计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202330624
61(7): 1697−1712, 2024

面向信息系统推荐与决策的高阶张量分析方法

王贝伦　张嘉琦　蔡英豪　王兆阳　谈　笑　沈　典
（东南大学计算机科学与工程学院　南京　210096）
（beilun@seu.edu.cn）

High-Order Tensor Analysis Method for Information System Recommendations and Decisions

Wang Beilun, Zhang Jiaqi, Cai Yinghao, Wang Zhaoyang, Tan Xiao, and Shen Dian
（*School of Computer Science and Engineering*, *Southeast University*, *Nanjing* 210096）

Abstract　Tensor data (or multi-dimensional array data) are often generated in information systems of various industries, such as functional magnetic resonance imaging (fMRI) data in medicine systems and user-product data in product information systems. By using these data to predict the relationship between tensor features and univariate responses, data empowerment can be achieved, providing more accurate services or solutions, such as disease decision diagnosis or product recommendations. Currently available tensor regression methods, however, present two major shortcomings: the spatial information of tensors may be lost in these models, resulting in inaccurate prediction results; the calculation cost is too high, which results in untimely solutions or services. The two problems are more severe for large-scale data with high-order structures. Therefore, in order to achieve data empowerment, that is, to use tensor data to improve the quality and efficiency of information services or solutions, we propose sparse and low-rank tensor regression model (SLTR). This model enforces sparsity and low-rankness of the tensor coefficient by directly applying ℓ_1 norm and tensor nuclear norm on it respectively, such that not only the structural information of the tensor is preserved but also the data interpretation is convenient. To make the solving procedure scalable and efficient, SLTR makes use of the proximal gradient method to optimize the hybrid regularizer, which can be easily implemented parallelly. Additionally, a tight error bound of SLTR is theoretically proved. We evaluate SLTR on several simulated datasets and one video dataset. Experimental results show that, compared with previous models, SLTR is capable to obtain a better solution with much fewer time costs.

Key words　tensor regression; parallel proximal method; data interpretation; tensor norm; classification

摘　要　张量数据（或多维数组）在各个行业的信息系统中广泛存在，例如医疗系统中的功能性磁共振成像（fMRI）数据和商品数据信息系统中的用户-产品数据.将这些数据用以预测张量特征与单变量响应之间的关系，可以实现数据赋能，提供更精准的服务或解决方案，例如疾病决策诊断或商品推荐.然而，现有的张量回归方法存在2个主要问题：一是可能丢失了张量的空间信息，导致预测结果不准确；二是计算成本过高，导致服务或解决方案不及时.对于具有高阶结构的大规模数据而言，这2点则显得更为突出.

收稿日期：2023-07-31；**修回日期**：2024-02-02
基金项目：国家自然科学基金项目（61906040, 61972085, 62276063, 6227072991）；江苏省自然科学基金项目（BK20190345, BK20190335, BK20221457）；国家重点研发计划项目（2022YFF0712400）；中央高校基本科研业务费专项资金（2242021R41177）
This work was supported by the National Natural Science Foundation of China (61906040, 61972085, 62276063, 6227072991), the Natural Science Foundation of Jiangsu Province (BK20190345, BK20190335, BK20221457), the National Key Research and Development Program of China (2022YFF0712400), and the Fundamental Research Funds for the Central Universities (2242021R41177).
通信作者：沈典（dshen@seu.edu.cn）

因此为了实现数据赋能,即利用张量数据来提高信息服务或解决方案的质量和效率,提出了稀疏低秩张量回归模型(sparse and low-rank tensor regression model, SLTR). 该模型通过对张量系数应用ℓ_1范数和张量核范数使得张量系数具有稀疏性和低秩性两大特点,这样既保留了张量的结构信息又可以方便地解释数据. 利用近端梯度方法优化了混合正则化器,使得求解过程可扩展且高效. 除此之外证明了SLTR的严格误差界. 在多个模拟数据集和一个视频数据集上的实验结果表明,SLTR相比于之前的方法,在更短的时间内获得了更好的预测性能.

关键词 张量回归;并行近端法;数据可解释性;张量范数;分类

中图法分类号 TP391

张量数据(tensor data),也称为多维数组数据,作为一种十分常见的数据形式,经常出现在各个行业的信息系统中,比如医疗数据信息系统[1-2]、视频数据信息系统[3-5]和商品数据信息系统[6-7]. 利用张量数据的信息,可以实现数据赋能,提升信息服务或者方案的质量和效率. 例如,医疗中的功能性磁共振成像(functional magnetic resonance imaging, fMRI)数据是由一系列3D数据(3阶数据)组成,其构成为(time, neuron, neuron). 医学相关的研究人员经常使用此类数据来推测病人是否患有某种脑部疾病,例如轻度认知障碍和阿尔茨海默病[1]. 另一个例子是商品数据信息系统,平台常常将信息整合为(user, product, location, timestamp)[7]的4D数据,从而方便推出兴趣分析、商品推荐等下游服务. 在推荐服务中,平台需要快速分析客户与产品的关系,并在很短的时间内向目标客户提出建议. 在实时信息不断增长的动态信息系统中[6],这一点更为突出. 因此,从N个样本中快速分析M阶张量变量$\boldsymbol{\mathcal{X}}_i \in \mathbb{R}^{p_1 \times \cdots \times p_M}$与其标量响应$y_i (i = 1,2,\cdots,N)$之间的关系是许多信息系统不可或缺的关键步骤. 将分析过程进行数学建模:

$$y_i = \langle \boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{X}}_i \rangle + \gamma_i, \tag{1}$$

其中$\langle \cdot,\cdot \rangle$是内积运算符,$\gamma_i$是假设从正态分布$\mathcal{N}(0,\alpha)$中提取的噪声,$\alpha$和$\boldsymbol{\mathcal{W}} \in \mathbb{R}^{p_1 \times \cdots \times p_M}$是需要使用所谓的张量回归来估计的系数.

在信息系统的实际应用中,张量数据通常包含2个阻碍系数预测的问题:1)超高维设置,也就是样本数量远小于变量数量. 例如,CMU2008 fMRI数据集[8]的每个样本都是具有71 553个体素的51×61×23的3D张量. 然而,数据集中仅包含了360个样本. 2)高阶数据是指阶数M很大的数据. 数据的高阶结构存在于许多领域. 例如,在处理视频分类、视频监控或手势识别等处理视频数据的任务时[3-4,9-10],需要收集(time, pixel, pixel, color_channel)形式的高阶张量数据.

张量数据的这些特性给张量回归带来了2个挑战:1)低性能. 高维设置会导致估计的解决方案出现问题,因为这试图从有限的观测值中推断一个大规模模型,即观测值的数量明显少于未知数的数量. 2)昂贵的计算成本. 高阶数据上的张量回归包含大量未知变量,因此需要非常昂贵的计算成本. 近年来提出的方法引入了稀疏或低秩约束来解决这2个挑战. 一方面,稀疏方法通常会过滤掉“无用”变量(通常通过变量选择获得),以减少高维情况下的变量数量;另一方面,对问题实施低秩约束会减少未知变量的数量,这使得张量回归问题更容易处理并且更快地解决.

本文将先前提出的具有这2个结构约束的张量回归方法分为三大类. 第1类是最常用的方法,它将现有的线性回归模型直接应用于向量化张量数据,例如将张量元素逐个堆叠到向量中. 尽管直观且直接,但这种方法可能会丢失数据的空间结构,例如图片中的像素关系或视频中的时间顺序. 为了保留空间结构,引入的第2类方法[11-12]基于CP(candecomp/parafac)分解,它表示对有多个分量的M阶张量添加结构约束. 然而,所有基于CP分解的方法都存在收敛速度慢[13]和预测不准确的缺点,因为最佳CP近似可能不存在[14]. 为了避免这些问题,第3类方法[15-16]直接对张量数据应用结构约束(如稀疏性的ℓ_1范数),而不是其分量向量. 然而,由于计算过程中包含多个高计算复杂度内容(包括优化多个核范数),这类方法的计算成本通常很高. 由于其缺点,上述三类方法无法快速有效地获得大规模数据的解决方案,因此需要一个快速且可扩展的张量回归估计器.

本文提出了稀疏低秩张量回归模型(sparse and low-rank tensor regression model, SLTR)方法,该方法直接应用ℓ_1范数和系数$\boldsymbol{\mathcal{W}}$的核范数,以此来降低张量回归问题的复杂性. 除此之外利用并行近端方法,提出了一种快速且可扩展的解决方案. 由于该方法可以通过多线程计算或GPU并行实现,其计算时间成

本被大大降低，同时数据的结构信息也可以保留. 实验表明，SLTR 比以前的方法要快得多，同时保持的性能不弱于其他方法. 总而言之，本文做出了 4 方面贡献：

1）提出稀疏低秩张量回归模型. 通过在张量上应用ℓ_1范数和张量核范数，提出稀疏低秩张量回归模型.

2）提出快速且可扩展的模型方法. 为提出的模型提供了快速计算的解决方案，这可以通过 2 层并行化得到. 同时还证明了 SLTR 在并行实现时具有较低的时间成本.

3）理论上证明稀疏低秩张量回归的收敛速度. 从理论上证明了模型的尖锐误差界限. 具体来说，本文为 3 阶数据提供了更精确的误差界，证明了应用ℓ_1范数和核范数的稀疏低秩张量回归的误差界. 此外还证明了张量核范数的可分解性，这有助于其高维分析.

4）完成模拟和真实数据集的实验. 在多个模拟数据集和 UCF101 视频数据集[9]上，将 SLTR 的预测精度和计算时间成本与 4 种最先进的基线方法进行了比较. 结果表明 SLTR 可以用更少的时间成本来获得准确且可解释的解决方案.

1 符　　号

用$\boldsymbol{\mathcal{A}}\in\mathbb{R}^{p_1\times\cdots\times p_M}$表示一个$M$阶张量，大写字符表示矩阵$\boldsymbol{A}\in\mathbb{R}^{p_1\times p_2}$，小写字符表示向量$\boldsymbol{y}\in\mathbb{R}^{p_1}$. 对于矩阵，$\|\cdot\|_1$表示逐元素的$\ell_1$范数，$\|\cdot\|_\infty$表示逐元素的$\ell_\infty$范数，$\|\cdot\|_2$表示谱范数，$\|\cdot\|_{\mathrm{F}}$是弗罗贝尼乌斯范数，$\|\boldsymbol{A}\|_*$表示矩阵核范数，$\|\boldsymbol{\mathcal{A}}\|_*$表示张量核范数，这些可以根据上下文来区分.

接下来介绍张量数据的基本操作. 2 个张量$\boldsymbol{\mathcal{A}},\boldsymbol{\mathcal{B}}\in\mathbb{R}^{p_1\times\cdots\times p_M}$的内积是每个条目的乘积之和，定义为$\langle\boldsymbol{\mathcal{A}},\boldsymbol{\mathcal{B}}\rangle=\sum_{i_1=1}^{p_1}\cdots\sum_{i_M=1}^{p_M}\mathcal{A}_{i_1\cdots i_M}\mathcal{B}_{i_1\cdots i_M}$. M阶张量$\boldsymbol{\mathcal{A}}$与矩阵$\boldsymbol{A}\in\mathbb{R}^{J\times p_m}$的$m$阶乘积，表示为$\boldsymbol{\mathcal{A}}\times_m\boldsymbol{A}$，其结果是形状为$\mathbb{R}^{p_1\times\cdots\times p_{m-1}\times J\times p_{m+1}\times\cdots\times p_M}$的张量. 这里，$m$阶乘积的每个条目为$(\boldsymbol{\mathcal{A}}\times_m\boldsymbol{A})_{i_1\cdots i_{m-1}ji_{m+1}\cdots i_M}=\sum_{i_m=1}^{p_m}\mathcal{A}_{i_1\cdots i_M}a_{ji_m}$.

附录 A 中提供了详细的符号介绍.

2 背　　景

2.1 线性回归的基本估计器

线性回归（LR）总是假设预测变量和响应之间存在线性关系；例如，$\boldsymbol{y}=\boldsymbol{X}\boldsymbol{w}+\boldsymbol{\gamma}$，其中$\boldsymbol{y}\in\mathbb{R}^N$表示响应，$\boldsymbol{X}\in\mathbb{R}^{N\times p}$表示低阶变量，$\boldsymbol{w}\in\mathbb{R}^p$是需要估计的系数，$\boldsymbol{\gamma}$是噪声项. $\boldsymbol{w}$的传统估计是$\hat{\boldsymbol{w}}=\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}$. 然而，在高维设置中，样本数量小于变量数量（$N<p$），协方差$\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$不再可逆，从而给估计带来困难. 在过去的几十年中，人们提出了许多方法来解决高维环境中估计困难的问题. 最常用的方法是变量选择，它通过仅选择有用变量的子集来解决问题，从而大大减少p. 文献 [17] 提出了一种名为基本估计器（elementary estimator, EE）的高维线性回归快速估计器，旨在解决：

$$\begin{cases}\hat{\boldsymbol{w}}=\arg\min\limits_{\boldsymbol{w}}\|\boldsymbol{w}\|_1,\\ \text{s.t. }\left\|\boldsymbol{w}-\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\varepsilon\boldsymbol{I}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}\right\|_\infty\leqslant\lambda,\end{cases}\tag{2}$$

其中$\boldsymbol{I}$是单位矩阵，ε是处理不可逆样本协方差的参数. 该估计器有一个闭式解$\hat{\boldsymbol{w}}=S_\lambda((\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\varepsilon\boldsymbol{I})^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y})$，其中$S_\lambda(\cdot)$是逐元素软阈值运算符，即$[S_\lambda(\boldsymbol{w})]_i=\mathrm{sgn}(\boldsymbol{w}_i)\times\max\{|w_i|-\lambda,0\}$.

