, \mathbf{r}^*) \leq N \cdot L(\mathcal{X}, \mathbf{r}^*),$$

其中，第一个不等式来自对代理损失进行优化，第二个不等式则来自 引理 A.4，因为该结论对所有核心形状都成立。 □

A.3. 困难性

定义 A.6。 设 $N \geq 2$ 为偶整数，且 w_1、……、$w_N \geq 1$ 为整数。等分问题要求判断是否存在一个大小为 $n/2$ 的子集 $S \subseteq [N]$，使得

$$\sum_{i \in S} w_i = \sum_{i \in [N] \setminus S} w_i.$$

引理 A.7 (Garey 和 Johnson, 1979, SP12)。 等分问题 是NP 完全。

现在我们给出从等分问题到 Tucker 装箱问题的归约。

定理 3.6。 *Tucker* 装箱问题是NP难的。

证明。 设 $T, w_1, \ldots, w_T$是等分划分的一个实例，其中对所有 $n \in [T]$，都有 $w_n \geq 2$。注意，假设$w_n \geq 2$不失一般性，因为我们可以将所有值 $w_1, \ldots, w_T$乘以二。

设 $M = \sum_{n \in [T]} w_n$为所有权重之和，并令 $N \geq T$为满足 $2^{N-T/2} > 4(N-T)+3M/2$的最小整数。现在我们构造一个 Tucker 装箱问题的实例。对于每个 $n \in [T]$，令：

- $I_n = w_n$

- $a_1^{(n)} = 2M$
- $a_2^{(n)} = M + w_n$
- $a_{i_n}^{(n)} = 0$, for all $i_n \in [I_n] \setminus \{1, 2\}$

接下来，对于每个 $n \in [N] \setminus [T]$，令:

- $I_n = 2$
- $a_1^{(n)} = a_2^{(n)} = 2M$

最后，将预算设为 $c = 2^{N-T/2} + 4(N - T) + 3M/2$。

首先，注意到这是一个有效的 Tucker 装箱问题实例，因为对于所有 $n \in [N]$，$a_1^{(n)} \geq a_2^{(n)} \geq \cdots \geq a_{I_n}^{(n)}$。此外，由于 $N = O(T \cdot \log(3M/2))$ 且对于 $i_n \in [I_n] \setminus \{1, 2\}$，$a_{i_n}^{(n)} = 0$，因此该问题描述的规模相对于对应等分问题描述的规模是多项式的。

现在，我们考虑这个 Tucker 装箱问题的一个判定版本，其中要求判断是否存在一个可行解 $(R_1, \ldots, R_N)$，使得

$$\sum_{n=1}^{N} \sum_{i_n=1}^{R_n} a_{i_n}^{(n)} \geq M(4N - 3T/2) + M/2. \tag{21}$$

我们证明，对于（21）中的 Tucker 装箱问题判定版本，若答案为是，则等分问题的答案也为是，反之亦然。

假设 Tucker 装箱问题的判定版本的答案为 YES，并且$\mathbf{r}^*$ 是一个最优解，满足

$$\sum_{n=1}^{N} \sum_{i_n=1}^{R_n^*} a_{i_n}^{(n)} \geq M(4N - 3T/2) + M/2 \quad \text{and} \quad \prod_{n=1}^{N} R_n^* + \sum_{n=1}^{N} I_n R_n^* \leq c.$$

由于

$$c = 2^{N-T/2} + 4(N - T) + 3M/2 < 2 \cdot 2^{N-T/2} = 2^{N-T/2+1},$$

至多有 $N - T/2$ 个 R_n^* 的取值满足 $R_n^* \geq 2$。此外，由于对所有 $i_n \geq 3$都有 $a_{i_n}^{(n)} = 0$，因此在最小最优解中绝不会出现 $R_n^* > 2$。由此可知，对所有 $n \in [N]$都有 $R_n^* \in \{1, 2\}$，并且

