

基于单向流模型的自适应张量链式学习算法

马宝泽^{1,2}, 李国军^{1,2}, 邢隆^{1,2}, 叶昌荣^{1,2}

(1. 重庆邮电大学光电工程学院, 重庆 400065; 2. 重庆邮电大学超视距可信信息传输研究所, 重庆 400065)

摘要: 针对单向流模型中高阶张量在线分解问题, 研究了一种自适应张量链式(TT)学习算法。首先, 推导出单向流增量仅改变时序TT核的维度; 然后, 引入遗忘因子和正则项构造指数权重最小二乘目标函数; 最后, 利用块坐标下降学习策略分别估计时序和非时序TT核。对所提算法在增量大小、TT秩、噪声和时变强度等方面分别进行了验证, 结果表明, 所提算法的平均相对误差和运算时间均小于对比算法, 并在视频自适应分析中表现出优于对比算法的张量切片重构能力。

关键词: 自适应学习算法; 张量链式分解; 单向流模型; 泛在数据流

中图分类号: TN911.6

文献标志码: A

DOI: 10.11959/j.issn.1000-436x.2023154

Adaptive tensor train learning algorithm based on single-aspect streaming model

MA Baoze^{1,2}, LI Guojun^{1,2}, XING Long^{1,2}, YE Changrong^{1,2}

1. School of Optoelectronic Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

2. Lab of Beyond LoS Reliable Information Transmission, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Abstract: An adaptive tensor train (TT) learning algorithm for the online decomposition problem of high-order tensors in single-aspect streaming model was investigated. Firstly, it was deduced that single-aspect streaming increment only changes the dimension of temporal TT core. Secondly, the forgetting factor and regularization item were introduced to construct the objective function of exponentially weighted least-squares. Finally, the block-coordinate descent learning strategy was used to estimate the temporal and non-temporal TT core tensors respectively. Simulation results demonstrate that the proposed algorithm is validated in terms of increment size, TT-rank, noise and time-varying intensity, the average relative error and operation time are smaller than that of the comparison algorithms. The tensor slice reconstruction ability is superior than that of the comparison algorithms in the video adaptive analysis.

Keywords: adaptive learning algorithm, tensor train decomposition, single-aspect streaming model, ubiquitous data stream

0 引言

随着传感器的广泛应用和物联网的迅猛发

展, 泛在的实时信号和数据不断产生, 并催生了大数据智能、信息流处理、泛在感知等新兴研究热点, 推动新生业态的数字、信息技术飞速发展。

收稿日期: 2023-05-11; 修回日期: 2023-07-16

通信作者: 邢隆, xinglong@cqupt.edu.cn

基金项目: 国家重点研发计划基金资助项目(No.2019YFC1511300); 国家自然科学基金资助项目(No.62201113, No.U22A2006); 重庆市重点研发计划基金资助项目(No.cstc2021ycjh-bgzxm0072); 重庆市教委科学技术研究基金资助项目(No.KJQN202300625)

Foundation Items: The National Key Research and Development Program of China (No.2019YFC1511300), The National Natural Science Foundation of China (No.62201113, No.U22A2006), Chongqing Key Research and Development Project (No.cstc2021ycjh-bgzxm0072), The Science and Technology Research Program of Chongqing Municipal Education Commission (No.KJQN202300625)

在数字经济赋能各行各业的大背景下，人们对高维实时/在线数据流自适应处理的需求处于持续增长状态^[1]。特别是，大量的应用程序在某一维度上随着时间的推移将产生庞大的在线信号，及时有效地分析基于单向流模型的海量实时数据信息，实现信号在线自适应处理成为当前研究的热点^[2]。然而，信号流的数据规模越来越大、传输速度越来越快、潜在成分越来越复杂，这些固有的问题都给实时信号流处理带来了挑战。此外，信息技术的飞速发展使数据形式逐渐趋向高维化，通常需用张量的形式表征^[3]。因此，利用张量分析的相关技术处理实时的信号流数据成为一种潜在研究手段。

张量分解技术已经得到了广泛的应用，成为解决高维信息分析处理的有效方法^[4-5]，并成功应用于神经科学^[6-7]、无线通信^[8-9]、社交网络^[10-11]等诸多领域。当使用张量表示信号/数据流时，张量分析技术通常被称为张量跟踪或自适应在线/实时流张量分解。CP (CANDECOMP/ PARAFAC)分解^[12]和Tucker分解^[13]作为经典的张量分解方法，均为张量奇异值分解(SVD, singular value decomposition)的扩展形式。为了解决CP分解中的秩估计难题和克服Tucker分解中的数据维度灾难缺点，Cichocki等^[14-15]扩展了2种典型的多维张量分析模型，对张量网络进行了全面的研究。其中，张量链式(TT, tensor train)分解^[3,16]表现出处理高阶张量的强大能力，成为多维信号处理和大规模数据分析领域的有效分析手段。然而，传统张量分解方法均属于批处理范畴，并不能对单向流模型下的多维信号流数据进行有效处理。