EE 的思想是通过最小化ℓ_1目标函数来获得稀疏性，同时通过ℓ_1的对偶范数（即约束中的ℓ_∞范数）回归来预测变量$\boldsymbol{y}$. 由于上述方法仅适用于向量（即 2 阶）数据，因此本文将推导出一个新模型，通过张量回归来学习标量响应与高阶张量之间的线性关系.

2.2 正则化张量回归

为了降低问题的复杂性，最先进的张量回归方法总是给出稀疏或低秩的估计假设，并使用基于范数的正则化来求解模型.

在许多研究中，稀疏性是通过元素级ℓ_1范数来实现的[11-12]. 大规模的张量数据的问题求解异常耗时，尤其是在实时或快速响应的情境下. 通过引入稀疏性，能够将注意力集中在少数有实际意义的数据点上，过滤掉“无用”的变量，减少了变量的数量，从而降低了计算和存储的负担. 具体来说，假设权重张量$\boldsymbol{\mathcal{W}}$中只有一小部分系数是有意义的，因此$\boldsymbol{\mathcal{W}}$有许多零元素. 本文使用逐元素ℓ_1范数来约束权重的稀疏性

$$\|\boldsymbol{\mathcal{W}}\|_1=\sum_{i_1=1}^{p_1}\cdots\sum_{i_M=1}^{p_M}\left|\boldsymbol{\mathcal{W}}_{i_1\cdots i_M}\right|.\tag{3}$$

另一方面，之前的工作[15-16]也假设$\boldsymbol{\mathcal{W}}$是低秩的. 高维度数据可能导致模型过于复杂，难以拟合数据，甚至引发过拟合问题. 在这种情况下，强制执行低秩约束有助于降低模型拟合数据的复杂性，同时保留关键的数据模式. 在数据驱动的信息系统服务领域，这可以有效地处理高维数据，为服务的开发提供更坚实的基础. 可以通过张量分解进行优化，其中 CP 分解和 Tucker 分解是 2 种广泛使用的张量分解技术.

与 CP 分解相比，Tucker 分解是奇异值分解(SVD)的直接扩展，因此它更具可解释性并且可以与 SVD 方法配合. 因此在本文中，假设$\mathcal{W}$可以通过 Tucker 分解[18-19]分解为

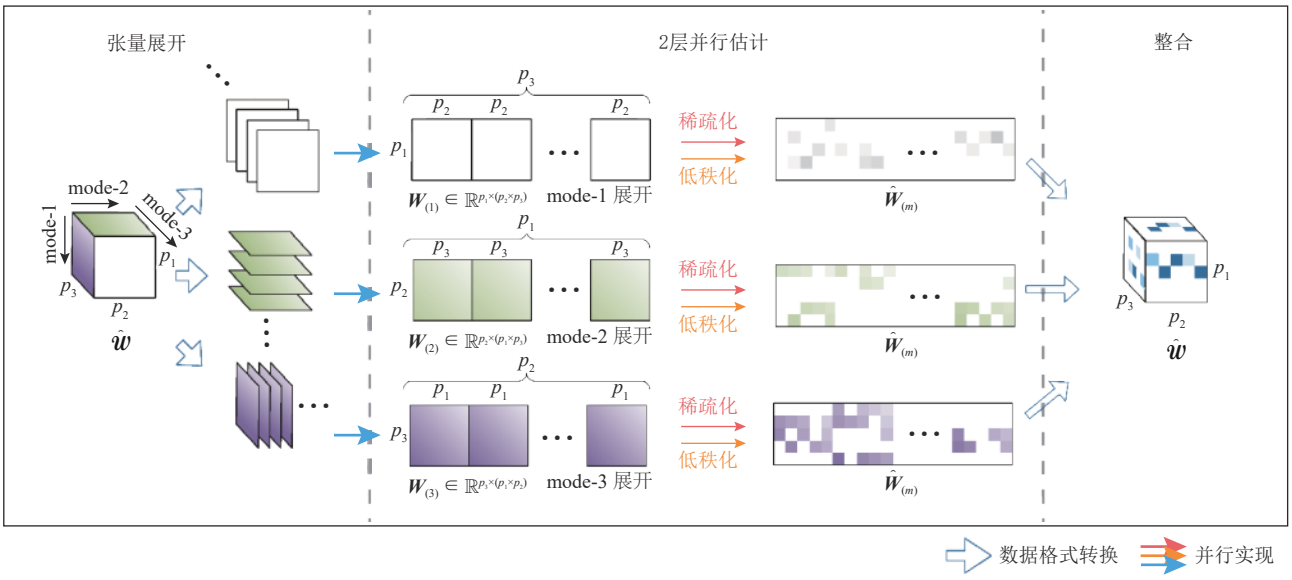
$$\mathcal{W}=\mathcal{C}\times_1\boldsymbol{W}_{(1)}\times_2\boldsymbol{W}_{(2)}\times_3\cdots\times_M\boldsymbol{W}_{(M)},\tag{4}$$

其中$\mathcal{C}\in\mathbb{R}^{p_1\times\cdots\times p_M}$是核心张量，$\times_m$是$m$阶乘积运算符，矩阵$\boldsymbol{W}_{(m)}\in\mathbb{R}^{p_m\times\prod_{k\neq m}p_k}$是张量$\mathcal{W}$沿第$m$阶展开的结果，图 1 展示了 3 阶张量的示例. 如此一来通过低秩的$\{\boldsymbol{W}_{(1)},\cdots,\boldsymbol{W}_{(M)}\}$和低秩的核心张量，可以去尽可能地估计一个低秩的权重张量$\mathcal{W}$.

但单独通过 Tucker 分解很难直接获得张量的秩. 而且正如文献 [20] 所说，张量秩的计算是 NP 困难的. 幸运的是，文献 [21–22] 证明了在M个矩阵核范数求和的公式中，可以使用凸张量核范数正则化作为张量秩的凸松弛

$$\|\mathcal{W}\|_*=\frac{1}{M}\sum_{m=1}^{M}\|\boldsymbol{W}_{(m)}\|_*,\tag{5}$$

其中$\|\boldsymbol{W}_{(m)}\|_*$是矩阵核范数. $\|\mathcal{W}\|_*$是矩阵核范数的扩展，并且在文献 [21] 中被证明能够准确可靠地自动获得低秩张量.



注：首先沿每阶展开张量；然后对于每种阶展开，并行估计权重，同时通过正则化保证稀疏性和低秩；最后组合每阶的结果，得到最终结果.

Fig. 1 The basic idea of SLTR

图 1 SLTR 的基本思想

最近，文献 [15] 提出了一种名为 Remurs 的最先进的张量回归模型，它在优化问题中对低秩和稀疏进行约束. 文献 [15] 利用交替方向乘子法(alternating direction method of multipliers, ADMM)[23]来估计$\mathcal{W}$，但对于高阶结构数据问题而言，其计算成本很高.

3 方 法

为了解决高计算(时间)成本的问题，本文将 EE 扩展到张量数据来并行化地解决稀疏和低秩正则化，提出 SLTR 方法. 具体来说，给定M阶张量$\mathcal{X}\in\mathbb{R}^{N\times p_1\times\cdots\times p_M}$的$N$个样本以及相应的响应$\boldsymbol{y}\in\mathbb{R}^N$，SLTR 旨在解决

$$\begin{cases}\hat{\mathcal{W}}=\arg\min\limits_{\mathcal{W}}\|\mathcal{W}\|_1+\|\mathcal{W}\|_*,\\ \text{s.t.}\begin{cases}\left\|\mathcal{W}-\mathcal{T}_{\mathcal{P}}\left(\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\varepsilon\boldsymbol{I}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}\right)\right\|_\infty\leqslant\lambda,\\ \left\|\mathcal{W}-\mathcal{T}_{\mathcal{P}}\left(\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\varepsilon\boldsymbol{I}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}\right)\right\|_2\leqslant\tau.\end{cases}\end{cases}\tag{6}$$

这里，$\|\cdot\|_1$和$\|\cdot\|_*$被用来解决稀疏和低秩正则化问题. 采用类似式(2)的框架，$\|\cdot\|_\infty$是$\|\cdot\|_1$的对偶范数，$\|\cdot\|_2$是$\|\cdot\|_*$的对偶范数①. 张量化运算$\mathcal{T}_{\mathcal{P}}(\cdot)$可以被视为向量化运算的逆运算，在给定每阶的维数$\mathcal{P}=\{p_1,p_2,\cdots,p_M\}$情况下，它可以将形$\mathbb{R}^{\prod_m p_m}$的向量转换为相应的张量. 参数$\varepsilon,\tau,\lambda$是可以调整的参数. 此外，

① 此处没有直接将这些范数函数应用于张量数据. 请注意，ℓ_1范数和张量核范数都可以轻松地重新表示为多个矩阵范数的组合，可以直接定义它们以及张量上相应的双范数，参见式(7). 关于对偶范数的详细内容见附录 C.

$\boldsymbol{X}=concat(\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_N)\in\mathbb{R}^{N\times\prod_m p_m}$，其中 $\boldsymbol{X}_i=vec(\boldsymbol{\mathcal{X}}_i)$ 是通过对第i个样本进行向量化而获得的.

3.1 2 层并行方案

由元素级ℓ_1范数的定义，得到$\|\boldsymbol{\mathcal{W}}\|_1=\dfrac{1}{M}\sum_{m=1}^{M}\|\boldsymbol{W}_{(m)}\|_1$. 将其与式 (5) 张量核范数的定义相结合，可以将式(6)分解为M个并行的子任务：

$$\begin{cases}\arg\min\limits_{\boldsymbol{W}_{(1)},\cdots,\boldsymbol{W}_{(M)}}\dfrac{1}{M}\sum\limits_{m=1}^{M}\|\boldsymbol{W}_{(m)}\|_1+\dfrac{1}{M}\sum\limits_{m=1}^{M}\|\boldsymbol{W}_{(m)}\|_*,\\ \text{s.t.}\begin{cases}\|\boldsymbol{W}_{(m)}-\tilde{\boldsymbol{W}}_{(m)}\|_\infty\leqslant\lambda,\\ \|\boldsymbol{W}_{(m)}-\tilde{\boldsymbol{W}}_{(m)}\|_2\leqslant\tau.\end{cases}\end{cases}\tag{6}$$

这里，$\tilde{\boldsymbol{\mathcal{W}}}=\mathcal{T}_{\mathcal{P}}\left(\left(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}+\varepsilon\boldsymbol{I}\right)^{-1}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{y}\right)$可以被视为$\boldsymbol{\mathcal{W}}$的初始近似值并预先计算 1 次. 在计算完数据阶数$m=1,2,\cdots,M$之后，将它们组合起来以获得最终的估计.

$$\hat{\boldsymbol{\mathcal{W}}}=\frac{1}{M}\sum_{m=1}^{M}\mathcal{F}_m\left(\hat{\boldsymbol{W}}_{(m)}\right).\tag{7}$$

这里，$\mathcal{F}_m(\cdot)$是将矩阵$\boldsymbol{A}\in\mathbb{R}^{p_m\times\prod_{k\neq m}p_k}$沿着第$m$阶折叠成对应的张量$\boldsymbol{\mathcal{A}}\in\mathbb{R}^{p_1\times p_2\times\cdots\times p_M}$的运算符.