$$\prod_{n=1}^{N} R_n^* \leq 2^{N-T/2}.$$

接下来，我们建立此 Tucker 打包实例的最优解结构。注意，$\hat{\mathbf{r}} = (R_1, \ldots, R_N)$，其中 $R_1 = \cdots = R_T = 1$，且 $R_{T+1} = \cdots = R_N = 2$是一个$\mathbf{r}$可行解，其目标值为 $M(4N - 2T)$。现在考虑任意一个可行解，其中存在 $i \in [T]$和 $j \in [N] \setminus [T]$，使得 $R_i = 2$且 $R_j = 1$。如果交换 R_i和 R_j的值，那么代价会降低 $w_i - 2 \geq 0$ （也就是说，该解仍然可行），而目标值会增加 $M - w_i > 0$。因此，由于 $\hat{\mathbf{r}}$是可行的，在最优解中，对所有 $n \in [N] \setminus [T]$都有 $R_n = 2$，且对于 $n \in [T]$，至多有 R_n中的 $T/2$个等于二。

令 $S = \{i \in [T] : R_i^* = 2\}$。根据构造，我们有

$$\sum_{n \in S} a_2^{(n)} = \sum_{n \in S} (M + w_n) = M|S| + \sum_{n \in S} w_n < M(|S| + 1).$$

此外，由于判定问题的答案为“是”，且在一个最优解中，对于所有 $n \in [N] \setminus [T]$都有 $R_n^* = 2$，因此可得

$$\begin{aligned}\sum_{n\in S} a_2^{(n)} &= f(\mathbf{r}^*) - \sum_{n=1}^{N} a_1^{(n)} - \sum_{n=T+1}^{N} a_2^{(n)} \\ &= f(\mathbf{r}^*) - 2NM - 2(N-T)M \\ &\geq M(4N - 3T/2) + M/2 - 2NM - 2(N-T)M \\ &= MT/2 + M/2.\end{aligned} \tag{22}$$

因此，

$$M(|S|+1) > MT/2 + M/2,$$

这意味着 $|S| > T/2 - 1/2$，因此 $|S| \geq T/2$，因为 $|S|$ 和 $T/2$ 都是整数。利用上述关于最优解的刻画，以及预算严格小于 $2^{N-T/2+1}$ 这一事实，可得 $|S| \leq T/2$。因此，判定问题的答案为 YES 意味着 $|S| = T/2$，进而意味着 $\prod_{n=1}^{N} R_n^* = 2^{N-T/2}$。
由我们对预算 c 的选择可知

$$\sum_{n=1}^{N} I_n R_n^* \leq c - 2^{N-T/2} = 4(N-T) + 3M/2,$$

根据 I_n 的定义，这意味着

$$\left(\sum_{n=1}^{T} w_n R_n^* + \sum_{n=T+1}^{N} 2R_n^*\right) = \left(M + \sum_{n\in S} w_n\right) + 4(N-T) \leq 4(N-T) + 3M/2 \implies \sum_{n\in S} w_n \leq M/2.$$

此外，利用（22）、$a_{i_n}^{(n)}$ 的定义以及 $|S| = T/2$ 这一事实，可得

$$\sum_{n\in S} a_2^{(n)} = \sum_{n\in S}(M + w_n) = |S|M + \sum_{n\in S} w_n \geq MT/2 + M/2 \implies \sum_{n\in S} w_n \geq M/2.$$

把所有内容放在一起，我们得到 $\sum_{n\in S} w_n = M/2$。因此，S 是等分问题的一个解。
现在假设等分问题的答案是 YES。令 $S \subseteq [T]$ 使得 $|S| = T/2$ 和 $\sum_{n\in S} w_n = M/2$。按如下方式构造 $\mathbf{r}^*$：对于每个 $n \in S \cup ([N] \setminus [T])$，设 $R_n^* = 2$；对于每个 $n \in [T] \setminus S$，设 $R_n^* = 1$。
那么，根据上述 I_n 和 $a_{i_n}^{(n)}$ 的定义，我们有

$$\sum_{n=1}^{N} \sum_{i_n=1}^{R_n^*} a_{i_n}^{(n)} \geq M(4N - 3T/2) + M/2 \quad \text{and} \quad \prod_{n=1}^{N} R_n^* + \sum_{n=1}^{N} I_n R_n^* \leq c,$$