式(7)中的每个$\boldsymbol{W}_{(m)}$都可以由并行近端算法[24]独立求解，该算法在满足 2 个约束条件的同时，优化了核范数和ℓ_1范数. 如果定义$\boldsymbol{w}\triangleq\boldsymbol{W}_{(m)}$，那么估计$\boldsymbol{W}_{(m)}$相当于

$$\begin{cases}\arg\min f_1(\boldsymbol{w}_1)+f_2(\boldsymbol{w}_2)+f_3(\boldsymbol{w}_3)+f_4(\boldsymbol{w}_4),\\ \text{s.t. }\boldsymbol{w}_1=\boldsymbol{w}_2=\boldsymbol{w}_3=\boldsymbol{w}_4,\end{cases}\tag{8}$$

其中，$f_1(\boldsymbol{w})=\|\boldsymbol{w}\|_1$，$f_2(\boldsymbol{w})=\|\boldsymbol{w}\|_*$，$f_3(\boldsymbol{w})=\mathcal{I}_{\{\|\boldsymbol{w}-\tilde{\boldsymbol{w}}_{(m)}\|_\infty\leqslant\lambda\}}(\boldsymbol{w})$，$f_4(\boldsymbol{w})=\mathcal{I}_{\{\|\boldsymbol{w}-\tilde{\boldsymbol{w}}_{(m)}\|_2\leqslant\tau\}}(\boldsymbol{w})$. $\mathcal{I}_C(\boldsymbol{w})$表示集合$C$的指示函数，如果$\boldsymbol{w}\in C$，则$\mathcal{I}_C(\boldsymbol{w})=0$，否则$\mathcal{I}_C(\boldsymbol{w})=\infty$. 为了优化式(9)，本文提出了一种如算法 1 中所总结的基于并行近似的算法. 4 个邻近算子的计算是独立的，因此可以并行计算.

算法 1. 基于并行近端的 SLTR 方法.

输入：$\boldsymbol{\mathcal{X}}\in\mathbb{R}^{N\times p_1\times p_2\times\cdots\times p_M}$，$\boldsymbol{y}\in\mathbb{R}^N$，初始近似值 $\tilde{\boldsymbol{\mathcal{W}}}$，最大迭代次数$T$，学习率$\rho\in[0,2]$，以及调整参数$\boldsymbol{c}=(\lambda,\lambda,\tau,\tau)$；

输出：$\hat{\boldsymbol{\mathcal{W}}}=\dfrac{1}{M}\sum_{m=1}^{M}\hat{\boldsymbol{\mathcal{W}}}_m$.

① for $m=1$ to M 并行 do

② 初始化$\boldsymbol{W}_{(m)}=\boldsymbol{W}^1_{(m)1}=\boldsymbol{W}^1_{(m)2}=\boldsymbol{W}^1_{(m)3}=\boldsymbol{W}^1_{(m)4}=\tilde{\boldsymbol{W}}_{(m)}$；

③ for $t=1$ to T do

④ for $i=1,2,3,4$ 并行 do

⑤ $\boldsymbol{a}_i^t=prox_{4c_jf_i}\left(\boldsymbol{W}^t_{(m)i}\right)$；

⑥ end for

⑦ $\boldsymbol{a}^t=\dfrac{1}{4}\sum_{i=1}^{4}\boldsymbol{a}_i^t$；

⑧ for $i=1,2,3,4$ do

⑨ $\boldsymbol{W}^{t+1}_{(m)i}=\boldsymbol{W}^t_{(m)i}+\rho\left(2\boldsymbol{a}^t-\boldsymbol{W}_{(m)}-\boldsymbol{a}_i^t\right)$；

⑩ end for

⑪ $\boldsymbol{W}_{(m)}=\boldsymbol{W}_{(m)}+\rho\left(\boldsymbol{a}^t-\boldsymbol{W}_{(m)}\right)$；

⑫ end for

⑬ $\hat{\boldsymbol{\mathcal{W}}}_m=\mathcal{F}_m(\boldsymbol{W}_{(m)})$；

⑭ end for

由于求解每个$\boldsymbol{W}_{(m)}$的过程可以并行实现，并且在求解每个子问题时，4 个邻近算子的计算可以进一步并行化，因此 SLTR 方法能够以 2 层并行的方式获得解. 附录 B 给出了这 4 个近端算子的公式.

3.2 SLTR 应用示例

本节简要说明了所设计的 SLTR 张量回归方法是如何在信息系统中进行应用的. 以第 6.2 节的 UCF101 数据集上的动作识别实验为例，说明了这一应用可以提高系统的智能性和个性化服务水平，并为用户和决策者提供更多有用的信息和功能.

医疗康复监测任务在医疗保健领域至关重要，它涉及对患者康复过程的实时追踪和深度分析. SLTR 正如第 6.2 节中的动作识别结果所揭示的那样，能够快速准确地分析和识别视频中的动作，通过与医疗信息系统的融合，可以为康复领域带来更多的潜力. 举例来说，当患者进行康复训练时，首先系统利用摄像头捕获他们的运动，接着 SLTR 可以对这段形式为(time, pixel, pixel, color_channel)的 4 维张量数据进行快速动作识别. 这一过程不仅可以确保患者按照正确的方式执行康复动作，避免不正确的姿势或运动，还能够提供有关患者康复进展的实时反馈.

除此之外，也可以将 SLTR 方法与智能视频监控系统相结合. 在智能视频监控中，及时检测和响应异常事件至关重要. SLTR 能够快速分析识别视频流张量数据中的动作，从而检测到不寻常的行为或事件. 例如，监控摄像头可以自动触发警报，当系统检测到可能的入侵、盗窃或其他异常活动时，警报会立即传送给相关部门或安全人员，从而提高了监控系统的效率和效力.

总的来说，SLTR 张量回归方法在信息系统中的应用加强了数据的智能分析、个性化服务和决策支持能力，为各行业的信息系统提供了更多的创新和优化机会.

4 理论分析

4.1 时间复杂度分析

SLTR 的总计算复杂度为 $O\left(\max_m\left\{p_m\left(\prod_{k\neq m}p_k\right)^2\right\}\right)$，其中分为计算初始近似值和并行求解 M 个估计 2 部分：

1)计算初始近似值 $\tilde{\boldsymbol{W}}$ 只涉及矩阵乘法、矩阵求逆等简单运算，可以通过多线程计算或 GPU 轻松加速，并作为方法的前提条件快速获得. $\tilde{\boldsymbol{W}}$ 仅需计算 1 次并在方法中重复使用.

2)并行求解总共 M 个估计时，每个子任务由时间复杂度为 $O\left(p_m\left(\prod_{k\neq m}p_k\right)^2\right)$ 的 SVD 过程主导. 由于并行性，求解 SLTR 的计算时间复杂度由 M 个估计中最长的子任务主导，即 $O\left(\max_m\left\{p_m\left(\prod_{k\neq m}p_k\right)^2\right\}\right)$.

4.2 可分解性和误差界分析

首先在文献 [25] 的帮助下证明式 (5) 中张量核范数是可分解的.

定理 1. 给定一对正确定义的子空间 $\left(\mathcal{M},\overline{\mathcal{M}}^{\perp}\right)$，其中 $\mathcal{M}\subseteq\overline{\mathcal{M}}$，张量核范数 $\|\cdot\|_*$ 是可分解的. 具体来说，对于所有 $\boldsymbol{\mathcal{A}}\in\mathcal{M}$ 和 $\boldsymbol{\mathcal{B}}\in\overline{\mathcal{M}}^{\perp}$，有 $\|\boldsymbol{\mathcal{A}}+\boldsymbol{\mathcal{B}}\|_*=\|\boldsymbol{\mathcal{A}}\|_*+\|\boldsymbol{\mathcal{B}}\|_*$.

附录 D 中提供了子空间对 $\left(\mathcal{M},\overline{\mathcal{M}}^{\perp}\right)$ 的定义和定理 1 的证明.

然后陈述误差界分析的 2 个基本假设为：

1) 稀疏性(C1). 假设真实系数 $\boldsymbol{\mathcal{W}}^*$ 具有 k 个非零元素.

2) 低秩性(C2). 真实系数 $\boldsymbol{\mathcal{W}}^*$ 是 R 阶张量，其中 $R=\max_{\boldsymbol{\mathcal{A}}\in\mathbb{R}^{P_1\times P_2\times\cdots\times P_M}}(r_\perp(\boldsymbol{\mathcal{A}}))$，$r_\perp(\boldsymbol{\mathcal{A}})$ 表示 $\boldsymbol{\mathcal{A}}$ 的正交秩. 正交秩是满足 $\boldsymbol{\mathcal{A}}=\sum_{r=1}^{r_\perp(\boldsymbol{\mathcal{A}})}\boldsymbol{\mathcal{U}}_r$ 的最小值，其中 $\langle\boldsymbol{\mathcal{U}}_{r_1},\boldsymbol{\mathcal{U}}_{r_2}\rangle\geqslant 0$，$r_1\neq r_2$，$1\leqslant r_1\leqslant r_\perp(\boldsymbol{\mathcal{A}})$，$1\leqslant r_2\leqslant r_\perp(\boldsymbol{\mathcal{A}})$.

基于张量核范数的可分解性和上述 2 个假设证明了 SLTR 的收敛速度. 本文证明了对系数直接采用约束的张量回归模型的误差界.

定理 2. 假设真实系数张量 $\boldsymbol{\mathcal{W}}^*$ 满足条件稀疏性和低秩性. 此外，假设通过控制满足约束的参数 λ 和 τ 来求解式(6)，那么最优解满足误差界：

$$\|\hat{\boldsymbol{\mathcal{W}}}-\boldsymbol{\mathcal{W}}^*\|_{\mathrm{F}}\leqslant 4\sqrt{2}\left(\lambda\sqrt{\prod_{m=1}^{M}p_m}+\tau\sqrt{R}\right). \tag{9}$$

推论 1. 如果系数是 3 阶张量，使得 $\boldsymbol{\mathcal{W}}\in\mathbb{R}^{p_1\times p_2\times p_3}$ 且条件稀疏性和低秩性对于真实系数 $\boldsymbol{\mathcal{W}}^*$ 而言成立，式(6)的最优解满足误差界限：

$$\|\hat{\boldsymbol{\mathcal{W}}}-\boldsymbol{\mathcal{W}}^*\|_{\mathrm{F}}\leqslant 4\sqrt{2}\left(\lambda\sqrt{\prod_{m=1}^{M}p_m}+\tau\max_{k=1,2,3}\{R'_k\}\right), \tag{10}$$

其中 $r_m=\mathrm{rank}\left(\boldsymbol{\mathcal{W}}_{(m)}\right)$ 表示展开张量的秩，$R'_1=\sqrt{r_1\min\{r_2,r_3\}}$，$R'_2=\sqrt{r_2\min\{r_1,r_3\}}$，$R'_3=\sqrt{r_3\min\{r_1,r_2\}}$.

更详细的证明见附录 E.

5 涉及张量的相关工作

文献 [11–12,26] 是基于 CP 分解提出的. 一般来说，这些方法不是直接估计系数张量 $\boldsymbol{\mathcal{W}}$，而是推断每个子任务中每个分量的向量. 例如，文献 [11] 提出使用广义线性模型(generalized linear model, GLM)来解决每个子任务的广义线性张量回归模型(generalized linear tensor regression model, GLTRM). 此外，最近文献 [12] 提出了利用分而治之策略的 SURF，其中子任务具有和 ENet[27] 类似的公式. 几乎所有基于 CP 分解的方法都需要 CP 秩 R 的先验知识. 然而在实际应用中始终对其了解甚少. 即使可以使用交叉验证等技术从一定范围中选择 R 的值，但是对于大规模数据来说，选择 R 值的过程也会变得复杂且计算成本昂贵. 此外，R 越大，这些方法所需的计算时间就越多.

另一组方法将结构约束直接应用于 $\boldsymbol{\mathcal{W}}$ 而不是应用在分量向量. 例如，在 Remurs[15] 中，使用了张量核范数和 ℓ_1 范数. 然而，这些方法的计算成本很高，因为它们的目标函数中存在非微分正则化器 ℓ_1 范数或核范数并且缺乏并行性. 文献 [28] 评论了许多张量回归模型. 在表 1 中，将 SLTR 与最先进的张量回归模型进行比较. 显然，SLTR 在模型性能和计算时间复杂度方面都优于其他方法.

6 实 验

本节将介绍本文进行的实验.实验结果表明，SLTR 可以用更少的时间成本来获得准确且可解释的解决方案.

1) 基线. 使用之前提出的 4 种方法作为基线，代表不同组的张量回归方法，包括：1)线性回归模型，具体来说是 Lasso 和 ENet(ℓ_1 范数和 ℓ_2 范数之间的权衡比为 0.5)；2)Remurs[15]；3)SURF[12]. 所有方法均在 MATLAB 中实现.

2) 指标. 对于模拟数据的实验，记录了所有方法的计算时间成本和预测均方误差(mean squared error, MSE). 对于真实世界视频分类数据的实验，呈现了

Table 1 Comparison Between SLTR and Other Tensor Regression Models

表 1 SLTR 与其他张量回归模型的比较

方法	计算瓶颈	无需指定秩	充分稀疏	结构保留
SLTR（本文）	$O\left(\max_m\{p_m P_{\backslash m}^2\}\right)$	✓	✓	✓
Remurs	$O\left(\sum_{m=1}^{M}\{p_m P_{\backslash m}^2\}\right)$	✓	✓	✓
GLTRM	$O\left(R\sum_{m=1}^{M} p_m^3\right)$	×	✓	✓
orTRR	$O\left(MP^3\right)$	✓	×	✓
SURF	$O\left(TN\sum_{m=1}^{M} P_{\backslash m}\right)$	×	✓	✓
LR	$O\left(NP^2\right)$	×	✓	×

注：T表示迭代方法的迭代次数，N是样本数，M是数据阶数，R是 CP-rank. $P=\prod_{m=1}^{M} p_m$和 $P_{\backslash m}=\prod_{k\neq m} p_m$. 无需指定秩是指方法能否自动地探索所需要的最佳的张量秩而不是通过超参数提前指定；充分稀疏是指模型权重是否具有充分的稀疏性；结构保留是指方法能否保留数据中的结构信息.

所有方法的计算时间成本和 ROC 曲线下的面积（area under the ROC curve, AUROC）.

3) 调整参数. 所有方法的调整参数都是通过 1 折交叉验证过程选择的，该过程以验证数据集的平均性能作为选择标准. 有关调整参数范围的详细说明见附录 F.

4) 其他设置. 将所有方法的最大迭代次数设置为 1 000，并让它们在 $\frac{\|\boldsymbol{\mathcal{W}}^{t+1}-\boldsymbol{\mathcal{W}}^{t}\|_{\mathrm{F}}}{\|\boldsymbol{\mathcal{W}}^{t}\|_{\mathrm{F}}}\leqslant 10^{-4}$时终止. 将每个实验运行 10 次，并报告这 10 次实验的指标平均值. 此外，在报告计算（时间）成本时，删除了计算初始近似值$\tilde{\boldsymbol{\mathcal{W}}}$所花费的时间，因为这个初始近似值被认为是求解 SLTR 的前提，并且在良好的实验环境下可以在很短的时间内获得近似.

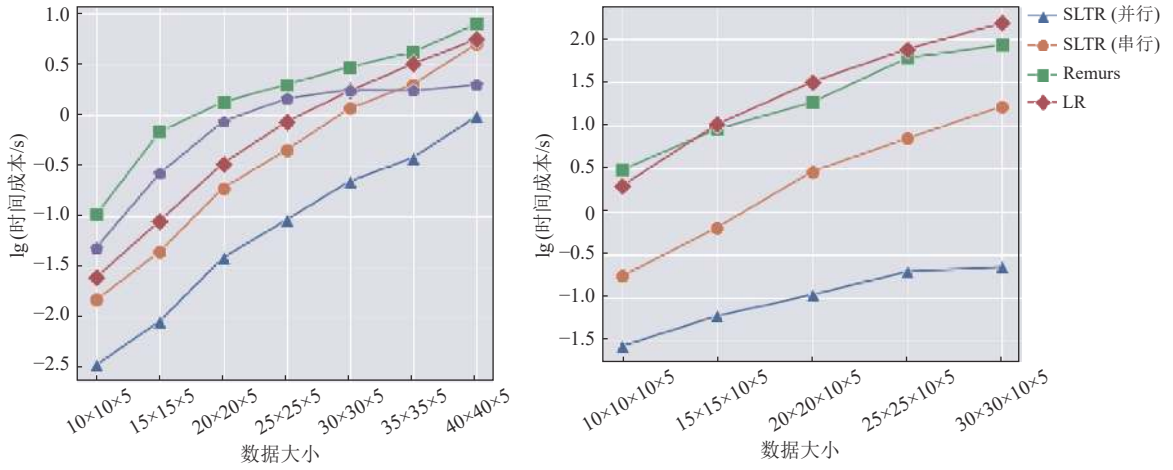
6.1 模拟数据实验

模拟数据集通过 3 个步骤生成：1）生成$\boldsymbol{\mathcal{W}}^{*}\in\mathbb{R}^{p_1\times p_2\times\cdots\times p_M}$和$\boldsymbol{\mathcal{X}}\in\mathbb{R}^{N\times p_1\times p_2\times\cdots\times p_M}$，其中每个元素均取自正态分布$\mathcal{N}(0,1)$. 2）随机设置$\boldsymbol{\mathcal{W}}$的$s$个元素为 0. 3）计算关于$y_i=\langle\boldsymbol{\mathcal{W}}^{*},\boldsymbol{\mathcal{X}}_i\rangle+\varepsilon_i$的响应$\boldsymbol{y}\in\mathbb{R}^{N}$. 这里，$\alpha$控制噪声的比例（一般设置为一个较小的值），$\varepsilon_i$也是从正态分布$\mathcal{N}(0,1)$生成的.

首先，在 3D 和 4D 模拟数据上实验 SLTR. 设置稀疏度$s=80\%$和噪声因子$\alpha=0.1$. 3D 和 4D 数据的样本数量分别通过$N=8\%\prod_{m=1}^{M} p_m$确定. 所有方法的时间成本如图 2 所示. 显然与基线相比，不仅 SLTR 的串行版本具有较低的时间成本，而且 SLTR 的并行实现在时间成本上也取得了更明显的改进. 值得注意的是，SURF 的代码不适用于 4D 数据，因此在子图中相应地省略了它，这也表明了其在更广泛应用中的局限性.

表 2 中呈现了这些模拟数据集的估计 MSE. 结果表明，在大多数情况下，SLTR 获得了最佳解决方案，而在其他情况下，SLTR 计算的估计仅比最佳解决方案差一点. 图 2 和表 2 中的实验结果表明 SLTR 能够花费更少的时间获得足够好的解决方案.

此外，让数据的形状为$20\times20\times5$，N从 50 到 400 变化，设置稀疏度$s=80\%$和$\alpha=0.1$. 附录 F 中表 F4 所示的 MSE 值表明 SLTR 在几乎所有条件下都获得了最优解. 注意，对于$N=150$，SLTR 的 MSE 值是 0.929 5，仅仅比 Remurs 的 MSE 值（0.919 0）多一点，但这个值比其他方法的 MSE 值少很多，比如 SURF 的是 0.995 3



(a) 3D 数据下的时间成本 (b) 4D 数据下的时间成本

Fig. 2 Logarithmic computational (time) cost of SLTR and baselines on simulated datasets with the sparsity level $s=80\%$

图 2 在稀疏级别 $s=80\%$ 的模拟数据集上 SLTR 和基线的对数计算（时间）成本

Table 2 MSE of Different Methods on Simulated Dataset with Different Sizes of Data

表 2 各个方法在具有不同数据大小的模拟数据集的 MSE

数据大小	SLTR（本文）	Remurs	SURF	Lasso	ENet
30×30×5	**0.918 6**	0.9190	0.928 9	1.938 1	1.937 7
35×35×5	**0.933 6**	0.9370	0.952 7	2.014 7	2.014 7
40×40×5	0.9073	**0.907 2**	1.000 6	2.105 9	2.106 5
20×20×10×5	**0.915 0**	0.9177		2.138 8	2.137 3
25×25×10×5	**0.907 1**	0.9101		1.969 6	1.969 6
30×30×10×5	**0.911 0**	0.9123		1.975 4	1.974 5

注：粗体数字表示最佳结果，下划线值表示次佳结果.

和 ENet 的是 1.6502.

此外记录了 SLTR 和基线交叉验证的总时间来比较计算成本. 由于 SURF 的精度较差，因此不予考虑. 结果显示在表 3 中. 实验中，为了公平起见，SLTR 和 Remurs 的交叉验证组数分别为 147 和 49. 结果表明，在交叉验证中 SLTR 的时间比 Remurs 少 1~2 个数量级，并且随着数据规模的增大，这种优势变得越来越明显. 原因就在于 SLTR 的初始估计在交叉验证之前仅计算 1 次，而对于 Remurs，则在每次迭代中都需要再次计算.

Table 3 Total Time of Cross Validation in Different Methods

表 3 不同方法的交叉验证总时间 s

数据大小	SLTR（本文）	Remurs
30×30×5	**16.793 568**	510.050 226
35×35×5	**20.212 862**	602.292 492
40×40×5	**6.509 343**	684.737 448
20×20×10×5	**10.975 796**	1 474.090 235
25×25×10×5	**16.936 689**	3 060.085 998
30×30×10×5	**23.276 677**	3 394.478 247

注：粗体数字为最短时间， 表示计算成本最小.

6.2 动作识别视频数据集实验

动作识别任务经常应用于许多信息系统当中，包括医疗系统、视频监控系统等. UCF101[9] 是一个用真实动作视频做动作识别的数据集，本文在该数据集上测试了提出的方法. 该数据集从 YouTube 收集了 101 个动作类别的 13 320 个视频. 每个视频的时长不同，从小于 2s 到大于 10s 不等，每帧的分辨率为 320 像素×240 像素. 实验专注于二元分类任务，并选择 3 对类别：ApplyEyeMakeup 与 ApplyLipstick、BaseballPitch 与 Basketball 以及 BodyWeightSquats 与 Bowling. 从每个视频中统一提取固定间隔的 15 帧并将其转换为灰度图. 对于每一帧，通过平均相邻像素将其大小调整为32像素×24像素，所以每个样本都是一个15×32×24张量. 对于每对类别，选择80%的样本作为训练样本，20% 的样本用于测试. 实验采用 2 个指标，即运行时间和 AUROC 值来比较方法的性能.

表 4 显示了 SLTR 几乎达到了最佳 AUROC 值，并且其时间成本优于所有基线，方法 SLTR 比其他方法快几个数量级. 对于其他方法，只有 Remurs 的 AUROC 值与 SLTR 相当. 在第 1 对标签中，SLTR 的 AUROC 值稍小，但其时间成本比 Remurs 要少得多，同时 SLTR 的 AUROC 值比 SURF 和其余 2 个线性模型要好得多. 在其他 2 种情况下，SLTR 具有更准确的估计和更少的时间成本.

文献 [29] 强调了机器学习方法的可解释性. 本文在图 3 可视化了估计权重，热图显示了 SLTR 的可解释性. 例如，在“ApplyEyeMakeup”标签的视频中，模型的注意力(即高估计权重)主要集中在眼睛上(图 3 中的第 1 行). Remurs 显示了类似的解释，而 SLTR 的估计则更加稀疏. SURF 估计值全为 0，这解释了其糟糕的 AUROC 值. 线性方法只关注一些无法帮助解释的点. 此分类任务的实验表明 SLTR 能够以更少的时间成本获得准确且可解释的解决方案.

比较 SLTR 和 Remurs 的交叉验证时间，结果列于表 5 中. 表 5 表明 SLTR 在交叉验证中比 Remurs 具有更显著的时间优势，因为它比 Remurs 少 2 个数量级.

表 5 的实验结果表明，SLTR 能够在几乎达到最佳 AUROC 值的同时，具有较低的时间成本. 这对于信息系统中的实时决策支持至关重要. 例如，在智能视频监控系统中，捕获到视频的张量数据之后，

Table 4 Time Cost and AUROC Value on UCF101 Dataset

表 4 UCF101 数据集上的时间成本和 AUROC 值

标签对	SLTR（本文）		Remurs		SURF		Lasso		ENet	
	AUROC	运行时间/s	AUROC	运行时间/s	AUROC	运行时间/s	AUROC	运行时间/s	AUROC	运行时间/s
ApplyEyeMakeup 与 ApplyLipstick	0.931 953	0.002 407	**0.945 266**	47.853 97	0.640 53	0.037 561	0.880 06	0.435 661	0.887 556	0.423 819
BaseballPitch 与 Basketball	**0.995 074**	0.000 889	0.995 074	57.556 61	0.786 95	0.504 165	0.964 194	0.147 045	0.965 473	0.603 081
BodyWeightSquats 与 Bowling	**0.977 56**	0.001 211	0.946 704	64.628 6	0.322 58	0.067 25	0.919 753	0.557 569	0.930 556	0.480 47

注：粗体数字表示最佳结果， 下划线值表示次佳结果.

注：选择 4 个示例视频中的 1 帧并显示估计权重的热图.

Fig. 3 Examples of estimated weights

图 3 估计权重的示例

Table 5 Detailed Time on UCF101 Video Data in Cross Validation

表 5 UCF101 视频数据在交叉验证中的详细时间 s

标签对	SLTR（本文）	Remurs
ApplyEyeMakeup 与 ApplyLipstick	6.889 112 525	692.231 181
Archery 与 BabyCrawling	7.028 544 625	625.536 704
BalanceBeam 与 BandMarching	7.279 161 013	633.608 475
BaseballPitch 与 Basketball	6.865 089 138	678.267 903
BasketballDunk 与 BenchPress	6.952 039 188	708.656 48
Biking 与 Billiards	6.930 801 7	698.477 735
BlowDryHair 与 BlowingCandles	6.749 710 2	839.154 294
BodyWeightSquats 与 Bowling	6.638 173 225	701.628 673
BoxingPunchingBag 与 BoxingSpeedBag	7.354 547 413	680.790 876
BreastStroke 与 BrushingTeeth	6.569 058 275	698.097 274

SLTR 快速完成对数据的分析，从而实时检测异常活动，为安全人员提供快速决策支持，减少响应时间，提高系统的效能.