由此完成证明。 □

A.4. 近似最大化与最小化之间的转换

我们证明了一个加性误差保证，说明 Tucker 装箱问题的一个 $(1-\varepsilon')$-近似解，即核心形状 $\mathbf{r} \in [I_1] \times \cdots \times [I_N]$，如何导致代理损失目标增加。

引理 A.8. 设 $\mathbf{r} \in [I_1] \times \cdots \times [I_N]$ 是任意一个核心形状，它能为 *Tucker* 装箱问题实现 $(1-\varepsilon/N)$-近似。则有 相对重构误差$(\mathcal{X}, \mathbf{r}) \leq N \cdot$ 相对重构误差$(\mathcal{X}, \mathbf{r}^*) + \varepsilon$，其中 相对重构误差$(\mathcal{X}, \mathbf{r}) := L(\mathcal{X}, \mathbf{r})/\|\mathcal{X}\|_{\mathrm{F}}^2$。

证明。令 $\varepsilon' = \varepsilon/N$ 和 $\tilde{\mathbf{r}}^*$ 为代理损失 L 的最优形状。若 $\mathbf{r} = (R_1, \ldots, R_N)$ 是 Tucker 装箱问题的 $(1-\varepsilon')$-近似解，

则可得

$$\begin{aligned}\frac{\widetilde{L}(\mathcal{X},\mathbf{r})}{\|\mathcal{X}\|_{\mathrm{F}}^2} &= \frac{N\|\mathcal{X}\|_{\mathrm{F}}^2 - \sum_{n=1}^{N}\sum_{i_n=1}^{R_n}\left(\sigma_{i_n}^{(n)}\right)^2}{\|\mathcal{X}\|_{\mathrm{F}}^2} \\ &\le \frac{N\|\mathcal{X}\|_{\mathrm{F}}^2 - (1-\varepsilon')\sum_{n=1}^{N}\sum_{i_n=1}^{\widetilde{R}_n^*}\left(\sigma_{i_n}^{(n)}\right)^2}{\|\mathcal{X}\|_{\mathrm{F}}^2} \\ &= \frac{\widetilde{L}(\mathcal{X},\widetilde{\mathbf{r}}^*)}{\|\mathcal{X}\|_{\mathrm{F}}^2} + \varepsilon'\left(N - \frac{\widetilde{L}(\mathcal{X},\widetilde{\mathbf{r}}^*)}{\|\mathcal{X}\|_{\mathrm{F}}^2}\right) \\ &\le \frac{\widetilde{L}(\mathcal{X},\widetilde{\mathbf{r}}^*)}{\|\mathcal{X}\|_{\mathrm{F}}^2} + \varepsilon.\end{aligned}$$

~ ~ ~定理 3.2 给出了 $L(\mathcal{X},\mathbf{r}) \le L(\mathcal{X},\mathbf{r}) \le N \cdot L(\mathcal{X},\mathbf{r})$。根据定义 $L(\mathcal{X},\tilde{}\mathbf{r}^*) \le L(\mathcal{X},\mathbf{r}^*)$，因此我们有

$$\mathrm{RRE}(\mathcal{X},\mathbf{r}) = \frac{L(\mathcal{X},\mathbf{r})}{\|\mathcal{X}\|_{\mathrm{F}}^2} \le \frac{\widetilde{L}(\mathcal{X},\mathbf{r})}{\|\mathcal{X}\|_{\mathrm{F}}^2} \le \frac{\widetilde{L}(\mathcal{X},\widetilde{\mathbf{r}}^*)}{\|\mathcal{X}\|_{\mathrm{F}}^2} + \varepsilon \le \frac{\widetilde{L}(\mathcal{X},\mathbf{r}^*)}{\|\mathcal{X}\|_{\mathrm{F}}^2} + \varepsilon \le N \cdot \frac{L(\mathcal{X},\mathbf{r}^*)}{\|\mathcal{X}\|_{\mathrm{F}}^2} + \varepsilon = N \cdot \mathrm{RRE}(\mathcal{X},\mathbf{r}^*) + \varepsilon,$$