6.3 实验总结

总的来说，本文在模拟数据和 UCF101 视频数据集上，将 SLTR 与 4 种最先进的基线方法进行了预测精度和时间成本上的比较. 结果为：

1) 模拟数据集的结果表明，SLTR 在大多数情况下获得了最佳解决方案，同时花费更少的时间. 在 SLTR 和基线 Remurs 交叉验证的总时间的比较上，SLTR 比 Remurs 少 1~2 个数量级.

2) 在动作识别视频数据集实验中，SLTR 几乎达到了最佳 AUROC 值，并且其时间成本优于所有基线，在交叉验证中比 Remurs 具有显著的时间优势.

3) SLTR 在模拟数据集和 UCF101 视频数据集上的出色表现意味着在信息系统中，使用 SLTR 可以更准确地预测与数据相关的事件、趋势或用户行为，同时在时间成本上的明显优势也意味着可以更快速地响应数据变化，提供实时决策支持. 这对于个性化推荐、异常检测等领域的决策支持非常重要，为信息系统提供了更高效、更准确的数据分析和决策支持.

7 总 结

本文提出了一种快速且可扩展的张量回归方法 SLTR，SLTR 直接对问题施加结构约束，同时还提出了 2 层并行解决方案来有效地求解模型. 本文的工作，即快速分析张量数据的关系，为各类信息系统研发下游服务提供了可靠方法. 例如，在医疗数据信息系统中，该模型可以应用于高维功能性磁共振成像数据，以分析大脑活动与疾病症状之间的关系，辅助医疗人员诊断疾病.

未来的工作将探索 SLTR 在数据维度表示特殊关系数据上的变化. 例如，视频数据的时间维度应该具有时间依赖性，在这种关系上添加额外的约束将提高 SLTR 在特定视频分析应用程序中的性能. 此外将在更一般的情况下，把模型 SLTR 扩展到有关张量对向量或张量对张量的回归任务上.

作者贡献声明：王贝伦提出了论文的核心思想，设计了算法，并撰写了论文的部分内容；张嘉琦参与了论文的构思和算法的设计，对数据集进行了实验，撰写了论文的部分内容；蔡英豪参与了算法实现和对数据集的实验，撰写了论文的部分内容；王兆阳参与了算法实现和实验，撰写了部分内容；谈笑撰写了论文的部分内容，校对了论文的最终版本；沈典指导了整个研究过程，提出了修改意见，审阅了论文的最终版本.

参 考 文 献

[1] Zhu Dajiang, Zhang Tuo, Jiang Xi, et al. Fusing DTI and fMRI data: A survey of methods and applications[J]. NeuroImage, 2014, 102: 184−191

[2] Noroozi A, Rezghi M. A tensor-based framework for rs-fMRI classification and functional connectivity construction[J]. Frontiers in Neuroinformatics, 2020, 14: 581897

[3] Wu X, Lai J. Tensor-based projection using ridge regression and its application to action classification[J]. IET Image Processing, 2010, 4(6): 486−493

[4] Lui Y M. A least squares regression framework on manifolds and its application to gesture recognition[C]//Proc of 2012 IEEE Computer Society Conf on Computer Vision and Pattern Recognition Workshops. Piscataway, NJ: IEEE, 2012: 13−18

[5] Yang Yinchong, Krompass D, Tresp V. Tensor-train recurrent neural networks for video classification[C]//Proc of the 34th Int Conf on Machine Learning. New York: PMLR, 2017: 3891−3900

[6] Sharma L, Gera A. A survey of recommendation system: Research challenges[J]. International Journal of Engineering Trends and Technology, 2013, 4(5): 1989−1992

[7] Bhargava P, Phan T, Zhou Jiayu, et al. Who, what, when, and where: Multi-dimensional collaborative recommendations using tensor factorization on sparse user-generated data[C]//Proc of the 24th Int Conf on World Wide Web. New York: ACM, 2015: 130−140

[8] Mitchell T M, Shinkareva S V, Carlson A, et al. Predicting human brain activity associated with the meanings of nouns[J]. Science, 2008, 320(5880): 1191−1195

[9] Soomro K, Zamir A R, Shah M. UCF101: A dataset of 101 human actions classes from videos in the wild[J]. arXiv preprint, arXiv: 1212.0402, 2012

[10] Huang Qingqiu, Xiong Yu, Rao Anyi, et al. MovieNet: A holistic dataset for movie understanding[C]//Proc of the 16th European Conf. Berlin: Springer, 2020: 709−727

[11] Zhou Hua, Li Lexin, Zhu Hongtu. Tensor regression with applications in neuroimaging data analysis[J]. Journal of the American Statistical Association, 2013, 108(502): 540−552

[12] He Lifang, Chen Kun, Xu Wanwan, et al. Boosted sparse and low-rank tensor regression[J]. Advances in Neural Information Processing Systems, 2018, 31[2023-07-30].https://proceedings.neurips.cc/paper/2018/hash/8d34201a5b85900908db6cae92723617-Abstract.html

[13] Li Na, Stefan K, Carmeliza N. Some convergence results on the regularized alternating least-squares method for tensor decomposition[J]. Linear Algebra and Its Applications, 2013, 438(2): 796−812

[14] Cichocki A, Lee N, Oseledets I, et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions[J]. Foundations and Trends® in Machine Learning, 2016, 9(4/5): 249−429

[15] Song Xiaonan, Lu Haiping. Multi-linear regression for embedded feature selection with application to fMRI analysis[C/OL]//Proc of the 31st AAAI Conf on Artificial Intelligence. 2017[2023-06-30].https://ojs.aaai.org/index.php/AAAI/article/view/10871

[16] Li Wenwen, Lou Jian, Zhou Shuo, et al. Sturm: Sparse tubal-regularized multilinear regression for fMRI[C]//Proc of 10th Int Workshop on Machine Learning in Medical Imaging. Berlin: Springer, 2019: 256−264

[17] Yang E, Lozano A, Ravikumar P. Elementary estimators for high-dimensional linear regression[C]//Proc of the 31st Int Conf on Machine Learning. New York: PMLR, 2014: 388−396

[18] Tucker L R. Some mathematical notes on three-mode factor analysis[J]. Psychometrika, 1966, 31(3): 279−311

[19] Rabanser S, Shchur O, Günnemann S. Introduction to tensor decompositions and their applications in machine learning[J]. arXiv preprint, arXiv: 1711.10781, 2017

[20] Hillar C J, Lim L H. Most tensor problems are NP-hard[J]. Journal of the ACM, 2013, 60(6): 1−39

[21] Tomioka R, Hayashi K, Kashima H. Estimation of low-rank tensors via convex optimization[J]. arXiv preprint, arXiv: 1010.0789, 2010

[22] Liu Ji, Musialski P, Wonka P, et al. Tensor completion for estimating missing values in visual data[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 35(1): 208−220

[23] Boyd S, Parikh N, Chu E, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers[J]. Foundations and Trends® in Machine Learning, 2011, 3(1): 1−122

[24] Combettes P L, Pesquet J C. Proximal splitting methods in signal processing[J]. Fixed-point Algorithms for Inverse Problems in Science and Engineering, 2011, 49: 185−212

[25] Negahban S N, Ravikumar P, Wainwright M J, et al. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers[J/OL]. 2012[2023-07-31].https://projecteuclid.org/journals/statistical-science/volume-27/issue-4/A-Unified-Framework-for-High-Dimensional-Analysis-of-M-Estimators/10.1214/12-STS400.full

[26] Guo Weiwei, Kotsia I, Patras I. Tensor learning for regression[J]. IEEE Transactions on Image Processing, 2011, 21(2): 816−827

[27] Zou Hui, Hastie T. Regularization and variable selection via the elastic net[J]. Journal of the Royal Statistical Society Series B: Statistical Methodology, 2005, 67(2): 301−320

[28] Panagakis Y, Kossaifi J, Chrysos G G, et al. Tensor methods in computer vision and deep learning[J]. Proceedings of the IEEE, 2021, 109(5): 863−890

[29] Chen Kerui, Meng Xiaofeng. Interpretation and understanding in machine learning[J]. Journal of Computer Research and Development, 2020, 57(9): 1971−1986 (in Chinese)

（陈珂锐，孟小峰. 机器学习的可解释性[J]. 计算机研究与发展，2020, 57(9): 1971−1986）

Wang Beilun, born in 1990. PhD, associate professor. His main research interests include large-scale machine learning, graphical model, and multi-task learning.

王贝伦, 1990 年生. 博士，副教授. 主要研究方向为大规模机器学习、图模型、多任务学习.

Zhang Jiaqi, born in 1997. PhD candidate. His main research interests include graphical model and large-scale optimization.

张嘉琦, 1997 年生. 博士研究生. 主要研究方向为图模型、大规模优化.

Cai Yinghao, born in 2002. Undergraduate. His main research interests include graph neural network, knowledge graph, and machine learning.

蔡英豪, 2002 年生. 本科生. 主要研究方向为图神经网络、知识图谱、机器学习.

Wang Zhaoyang, born in 2000. Master candidate. His main research interests include machine learning and software-hardware algorithm acceleration.

王兆阳, 2000 年生. 硕士研究生. 主要研究方向为机器学习、软硬件算法加速.

Tan Xiao, born in 2000. PhD candidate. Her main research interests include graphical model, large-scale machine learning, and meta-learning.

谈　笑, 2000 年生. 博士研究生. 主要研究方向为图模型、大规模机器学习、元学习.

Shen Dian, born in 1988. PhD, associate professor. His main research interests include cloud (edge) computing, network intelligence, and intelligent algorithms and applications.

沈　典, 1988 年生. 博士，副教授. 主要研究方向为云(边缘)计算、网络智能、智能算法及应用.

附录 A. 可重构分组处理流水线开发接口规范.

1 向量和相关操作

定义 A1. M阶张量. 张量数据，也称为多维数组，是一种通用的多维数据形式. 具体来说，M阶张量 $\boldsymbol{\mathcal{A}}\in\mathbb{R}^{p_1\times p_2\times\cdots\times p_M}$ 总共具有M维(或M阶). 具体来说，1阶张量是向量，2阶张量是矩阵.

定义 A2. 向量化和张量化. 向量化表示为$\mathcal{V}(\cdot)$，是将M阶张量的元素堆叠到具有相应大小$\mathbb{R}^{p_1\times p_2\times\cdots\times p_M}$的矩阵中的过程. 张量化表示为$\mathcal{T}_{\mathcal{P}}(\cdot)$，是向量化的逆过程，它根据每阶的大小将向量重新排列为张量，该张量的每阶维数$\mathcal{P}=\{p_1,\cdots,p_M\}$. 值得注意的是，向量化和张量化的遍历元素的顺序应该相同. 以2×3张量为例，假设有

$$\boldsymbol{\mathcal{A}}=\begin{pmatrix}1&2&3\\4&5&6\end{pmatrix}\in\mathbb{R}^{2\times3}. \tag{A1}$$

然后通过先遍历第1维并接着遍历第2维来对其进行向量化，有

$$\boldsymbol{a}=V(\boldsymbol{\mathcal{A}})=(1\quad4\quad2\quad5\quad3\quad6)\in\mathbb{R}^6. \tag{A2}$$

给定$\mathcal{P}=\{2,3\}$，用$\mathcal{T}_{\mathcal{P}}(\boldsymbol{a})$进一步对其进行张量化，需要将1放在位置$(1,1)$，将4放在$(2,1)$位置，同样的方法分别定位其他元素. 因此，这样，张量化和矢量化是一对逆运算，有$\mathcal{T}_{\mathcal{P}}(\mathcal{V}(\boldsymbol{\mathcal{A}}))=\boldsymbol{\mathcal{A}}$.

定义 A3. 张量展开和折叠. 展开，也称为矩阵化，是将张量重新排列成矩阵的过程. 张量展开的结果由张量展开的阶决定. 例如，一个$3\times4\times5$张量可以沿1阶展开为3×20矩阵，或沿2阶展开为12×5矩阵，依此类推. 沿m阶展开张量由$\mathcal{U}_m(\boldsymbol{\mathcal{A}})=\boldsymbol{A}_{(m)}$表示，它将阶数$n$的纤维收集为所得矩阵的列. 使用文献[1]符号，位置$(i_1,i_2,\cdots,i_M)$处的元素映射到位置(i_m,j)处的矩阵元素，其中

$$j=1+\sum_{k=1,k\neq m}^{M}(i_k-1)J_k \text{ with } J_k=\prod_{l=1,l\neq m}^{m-1}p_l. \tag{A3}$$

假设有一个张量$\boldsymbol{\mathcal{A}}\in\mathbb{R}^{2\times2\times2}$，最后一个维度的切片为

$$\boldsymbol{\mathcal{A}}_{::1}=\begin{pmatrix}1&2\\3&4\end{pmatrix},\boldsymbol{\mathcal{A}}_{::2}=\begin{pmatrix}5&6\\7&8\end{pmatrix}. \tag{A4}$$

然后沿每阶展开$\boldsymbol{\mathcal{A}}$得到

$$\boldsymbol{A}_{(1)}=\begin{pmatrix}1&2&5&6\\3&4&7&8\end{pmatrix} \tag{A5}$$

$$\boldsymbol{A}_{(2)}=\begin{pmatrix}1&3&5&7\\2&4&6&8\end{pmatrix}, \tag{A6}$$

$$\boldsymbol{A}_{(3)}=\begin{pmatrix}1&3&2&4\\5&7&6&8\end{pmatrix}. \tag{A7}$$

需要注意的是，不同的论文可能会使用不同的列顺序来进行m阶展开. 然而总的来说，在本文中列的顺序并不重要.

与张量化和向量化类似，折叠操作是展开的逆操作. 令$\mathcal{F}_m(\cdot)$表示沿第m阶折叠展开的张量，有$\mathcal{F}_1\left(\boldsymbol{A}_{(1)}\right)=\boldsymbol{\mathcal{A}}$等. 一般来说，$\mathcal{F}_1(\mathcal{U}_m(\boldsymbol{\mathcal{A}}))=\boldsymbol{\mathcal{A}}$.

定义 A4. 张量内积. 2个大小相同的张量$\boldsymbol{\mathcal{A}}$，$\boldsymbol{\mathcal{B}}\in\mathbb{R}^{p_1\times p_2\times\cdots\times p_M}$的内积是所有张量元素的乘积之和.

$$\langle\boldsymbol{\mathcal{A}},\boldsymbol{\mathcal{B}}\rangle\geqslant\sum_{i_1=1}^{p_1}\cdots\sum_{i_M=1}^{p_M}\boldsymbol{\mathcal{A}}_{i_1\cdots i_M}\boldsymbol{\mathcal{B}}_{i_1\cdots i_M}, \tag{A8}$$

因此有$\langle\boldsymbol{\mathcal{A}},\boldsymbol{\mathcal{A}}\rangle=|(|\boldsymbol{\mathcal{A}}|)|_{\mathrm{F}}^2$.

定义 A5. m阶乘积. 张量$\boldsymbol{\mathcal{A}}\in\mathbb{R}^{p_1\times\cdots\times p_M}$与矩阵的$m$阶乘积$\boldsymbol{Z}\in\mathbb{R}^{J\times p_m}$表示为$\boldsymbol{\mathcal{A}}\times_m\boldsymbol{Z}$，结果的大小为 $p_1\times\cdots\times p_{m-1}\times J\times p_{m+1}\times\cdots\times p_M$. 具体来说，$m$阶元素的乘积是

$$(\boldsymbol{\mathcal{A}}\times_m Z)_{i_1\cdots i_{m-1}ji_{m+1}\cdots i_M}=\sum_{i_m=1}^{p_m}\boldsymbol{\mathcal{A}}_{i_1\cdots i_M}\boldsymbol{Z}_{ji_m}. \tag{A9}$$

定义 A6. CP(candecomp/parafac)分解. 张量可以分解为分量向量的外积之和. 例如，一个3阶张量$\boldsymbol{\mathcal{A}}^{J\times K\times L}$可以分解为

$$\boldsymbol{\mathcal{A}}=\sum_{r=1}^{R}\boldsymbol{a}_r\circ\boldsymbol{b}_r\circ\boldsymbol{c}_r, \tag{A10}$$

其中$R\in\mathbb{Z}_+$，被命名为CP-rank. 此外$\boldsymbol{a}_r\in\mathbb{R}^J$，$\boldsymbol{b}_r\in\mathbb{R}^K$，$\boldsymbol{c}_r\in\mathbb{R}^L$分别表示每阶的分量向量. 具体来说，每个元素可以写为

$$\mathcal{A}_{jkl}=\sum_{r=1}^{R}a_{rj}b_{rk}c_{rl}. \tag{A11}$$

定义 A7. Tucker分解. Tucker分解是奇异值分解(SVD)对高阶数据的直接扩展. 因此，很多论文都将其称为高阶奇异值分解(HOSVD). 通过Tucker分解，张量被分解为核心张量乘以每阶的矩阵. 具体来说，对于3阶张量$\boldsymbol{\mathcal{A}}\in\mathbb{R}^{J\times K\times L}$，有

$$\boldsymbol{\mathcal{A}}=\boldsymbol{\mathcal{C}}\times_1\boldsymbol{X}\times_2\boldsymbol{Y}\times_3\boldsymbol{Z}, \tag{A12}$$

其中$\boldsymbol{\mathcal{C}}\in\mathbb{R}^{D\times E\times F}$是核心张量，$\boldsymbol{X}\in\mathbb{R}^{J\times D}$，$\boldsymbol{Y}\in\mathbb{R}^{K\times E}$，$\boldsymbol{Z}\in\mathbb{R}^{L\times F}$是因子矩阵. 核心张量$\boldsymbol{\mathcal{C}}$中的元素表示不同成分之间的交互.

附录 B. SLTR 的近端算子.

当使用的算法求解SLTR时，需要为总共4个函数定义近端运算符. 具体来说，对于与ℓ_1范数相关的

这 2 个函数，有

$$prox_{4\lambda\|\boldsymbol{W}\|_1}(\boldsymbol{W}) = S_{4\lambda}(\boldsymbol{W}) \tag{B1}$$

和

$$prox_{\|\boldsymbol{W}-\boldsymbol{Z}\|_\infty \leqslant \lambda}(\boldsymbol{W}) = \begin{cases} Z_{ij}, \left|W_{ij}-Z_{ij}\right| \leqslant \lambda, \\ Z_{ij}+\lambda, W_{ij}-Z_{ij} > \lambda, \\ Z_{ij}-\lambda, W_{ij}-Z_{ij} < -\lambda. \end{cases} \tag{B2}$$

这里，$S_\lambda(\cdot)$是一个逐元素软阈值运算符，$[S_\lambda(\boldsymbol{A})]_{ij} = \operatorname{sgn}\left(A_{ij}\right)\max\{\left|A_{ij}\right|-\lambda, 0\}$.

然后，基于奇异值分解（SVD）计算另外 2 个与核范数相关的近端算子. 对于矩阵 $\boldsymbol{A}$，SVD 将其分解为 $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}}$，其中 $\boldsymbol{\Sigma}$ 是对角矩阵，对角线为奇异值，$\boldsymbol{U}$，$\boldsymbol{V}$ 是相应的左右奇异值向量. 基于 SVD，有

$$prox_{4\tau\|\boldsymbol{W}\|_*}(\boldsymbol{W}) = \boldsymbol{U}\boldsymbol{S}_{4\tau}(\boldsymbol{\Sigma})\boldsymbol{V}^{\mathrm{T}} \tag{B3}$$

和

$$prox_{\|\boldsymbol{W}-\boldsymbol{Z}\|_{\mathrm{spec}} \leqslant \tau}(\boldsymbol{W}) = \begin{cases} \boldsymbol{Z}, \sigma_{\max}(\boldsymbol{\Sigma}) \leqslant \tau. \\ \boldsymbol{U}\boldsymbol{S}_\tau(\boldsymbol{\Sigma})\boldsymbol{V}^{\mathrm{T}} + \boldsymbol{Z}, \sigma_{\max}(\boldsymbol{\Sigma}) > \tau. \end{cases} \tag{B4}$$

附录 C. 混合范数的对偶范数.

定理 C1. 假设有混合正则化器 $\mathcal{R}(\boldsymbol{w}) = \sum_\beta \boldsymbol{c}_\beta R_\beta(\boldsymbol{w}_\beta)$，其中 $\sum_\beta \boldsymbol{w}_\beta = \boldsymbol{w}$，$\beta$ 是 $\boldsymbol{I}$ 的元素，$\boldsymbol{c}$ 是包含每个分量范数 $\mathcal{R}_\beta(\cdot)$ 正则化参数的向量，那么它的对偶范数就是 $\mathcal{R}^*(\boldsymbol{w}) = \max_\beta \mathcal{R}_\beta^*(\boldsymbol{w}_\beta)/\boldsymbol{c}_\beta$.

证明. 对偶范数可以由下式导出：

$$\begin{aligned} \mathcal{R}^*(\boldsymbol{w}) = & \sup_{\boldsymbol{u}} \frac{\langle \boldsymbol{w}, \boldsymbol{u}\rangle}{\boldsymbol{u}} = \sup_{\boldsymbol{u}\beta} \frac{\sum_\beta \langle \boldsymbol{w}_\beta, \boldsymbol{u}\rangle}{\sum_\beta \boldsymbol{c}_\beta \mathcal{R}_\beta(\boldsymbol{w}_\beta)} = \\ & \sup_{\boldsymbol{u}\beta} \frac{\sum_\beta \langle \boldsymbol{w}_\beta, \boldsymbol{u}/\boldsymbol{c}_\beta\rangle}{\sum_\beta R_\beta(\boldsymbol{w}_\beta)} \leqslant \\ & \sup_{\boldsymbol{u}\beta} \frac{\sum_\beta \mathcal{R}_\beta^*\left(\boldsymbol{u}/\boldsymbol{c}_\beta\right)\mathcal{R}(\boldsymbol{w}_\beta)}{\sum_\beta R_\beta(\boldsymbol{w}_\beta)} \leqslant \\ & \max_\beta \mathcal{R}_\beta^*(\boldsymbol{u})/\boldsymbol{c}_\beta. \end{aligned} \tag{C1}$$

因此当用 $\boldsymbol{\mathcal{W}} = \sum_{m=1}^M \frac{1}{M}\boldsymbol{W}_{(m)}$ 代替 $\boldsymbol{w}$ 时，定义 $\boldsymbol{c} = (\lambda_1, \cdots, \lambda_M, \tau_1, \cdots, \tau_M)$，并定义

$$\ell(\boldsymbol{\mathcal{W}}) = \|y - \langle \boldsymbol{\mathcal{W}}, \boldsymbol{\mathcal{X}}\rangle\|_2^2 \tag{C2}$$

和

$$\boldsymbol{c}\mathcal{R}(\boldsymbol{\mathcal{W}}) = \inf\left\{\sum_{m=1}^M c_m\|\boldsymbol{W}_{(m)}\|_1 + \sum_{q=M+1}^{2M} c_q\|\boldsymbol{W}_{(q-M)}\|_*\right\}, \tag{C3}$$

有

$$\mathcal{R}^*(\boldsymbol{\mathcal{W}}) = \max\left\{\|\boldsymbol{W}_{(1)}\|_\infty, \cdots, \|\boldsymbol{W}_{(M)}\|_\infty, \|\boldsymbol{W}_{(1)}\|_2, \cdots, \|\boldsymbol{W}_{(M)}\|_2\right\}. \tag{C4}$$

证毕.

附录 D. 张量核范数的可分解性.

定理 D1. 张量核范数 $\|\boldsymbol{\mathcal{W}}\|_* = \frac{1}{M}\sum_{m=1}^M \|\boldsymbol{W}_{(m)}\|_*$ 是可分解的一对子空间 $\left(\mathcal{M}, \overline{\mathcal{M}}^\perp\right)$. 具体来说，

$$\|\boldsymbol{\mathcal{W}} + \boldsymbol{\mathcal{V}}\|_* = \|\boldsymbol{\mathcal{W}}\|_* + \|\boldsymbol{\mathcal{V}}\|_*. \tag{D1}$$

对于所有 $\boldsymbol{\mathcal{W}} \in \mathcal{M}(\boldsymbol{U}, \boldsymbol{V})$ 和 $\boldsymbol{\mathcal{V}} \in \overline{\mathcal{M}}^\perp(\boldsymbol{U}, \boldsymbol{V})$，$\mathcal{M}(\boldsymbol{U}, \boldsymbol{V}) = \{\boldsymbol{\mathcal{W}} \in \mathbb{R}^{p_1 \times p_2 \times \cdots \times p_M} | col(\boldsymbol{W}_{(1)}) \subseteq \boldsymbol{U}_1, \cdots, col(\boldsymbol{W}_{(M)}) \subseteq \boldsymbol{U}_M, row(\boldsymbol{W}_{(1)}) \subseteq \boldsymbol{V}_1, \cdots, row(\boldsymbol{W}_{(M)}) \subseteq \boldsymbol{V}_M\}$ 和 $\overline{\mathcal{M}}^\perp(\boldsymbol{U}, \boldsymbol{V}) = \{\boldsymbol{\mathcal{W}} \in \mathbb{R}^{p_1 \times \cdots \times p_M} | col(\boldsymbol{W}_{(1)}) \subseteq \boldsymbol{U}_1^\perp, \cdots, col(\boldsymbol{W}_{(M)}) \subseteq \boldsymbol{U}_M^\perp, row(\boldsymbol{W}_{(1)}) \subseteq \boldsymbol{V}_1^\perp, \cdots, row(\boldsymbol{W}_{(M)}) \subseteq \boldsymbol{V}_M^\perp\}$. 这里，$row(\cdot)$ 和 $col(\cdot)$ 分别是行空间和列空间. $\boldsymbol{U}_m$ 和 $\boldsymbol{V}_m$ 是一对沿着 m 阶展开的张量 $\boldsymbol{W}_{(m)}$ 的子空间，$\boldsymbol{U}_m \subseteq \mathbb{R}_m^p$ 和 $\boldsymbol{V}_m \subseteq \mathbb{R}^{p_1 \times p_2 \times \cdots \times p_{m-1} \times p_{m+1} \times \cdots \times p_M}$.