正如所需。 □

B. 第 4 节中缺失的分析

B.1. 引理的证明 引理 4.3

引理 B.1. 设 $0 < \varepsilon \le 1$ 和 $F \subseteq [I_1] \times \cdots \times [I_N]$ 向下封闭。对于每个 $n \in [N]$，定义

$$S_n^{(\varepsilon)} = \left\{\lceil (1+\varepsilon)^k \rceil : k \in \mathbb{Z}_{\ge 0}, \lceil (1+\varepsilon)^k \rceil \le I_n\right\}.$$

令 $\mathbf{r}^*$ 为广义问题的一个最优解

$$\begin{aligned}&\text{maximize} \quad \sum_{n=1}^{N}\sum_{i_n=1}^{R_n} a_{i_n}^{(n)} \\ &\text{subject to} \quad (R_1,\ldots,R_N) \in F\end{aligned} \tag{9}$$

并令 $\mathbf{r}^{(\varepsilon)} = (R_1^{(\varepsilon)},\ldots,R_N^{(\varepsilon)})$ 为以下问题的一个最优解

$$\begin{aligned}&\text{maximize} \quad \sum_{n=1}^{N}\sum_{i_n=1}^{R_n} a_{i_n}^{(n)} \\ &\text{subject to} \quad (R_1,\ldots,R_N) \in (S_1^{(\varepsilon)} \times \cdots \times S_N^{(\varepsilon)}) \cap F\end{aligned} \tag{10}$$

那么， $f(\mathbf{r}^{(\varepsilon)}) \ge (1+\varepsilon)^{-1} f(\mathbf{r}^*)$。此外，存在一种算法，可以在运行时间$O(\sum_{n=1}^{N} I_n + \varepsilon^{-N}\prod_{n=1}^{N}(1+\log_2(I_n)))$内找到式(10)的一个最优解。

证明。令 $k_n \ge 0$ 为满足对每个 $n \in [N]$ 均有 $(1+\varepsilon)^{k_n} \le R_n^*$ 的最大整数。此外，令

$$\widehat{R}_n = \lceil (1+\varepsilon)^{k_n} \rceil.$$

^ ^ ^由于 R_n^* 是一个整数，我们知道 $R_n \le R_n^*$。因此，由于 F 是向下封闭的， ^$\mathbf{r} = (R_1,\ldots,R_N)$ 是（10）的一个可行解。由此可得

$$f(\widehat{\mathbf{r}}) \le f(\mathbf{r}^{(\varepsilon)}). \tag{23}$$

Now 我们将证明 $f(\mathbf{r}^*) \leq (1+\varepsilon) f(\widehat{\mathbf{r}})$。由于 $a_1^{(n)} \geq \cdots \geq a_{I_n}^{(n)} \geq 0$ 对所有 $n \in [N]$ 都成立，我们有 e

$$(1+\varepsilon) \sum_{i_n=1}^{\widehat{R}_n} a_{i_n}^{(n)} \geq \sum_{i_n=1}^{\lfloor (1+\varepsilon)\widehat{R}_n \rfloor} a_{i_n}^{(n)}. \tag{24}$$

根据 k_n 的定义，可得

$$\begin{aligned}(1+\varepsilon)\widehat{R}_n &= (1+\varepsilon)\left\lceil (1+\varepsilon)^{k_n} \right\rceil \\ &\geq (1+\varepsilon)^{k_n+1} \\ &> R_n^*.\end{aligned}$$

由于 R_n^* 是整数，因此有 $\lfloor (1+\varepsilon) R_n \rfloor \geq R_n^*$。因此，利用公式（25）可得

$$(1+\varepsilon) \sum_{i_n=1}^{\widehat{R}_n} a_{i_n}^{(n)} \geq \sum_{i_n=1}^{R_n^*} a_{i_n}^{(n)}. \tag{25}$$

最后，对 $n \in [N]$ 求和并利用公式（23），可得

$$(1+\varepsilon) f(\widehat{\mathbf{r}}) \geq f(\mathbf{r}^*) \implies f(\mathbf{r}^{(\varepsilon)}) \geq (1+\varepsilon)^{-1} f(\mathbf{r}^*).$$

算法。 现在我们设 描述并分析一种用于求解网格搜索问题的简单算法（见（10）。首先注意到

$$|S_n^{(\varepsilon)}| = 1 + \left\lfloor \log_{1+\varepsilon}(I_n) \right\rfloor = 1 + \left\lfloor \frac{\log_2(I_n)}{\log_2(1+\varepsilon)} \right\rfloor.$$

由此可知，问题（10）的可行解数量为

$$O\left(\prod_{n=1}^{N} \left(1 + \frac{\log_2(I_n)}{\log_2(1+\varepsilon)} \right) \right) = O\left(\frac{1}{\log_2^N(1+\varepsilon)} \prod_{n=1}^{N} (1 + \log_2(I_n)) \right),$$

因为对于 $\varepsilon \leq 1$，$\log_2(1+\varepsilon) \leq 1$。此外，注意到对于 $0 < \varepsilon \leq 1$，$\log_2(1+\varepsilon) \geq \varepsilon$，可得上界为

$$O\left(\frac{1}{\varepsilon^N} \prod_{n=1}^{n} (1 + \log_2(I_n)) \right)$$

可行解的数量。

计算出 $S_n^{(\varepsilon)}$ 的元素的前缀和后，我们可以按字典序遍历 $S_1^{(\varepsilon)} \times \cdots \times S_N^{(\varepsilon)}$ 的元素，检查可行性，并在摊销时间 $O(1)$ 内计算任意候选解的目标值 $f(\mathbf{r})$。最后注意， $S_n^{(\varepsilon)}$ 的元素的前缀和可在 $O(I_n)$ 时间内计算。

□

算法 2 中采用两种不同预算划分方式的理由。算法 2。 我们不必考虑两种不同的预算划分方法（即，分别采用 c_{factor} 和 c_{core}），而可以只考虑以下形式问题上的预算划分 $c_{\text{factor}} \in [\sum_{n=1}^{N} I_n^2]$：

$$\begin{aligned} \text{maximize} \quad & f(\mathbf{r}) \\ \text{subject to} \quad & \textstyle\prod_{n=1}^{N} R_n \leq c - c_{\text{factor}} \\ & \textstyle\sum_{n=1}^{N} I_n R_n \leq c_{\text{factor}} \\ & R_n \in [I_n] \qquad \forall n \in [N] \end{aligned}$$

这种方法也能给出多项式时间近似方案（PTAS）。然而，运行时间为 $O((\sum_{n=1}^{N} I_n^2)(\sum_{n=1}^{N} I_n)^{O(1/\varepsilon)})$。相比之下，算法 2 的运行时间为 $O((\log_{1+\varepsilon}(c) + \lceil 1/\varepsilon \rceil \sum_{n=1}^{N} I_n)(\sum_{n=1}^{N} I_n)^{O(1/\varepsilon)})$。注意， $c \leq \sum_{n=1}^{N} I_n^2 + \prod_{n=1}^{N} I_n$。因此，$\log_{1+\varepsilon}(c) = O(\frac{1}{\varepsilon} \sum_{n=1}^{N} \log_2(I_n)) \ll \lceil 1/\varepsilon \rceil \sum_{n=1}^{N} I_n$，其中第一个等式由对数函数的凹性得出。因此，我们的运行时间为 $O((\lceil 1/\varepsilon \rceil \sum_{n=1}^{N} I_n)(\sum_{n=1}^{N} I_n)^{O(1/\varepsilon)})$，这始终小于 $O((\sum_{n=1}^{N} I_n^2)(\sum_{n=1}^{N} I_n)^{O(1/\varepsilon)})$。

C. 树张量网络

备注 C.1. 尽管树张量网络针对一般的希尔伯特空间进行考虑，并使用张量网络符号约定以完全一般的形式定义（例如，参见 Hackbusch (2019 年，第 11 章）和 Kr¨amer (2020 年，第 3 章）），为使符号约定更简洁，我们考虑有限维欧几里得空间。