证明. 对于任意一对张量 $\boldsymbol{\mathcal{W}} \in \mathcal{M}(\boldsymbol{U}, \boldsymbol{V})$ 和 $\boldsymbol{\mathcal{V}} \in \overline{\mathcal{M}}^\perp(\boldsymbol{U}, \boldsymbol{V})$，它们的 m 阶展开张量 $\boldsymbol{W}_{(m)}$ 和 $\boldsymbol{V}_{(m)}$ 具有正交的行空间和列空间. 因为 $row(\boldsymbol{W}_{(m)}) \subseteq \boldsymbol{V}_m$，$row\left(\boldsymbol{V}_{(m)}\right) \subseteq \boldsymbol{V}_m^\perp$，$col(\boldsymbol{W}_{(m)}) \subseteq \boldsymbol{U}_m$，$col\left(\boldsymbol{V}_{(m)}\right) \subseteq \boldsymbol{U}_m^\perp$ 和 $\boldsymbol{V}_m \perp \boldsymbol{V}_m^\perp$ 和 $\boldsymbol{U}_m \perp \boldsymbol{U}_m^\perp$. 这意味着它们满足矩阵核范数的可分解性，即矩阵核范数 $\|\boldsymbol{W}_{(m)} + \boldsymbol{V}_{(m)}\|_* = \|\boldsymbol{W}_{(m)}\|_* + \|\boldsymbol{V}_{(m)}\|_*$，这样就得到了展开张量的正交性.

证毕.

然后，注意展开操作不会改变张量的元素，因此，有 $(\boldsymbol{\mathcal{W}} + \boldsymbol{\mathcal{V}})_{(m)} = \boldsymbol{W}_{(m)} + \boldsymbol{V}_{(m)}$. 因此，2 个张量之和的张量核范数可以重新表述为

$$\begin{aligned} \|\boldsymbol{\mathcal{W}} + \boldsymbol{\mathcal{V}}\|_* = & \frac{1}{M}\sum_{m=1}^M \|(\boldsymbol{\mathcal{W}} + \boldsymbol{\mathcal{V}})_{(m)}\|_* = \frac{1}{M}\sum_{m=1}^M \|\boldsymbol{W}_{(m)} + \boldsymbol{V}_{(m)}\|_* = \\ & \frac{1}{M}\sum_{m=1}^M \left(\|\boldsymbol{W}_{(m)}\|_* + \|\boldsymbol{V}_{(m)}\|_*\right) = \|\boldsymbol{\mathcal{W}}\|_* + \|\boldsymbol{\mathcal{V}}\|_*. \end{aligned} \tag{D2}$$

因此，通过定理 D1 中正确定义的子空间对，证明了张量核范数的可分解性.

附录 E. 误差界证明.

稀疏性：真实系数 $\boldsymbol{\mathcal{W}}^*$ 完全稀疏，具有 k 个非零元素.

低秩性：真实系数 $\boldsymbol{\mathcal{W}}^*$ 是 R 阶张量，其中 $R =$

$\max\limits_{\mathcal{A}\in\mathbb{R}^{p_1\times p_2\times\cdots\times p_M}}(r_\perp(\mathcal{A}))$ 和 $r_\perp(\mathcal{A})$表示 $\mathcal{A}$的正交秩. 正交秩是满足 $\mathcal{A}=\sum\limits_{r=1}\mathcal{U}_r$ 的最小数, 其中 $\langle\mathcal{U}_{r_1},\mathcal{U}_{r_2}\rangle=0, r_1\neq r_2$ 对于 $1\leqslant r_1\leqslant r_\perp(\mathcal{A}), 1\leqslant r_2\leqslant r_\perp(\mathcal{A})$.

定理 E1. 假设真实系数张量 $\mathcal{W}^*$ 满足条件稀疏性和低秩性. 此外, 假设通过控制参数 λ 和 τ 满足约束来求解 SLTR, 那么, 最优解满足以下误差界:

$$\|\hat{\mathcal{W}}-\mathcal{W}^*\|_{\mathrm{F}}\leqslant 4\sqrt{2}\left(\lambda\sqrt{\prod_{m=1}^{M}p_m}+\tau\sqrt{R}\right). \tag{E1}$$

证明. 在本文方法中, 利用 2 个正则化器, 逐元素的 ℓ_1 范数 $\mathcal{R}_1(\cdot)=\|\cdot\|_1$ 和张量核范数 $\mathcal{R}_*(\cdot)=\|\cdot\|_*$. 因此, 首先令 $\Delta_1=\hat{\mathcal{W}}_{\|\cdot\|_1}-\mathcal{W}^*$ 和 $\Delta_2=\hat{\mathcal{W}}_{\|\cdot\|_*}-\mathcal{W}^*$, 其中 $\hat{\mathcal{W}}_{\|\cdot\|_1}$, $\hat{\mathcal{W}}_{\|\cdot\|_*}$ 分别是最小化元素级 ℓ_1 范数和张量核范数的解. 此外, 设 $\Delta=\Delta_1+\Delta_2$, 调整参数 $\beta=\lambda$, $\beta_2=\tau$, $\mathcal{I}=\{1,2\}$. 那么 Frobenius 误差 $\|\Delta\|_{\mathrm{F}}$ 有界如下:

$$\begin{aligned}\|\Delta\|_{\mathrm{F}}^2=\langle\Delta,\Delta\rangle=\sum_{\alpha\in\mathcal{I}}\langle\Delta,\Delta_\alpha\rangle\leqslant\sum_{\alpha\in\mathcal{I}}|\langle\Delta,\Delta_\alpha\rangle|\leqslant 2\sum_{\alpha\in\mathcal{I}}\beta_\alpha\mathcal{R}_\alpha(\Delta_\alpha)\leqslant\\ 2\sum_{\alpha\in\mathcal{I}}\beta_\alpha\left\{\mathcal{R}_\alpha\left[\prod_{\overline{\mathcal{M}}_\alpha}(\Delta_\alpha)\right]+\beta_\alpha\mathcal{R}_\alpha\left[\prod_{\overline{\mathcal{M}}_\alpha}(\Delta_\alpha)\right]\right\}\leqslant\\ 4\sum_{\alpha\in\mathcal{I}}\beta_\alpha\mathcal{R}_\alpha\left[\prod_{\overline{\mathcal{M}}_\alpha}(\Delta_\alpha)\right].\end{aligned} \tag{E2}$$

在式(E2) 中, 第 2 个不等式来自Hölder不等式的应用, 即对每个 $\alpha\in\mathcal{I}$, $|\langle\langle\Delta,\Delta_\alpha\rangle\rangle|\leqslant\mathcal{R}_\alpha^*(\Delta)\mathcal{R}_\alpha(\Delta_\alpha)\leqslant 2\beta_\alpha\mathcal{R}_\alpha(\Delta_\alpha)$, 其中 $\langle\langle\boldsymbol{A},\boldsymbol{B}\rangle\rangle=\sum\limits_i\sum\limits_j A_{ij}B_{ij}$ 表示迹内积, 这里用 $\Psi\left(\overline{\mathcal{M}}_\alpha\right)$ 来表示空间 $\overline{\mathcal{M}}_\alpha$ 的兼容性为:

$$\Psi\left(\overline{\mathcal{M}}_\alpha\right)=\sup_{\mathcal{U}\in\overline{\mathcal{M}}_\alpha\backslash\mathbf{0}}\frac{\mathcal{R}_\alpha(\boldsymbol{U})}{\|\boldsymbol{U}\|_{\mathrm{F}}}. \tag{E3}$$

有式(E3), 式 (E2) 被重写为

$$\|\Delta\|_{\mathrm{F}}^2\leqslant 4\sum_{\alpha\in\mathcal{I}}\beta_\alpha\Psi\left(\overline{\mathcal{M}}_\alpha\right)\left\|\prod_{\overline{\mathcal{M}}_\alpha}(\Delta_\alpha)\right\|_{\mathrm{F}}. \tag{E4}$$

现在, 考虑 $\left\|\prod\limits_{\overline{\mathcal{M}}_\alpha}(\Delta_\alpha)\right\|_{\mathrm{F}}$. 请注意, 映射算子是根据 Frobenius 范数定义的, 这使得它对于所有 α 来说成本并不高昂, 使得 $\left\|\prod\limits_{\overline{\mathcal{M}}_\alpha}(\Delta_\alpha)\right\|_{\mathrm{F}}$. 此外, 类似于式 (E2) 和式 (E4), 有 $\|\Delta_\alpha\|_{\mathrm{F}}^2\leqslant 4\beta_\alpha\Psi\left(\overline{\mathcal{M}}_\alpha\right)\left\|\prod\limits_{\overline{\mathcal{M}}_\alpha}(\Delta_\alpha)\right\|_{\mathrm{F}}$.

因此得到

$$\begin{aligned}\left(\sum_{\alpha\in\mathcal{I}}\left\|\prod_{\overline{\mathcal{M}}_\alpha}(\Delta_\alpha)\right\|_{\mathrm{F}}\right)^2\leqslant\left(\sum_{\alpha\in\mathcal{I}}\|\Delta_\alpha\|_{\mathrm{F}}\right)^2\leqslant|\mathcal{I}|\sum_{\alpha\in\mathcal{I}}\|\Delta_\alpha\|_{\mathrm{F}}^2\leqslant\\ 4|\mathcal{I}|\beta_\alpha\Psi\left(\overline{\mathcal{M}}_\alpha\right)\left\|\prod_{\overline{\mathcal{M}}_\alpha}(\Delta_\alpha)\right\|_{\mathrm{F}},\end{aligned} \tag{E5}$$

因此

$$\sum_{\alpha\in\mathcal{I}}\left\|\prod_{\overline{\mathcal{M}}_\alpha}(\Delta_\alpha)\right\|_{\mathrm{F}}\leqslant 4|\mathcal{I}|\beta_\alpha\Psi\left(\overline{\mathcal{M}}_\alpha\right). \tag{E6}$$

将式 (E6) 插回到式 (E4) 中, Frobenius 范数误差界导出为

$$\|\Delta\|_{\mathrm{F}}^2\leqslant 16|\mathcal{I}|\left(\sum_{\alpha\in\mathcal{I}}\beta_\alpha\Psi\left(\overline{\mathcal{M}}_\alpha\right)\right)^2, \tag{E7}$$

因此

$$\|\Delta\|_{\mathrm{F}}\leqslant 4\sqrt{|\mathcal{I}|}\sum_{\alpha\in\mathcal{I}}\beta_\alpha\Psi\left(\overline{\mathcal{M}}_\alpha\right). \tag{E8}$$

回顾方法的公式, 有 $\mathcal{I}=\{1,2\}$ 和 $\mathcal{R}_1(\Delta_1)=\|\Delta_1\|_1$, $\mathcal{R}_2(\Delta_2)=\|\Delta_2\|_*$. 对于 ℓ_1 范数 $\mathcal{R}_1(\Delta_1)=\|\Delta_1\|_1$, 文献 [1] 已经证明了它的可分解性. 因此, 很容易得到

$$\Psi\left(\overline{\mathcal{M}}_1\right)=\sup_{\mathcal{U}\in\overline{\mathcal{M}}_\alpha\backslash\mathbf{0}}\frac{\|\mathcal{U}\|_1}{\|\mathcal{U}\|_{\mathrm{F}}}\leqslant\sqrt{\prod_{m=1}^{M}p_m}. \tag{E9}$$

至于张量核范数 $\mathcal{R}_2(\Delta_2)=\|\Delta_2\|_*$, 已经在本文证明了它的可分解性, 并且之前的文献 [2−3] 已经证明了它的上限. 文献 [3] 提出了由 Frobenius 范数给出的张量核范数的最紧上限, 即 $\|\mathcal{A}\|_*\leqslant\sqrt{R}\|\mathcal{A}\|_{\mathrm{F}}$. 这里 $R=\max\limits_{\mathcal{B}\in\mathbb{R}^{p_1\times p_M}}\{\mathrm{rank}_\perp(\mathcal{B})\}$, 其中 $\mathrm{rank}_\perp(\mathcal{B})$ 表示任意非零张量 $\mathcal{B}$ 的正交秩. 因此, 有

$$\Psi\left(\overline{\mathcal{M}}_2\right)=\sup_{\mathcal{U}\in\overline{\mathcal{M}}_\alpha\backslash\mathbf{0}}\frac{\|\mathcal{U}\|_*}{\|\mathcal{U}\|_{\mathrm{F}}}\leqslant\sqrt{R}. \tag{E10}$$

将 $\Psi\left(\overline{\mathcal{M}}_1\right)$, $\Psi\left(\overline{\mathcal{M}}_2\right)$ 与式 (E8) 结合, 最终得到以下误差上限:

$$\|\Delta\|_{\mathrm{F}}\leqslant 4\sqrt{2}\left(\lambda\sqrt{\prod_{m=1}^{M}p_m}+\tau\sqrt{R}\right). \tag{E11}$$

证毕.