注 C.2. 定理 5.6 可由引理 4.3推出。

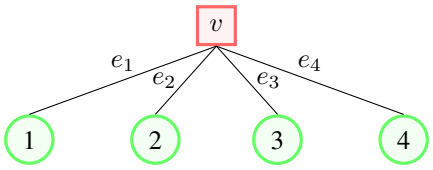

图 *3.* 与 Tucker 分解对应的树张量网络。

示例（Tucker 分解）。Tucker分解对应于深度为一的树张量。例如，图 3 中的树张量由矩阵 $\mathbf{A}_1 \in \mathbb{R}^{I_1 \times R_{e_1}}$、$\mathbf{A}_2 \in \mathbb{R}^{I_2 \times R_{e_2}}$、$\mathbf{A}_3 \in \mathbb{R}^{I_3 \times R_{e_3}}$、$\mathbf{A}_4 \in \mathbb{R}^{I_4 \times R_{e_4}}$ 以及张量$\mathcal{T}_v \in \mathbb{R}^{R_{e_1} \times R_{e_2} \times R_{e_3} \times R_{e_4}}$构成。对应的重构为

$$\widehat{\mathcal{X}} = \mathcal{T}_v \times_{e_4} \mathbf{A}_4 \times_{e_3} \mathbf{A}_3 \times_{e_2} \mathbf{A}_2 \times_{e_1} \mathbf{A}_1.$$

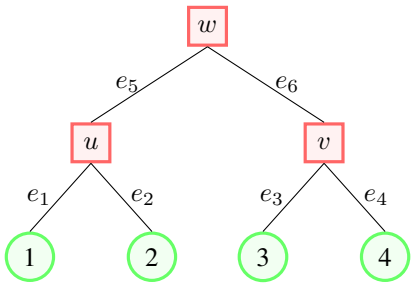

图 *4.* 与层次 Tucker 分解对应的树张量网络示例。

示例（层次 Tucker 分解）。为了更好地理解定义 5.1，我们给出一个四阶4张量的示例。考虑图 4中所示的树。该树张量网络对应于矩阵$\mathbf{A}_1 \in \mathbb{R}^{I_1 \times R_{e_1}}$、$\mathbf{A}_2 \in \mathbb{R}^{I_2 \times R_{e_2}}$、$\mathbf{A}_3 \in \mathbb{R}^{I_3 \times R_{e_3}}$、$\mathbf{A}_4 \in \mathbb{R}^{I_4 \times R_{e_4}}$ 以及张量 $\mathcal{T}_u \in \mathbb{R}^{R_{e_1} \times R_{e_2} \times R_{e_5}}$、$\mathcal{T}_v \in \mathbb{R}^{R_{e_3} \times R_{e_4} \times R_{e_6}}$、$\mathcal{T}_w \in \mathbb{R}^{R_{e_5} \times R_{e_6}}$。相应的重构为

$$\widehat{\mathcal{X}} = \mathcal{T}_w \times_{e_6} \mathcal{T}_v \times_{e_5} \mathcal{T}_u \times_{e_4} \mathbf{A}_4 \times_{e_3} \mathbf{A}_3 \times_{e_2} \mathbf{A}_2 \times_{e_1} \mathbf{A}_1.$$

D. 实验

我们简要介绍第 6 节中用于核形状实验的各个张量数据集，以帮助理解算法为何会构建出相应的核心形状。

心脏 MRI。$256 \times 256 \times 14 \times 20$ 张量，其元素为 MRI 测量值，按 (x、y、z、t) 索引，其中 (x、y、z) 是空间中的一个点，而 t 对应时间。

高光谱。$1024 \times 1344 \times 33$ 展示自然场景光照变化的延时高光谱辐射图像的张量（Nascimento 等人，2016年）。

VicRoads。$1084 \times 2033 \times 96$ 包含墨尔本及其周边郊区 2033 天交通流量数据的张量。这些数据来自由 1084 个道路传感器组成的网络，测量间隔为 15 分钟（Schimbinschi 等人，2015）。

COIL-100。$7200 \times 128 \times 128 \times 3$ 包含 100 种不同物体的 7200 张彩色照片（每种物体 72 张），拍摄时每次旋转 5 度（Nene 等人，1996年）。这是计算机视觉研究领域广泛使用的数据集。