推论 E1. 如果系数是一个 3 阶张量 $\mathcal{W}\in\mathbb{R}^{p_1\times p_2\times p_3}$ 且条件稀疏性和低秩性成立, SLTR 的最优解满足以下误差界:

$$\|\hat{\mathcal{W}}-\mathcal{W}^*\|_{\mathrm{F}}\leqslant 4\sqrt{2}\left(\lambda\sqrt{\prod_{m=1}^{M}p_m}+\tau R'\right), \tag{E12}$$

其中

$$R'=\max\left\{\sqrt{r_1\min\{r_2,r_3\}},\sqrt{r_2\min\{r_1,r_3\}},\sqrt{r_3\min\{r_1,r_2\}}\right\},\tag{E13}$$

其中$r_m=\mathrm{rank}(\boldsymbol{W}_{(m)})$表示展开张量的秩.

证明. 对于3阶张量, 在给定其展开矩阵的核范数的情况下, 文献[4]提出了张量核范数的上限. 具体来说, 令$r_m=\mathrm{rank}(\boldsymbol{W}_{(m)}),m=1,2,3$, 则

$$\begin{aligned}\|\mathcal{A}\|_*\leqslant&\frac{1}{3}\left(\sqrt{\min\{r_2,r_3\}}\|\mathcal{A}_{(1)}\|_*+\sqrt{\min\{r_1,r_3\}}\|\mathcal{A}_{(3)}\|_*+\right.\\&\left.\sqrt{\min\{r_1,r_2\}}\|\mathcal{A}_{(3)}\|_*\right)\leqslant\frac{1}{3}\left(\sqrt{r_1\min\{r_2,r_3\}}\|\mathcal{A}\|_{\mathrm{F}}+\right.\\&\left.\sqrt{r_2\min\{r_1,r_3\}}\|\mathcal{A}\|_{\mathrm{F}}+\sqrt{r_3\min\{r_1,r_2\}}\|\mathcal{A}\|_{\mathrm{F}}\right)\leqslant\\&\max\left\{\sqrt{r_1\min\{r_2,r_3\}},\sqrt{r_2\min\{r_1,r_3\}},\right.\\&\left.\sqrt{r_3\min\{r_1,r_2\}}\right\}\|\mathcal{A}\|_{\mathrm{F}}.\end{aligned}\tag{E14}$$

因此$\Psi\left(\overline{\mathcal{M}}_2\right)$即为

$$\begin{aligned}\Psi\left(\overline{\mathcal{M}}_2\right)=\sup_{\mathcal{U}\in\overline{\mathcal{M}}_\alpha\setminus\mathbf{0}}\frac{\|\mathcal{U}\|_*}{\|\mathcal{U}\|_{\mathrm{F}}}\leqslant\max\Big\{&\sqrt{r_1\min\{r_2,r_3\}},\\&\sqrt{r_2\min\{r_1,r_3\}},\sqrt{r_3\min\{r_1,r_2\}}\Big\}.\end{aligned}\tag{E15}$$

用式(E15)中的$\Psi\left(\overline{\mathcal{M}}_2\right)$替换掉式(E8)中的$\Psi\left(\overline{\mathcal{M}}_2\right)$, 有最终的误差界:

$$\begin{aligned}\|\mathcal{A}\|_{\mathrm{F}}\leqslant4\sqrt{2}\Big(&\lambda\sqrt{\prod_{m=1}^{M}p_m}+\tau\max\Big\{\sqrt{r_1\min\{r_2,r_3\}},\\&\sqrt{r_2\min\{r_1,r_3\}},\sqrt{r_3\min\{r_1,r_2\}}\Big\}\Big).\end{aligned}\tag{E16}$$

证毕.

附录F. 更多实验的内容.

1 实验设置

数据集: 将SLTR与模拟数据集和真实世界fMRI数据集的基线进行比较. 具体来说, 模拟数据集是通过3个步骤生成的:

1) 生成$\boldsymbol{\mathcal{W}}\in\mathbb{R}^{p_1\times p_2\times\cdots\times p_M}$和$\boldsymbol{\mathcal{X}}\in\mathbb{R}^{N\times p_1\times p_2\times\cdots\times p_M}$, 每个元素均取自正态分布$\mathcal{N}(0,1)$.

2) 随机设置$\boldsymbol{\mathcal{W}}$的s%的元素为0.

3) 计算相对于$y_i=\langle\boldsymbol{\mathcal{W}},\boldsymbol{\mathcal{X}}_i\rangle+\alpha\varepsilon_i$的响应$\boldsymbol{y}\in\mathbb{R}^N$. 这里$\alpha$控制噪声的比率, 噪声$\varepsilon_i$是根据正态分布$\mathcal{N}(0,1)$生成的.

为了在各种模拟数据集上测试SLTR和其他方法, 根据表F1中显示的设置生成多个模拟数据集. 除了模拟数据集之外, 本文还在UCF101数据集上进一步实验SLTR.

Table F1 Parameter Settings for Data Simulation

表F1 模拟数据的参数设置

数据	数据大小	样本数N	s/%	α
3阶数据	10×10×5	40	80	0.1
	15×15×5	90		
	20×20×5	160		
	25×25×5	250		
	30×30×5	360		
	35×35×5	400		
	40×40×5	640		
4阶数据	10×10×10×5	400		
	15×15×10×5	900		
	20×20×10×5	1 600		
	25×25×10×5	2 500		
	30×30×10×5	3 600		

其他设置: 在实验中, 所有方法的所有调整参数都是通过5倍交叉验证程序选择的, 该程序以验证数据集的平均MSE作为选择标准. 通过交叉验证选择的调整参数范围如表F2所示. 至于实验环境, 所有实验均在具有2个Intel Xeon Silver 4216 CPU和256 GB RAM的Linux服务器上实现. 而且, 出于比较的公平性, 使用MATLAB来实现所有方法. 将所有方法的最大迭代次数设置为1 000, 并让它们在$\frac{\left\|\boldsymbol{\mathcal{W}}^{t+1}-\boldsymbol{\mathcal{W}}^{t}\right\|_{\mathrm{F}}}{\left\|\boldsymbol{\mathcal{W}}^{t}\right\|_{\mathrm{F}}}\leqslant10^{-4}$时终止. 将每个实验运行10次, 并报告这10次实验的平均性能.

2 模拟数据上的实验

表F3显示了模拟数据集上方法的MSE值. 结果表明, 在大多数情况下, SLTR获得最佳解决方案, 而在其他情况下, SLTR计算的估计仅比最佳解决方案差一点. 因此, SLTR能够花费更少的时间获得不比通过基线方法获得的最佳解决方案差的解决方案.

此外, 在高维设置中测试SLTR, 让数据的形状为$20\times20\times5$, N从50~400变化, 稀疏度为$s=80\%$, 噪声因子设置为$\alpha=0.1$. 表F4中显示的MSE值表明SLTR在几乎所有条件下都获得了最佳解决方案. 请注意, 对于$N=150$, 即使SLTR的MSE为0.929 5, 这比Remurs的MSE (0.919 0)差一点, 但它比其他方法的结果要好得多: SURF的MSE为0.995 3和ENet的MSE为1.6502.

3 在行为识别视频数据集上的参数调整

在UCF101数据集上, 进行了5倍交叉验证来调整方法超参数. 在SLTR中, 在10^{-2}~1选择τ, 在10^{-4}~

Table F2 Range of Tuning Parameters of All Methods

表 F2 所有方法的参数调整范围

方法	参数	交叉验证范围
SLTR	ε	{0, 0.1, 0.2, 0.3, 0.4, 0.5}
	λ	$0, 10^{-4}, 5\times10^{-4}, 10^{-3}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}$
	τ	$0, 10^{-2}, 5\times10^{-2}, 10^{-1}, 5\times10^{-1}, 10^{0}, 5\times10^{0}, 10^{1}, 5\times10^{1}$
Remurs	α	$10^{-2}, 5\times10^{-2}, 10^{-1}, 5\times10^{-1}, 10^{0}, 5\times10^{0}, 10^{1}, 5\times10^{1}$
	β	$10^{-2}, 5\times10^{-2}, 10^{-1}, 5\times10^{-1}, 10^{0}, 5\times10^{0}, 10^{1}, 5\times10^{1}$
Lasso	λ的数量	100(稀疏度控制参数λ的数量)
ENet	λ的数量	100(稀疏度控制参数λ的数量)
	α	0.5(在ℓ_1和ℓ_2范数之间进行权衡)
SURF	α	$5\times10^{-4}, 10^{-4}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}, 10^{-1}$
	ε	$5\times10^{-4}, 10^{-4}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}, 10^{-1}$
	R	1, 2, 3, 4, 5, 6, 7, 8, 9, 10
GLTRM	R	1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Table F3 MSE of Different Methods on Simulated Dataset with Different Sizes of Data

表 F3 不同方法在具有不同数据大小的模拟数据集的 MSE

数据大小	SLTR(本文)	Remurs	SURF	Lasso	ENet
10×10×5	0.853 2	0.853 8	0.847 2	1.799 3	1.799 0
15×15×5	0.989 2	0.986 7	0.999 4	2.315 1	2.313 2
20×20×5	0.938 3	0.937 8	1.004 9	2.546 9	2.519 3
25×25×5	**0.927 5**	0.939 8	0.939 1	2.014 9	2.014 9
30×30×5	**0.918 6**	0.919 0	0.928 9	1.938 1	1.937 7
35×35×5	**0.933 6**	0.937 0	0.952 7	2.014 7	2.014 7
40×40×5	0.907 3	0.907 2	1.000 6	2.105 9	2.106 5
10×10×10×5	**0.918 3**	0.933 9		1.979 9	1.998 5
15×15×10×5	**0.912 7**	0.918 6		2.169 0	2.169 0
20×20×10×5	**0.915 0**	0.917 7		2.138 8	2.137 3
25×25×10×5	**0.907 1**	0.910 1		1.969 6	1.969 6
30×30×10×5	**0.911 0**	0.912 3		1.975 4	1.974 5

注：粗体数字表示最佳 MSE 值，下划线值表示次佳结果. 需要注意的是 SURF 无法适用于 2D 和 4D 的数据，GLTRM 只适用于 2D 数据.

5×10^{-3}选择λ, 在 0.1~0.4选择ε. Remurs 中，选择5×10^{-3}~5的 2 个参数α和β. 在 SURF 中，在5×10^{-4}~0.1中选择α, 在5×10^{-4}~0.1中选择ε, 在{1,2}中选择R. 随机选择 80% 的样本作为训练集，其余的样本作为测试样本.

Table F4 MSE of Every Method in High-Dimensional Settings

表 F4 高维设置中每种方法的 MSE

N	SLTR	Remurs	SURF	Lasso	ENet
50	**1.612 3**	1.619 8	1.643 9	4.508 3	4.575 9
100	**1.079 8**	1.094 6	1.710 1	1.643 3	1.643 3
150	0.929 5	0.919 0	0.995 3	1.677 7	1.650 2
200	**0.835 1**	0.846 9	0.837 6	1.907 2	0.837 6
250	**0.713 0**	0.726 7	0.719 9	1.375 7	1.370 8
300	**0.728 2**	0.732 5	0.752 4	1.731 6	1.693 8
350	**0.627 5**	0.627 5	0.637 9	1.320 7	1.280 4
400	**0.595 4**	0.596 9	0.597 5	1.148 7	1.145 0

注：生成模拟数据集，数据形状为$20\times20\times5$，稀疏度$s=80\%$，噪声系数$\alpha=0.1$. 样本数量为 50~400 不等. GLTRM 不适用于 3D 数据. 粗体数字表示最佳 MSE 值，下划线值表示次佳结果.

附录参考文献

[1] Kolda T G, Bader B W. Tensor decompositions and applications[J]. SIAM Review, 2009, 51(3): 455−500

[2] Negahban S N, Ravikumar P, Wainwright M J, et al. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers[J] Statistical Science, 2012, 27 (4): 538−557

[3] Friedland S, Lim L H. Nuclear norm of higher-order tensors[J]. Mathematics of Computation, 2018, 87: 1255−1281

[4] Xu Kong, Li Jicheng, Wang Xiaolong. New estimations on the upper bounds for the nuclear norm of a tensor[J]. Journal of Inequalities and Applications, 2018(1): 1−17