面向多维信号流数据的自适应分解问题，Nion和Sidiropoulos^[17]在传统CP分解的基础上提出了基于同步对角化和加权最小二乘准则的2种解决方案，为后续自适应张量学习算法的分析研究奠定了理论基础。随后，Mardani等^[18]提出了一种基于核范数正则化的指数加权最小二乘准则的在线子空间估计方法，通过跟踪低秩子空间揭示了潜在的高阶结构。Zhou等^[19]提出了一种有效的高阶在线学习算法，可实现具有任意阶数的动态张量CP分解，突破了张量阶数对自适应算法的限制。Nguyen等^[20]针对单向流的三阶张量，提出了基于交替最小二乘法结合牛顿型优化技术的快速自适应CP分解算法。Kasai^[21]假设数据位于低秩线性子空间中，提出了基于张量CP分

解的在线低秩子空间跟踪算法。文献[17-21]均是在传统CP分解的基础上探究的自适应张量分解方法，为克服CP分解中的秩估计难题，Liu等^[22]针对TT分解框架介绍了一种增量TT学习算法用于分解切块随时间递增的张量模型；Wang等^[23]则研究了能够分解工业物联网张量流的TT方法。然而，上述2种改进的TT分解技术可以归纳为增量式批处理学习方法，缺乏对张量切片数据的自适应处理能力，并且存在跟踪时变张量流数据较敏感的问题。

为了解决单向流模型中的高阶张量实时分析问题，同时解决CP秩选取难题且具备计算时变切片增量的在线分析能力，本文研究了一种自适应TT学习算法。该算法在理论上可以处理任意阶数的单向张量流数据，推导出时变增量对时序TT核张量估计及TT非时序核张量更新的影响，构造具有遗忘因子和正则项的指数权重最小二乘目标函数，利用块坐标下降(BCD, block coordinate descent)学习策略依次估计出时序TT核并对非时序TT核进行更新，从而实现对切片和切块等增量式实时数据的自适应分析研究。在模拟数据实验中，采用随机生成的四阶张量分别验证了所提算法在增量大小、不同TT秩、噪声强度和时变强度等条件下的重构性能，并且利用三阶张量进行了平均相对误差和平均运算时间的对比实验。此外，通过实测视频数据验证了多种算法对人体运动图像/视频实时分析的能力。

1 问题描述

1.1 传统TT分解模型

在传统TT分解模型中，张量 \mathcal{X} 可以表示为一系列三阶张量 $\{\mathcal{G}_n\}_{n=1}^N$ 多线性乘积的缩并形式^[16]。在不考虑加性噪声的情况下，张量 \mathcal{X} 和核张量 $\{\mathcal{G}_n\}_{n=1}^N$ 的关系可以表示为

$$\mathcal{X} = \mathcal{G}_1 \times_2^1 \mathcal{G}_2 \times_3^1 \cdots \times_N^1 \mathcal{G}_N \quad (1)$$

其中， $\mathcal{G}_n \in \mathbb{R}^{r_{n-1} \times I_n \times r_n}$ 表示第 n 个TT核张量， $n \in \{1, 2, \dots, N\}$ ， \times_n^1 表示张量间的mode-($n, 1$)缩约积， N 表示核张量数， $\{r_i\}_{i=0}^N$ 表示TT秩集合，且满足 $r_0 = r_N = 1$ ，即 $\mathcal{G}_1 \in \mathbb{R}^{I_1 \times r_1}$ ， $\mathcal{G}_N \in \mathbb{R}^{r_{N-1} \times I_N}$ 。

TT分解对高阶张量数据的处理具有以下优点：1) 给定任意张量 \mathcal{X} ，能够找到一系列TT核 $\{\mathcal{G}_n\}_{n=1}^N$ 和TT秩满足TT分解的要求；2) 相对于CP秩的

确定难题, TT 分解可以稳定有效地估计 TT 核张量; 3) TT 分解为高阶张量提供了一种节省内存的表示方式, 并且能够克服张量的维度灾难缺点。然而, 传统的 TT 分解属于全量式批处理框架, 在分析增量式自适应数据时存在重复计算和效率不高等问题。因此, 在传统 TT 分解算法的基础上开发能够处理在线数据流的自适应张量分解方法具有重要的研究意义。

1.2 单向流模型

在经典的在线应用场景中, 研究仅一个模态时变的高阶数据流分解问题受到了越来越多的关注, 将给定的 N 阶张量流分解为若干向量^[17]、矩阵^[18-20]、低秩张量^[22-23]的集合是一项具有潜在研究价值的课题。不失一般性, 假设流张量仅最后一个维度是时变的, 其余维度均固定不变, 则可以将流张量表示为 $\mathcal{X}^t \in \mathbb{R}^{I_1 \times \dots \times I_{N-1} \times I_N^t}$, 其中, I_n^t 为张量的第 N 个维度, 用于表征时变增量, $I_n, n \in \{1, 2, \dots, N-1\}$ 表示前 $N-1$ 个固定维度, 该问题模型被称为高阶张量的单向流模型。与之对应的多向流模型可定义为 $\mathcal{X}^t \in \mathbb{R}^{I_1^t \times \dots \times I_{N-1}^t \times I_N^t}$, 表示具有不止一维的流数据随时间增加的情况, 即高阶张量的时变维度 $I_n^t, n \in \{1, 2, \dots, N\}$ 均可表征时变增量。此外, 本文将针对单向流模型下实时张量数据的分析方法展开研究, 且时序切片或时序切块的定义对于描述单向流模型的张量跟踪问题至关重要, 为方便表述可将时序切片和时序切块统称为时序增量。

定义 1 时序切片和时序切块 (时序增量)。令第 t 时刻的流张量为 $\mathcal{X}^t \in \mathbb{R}^{I_1 \times \dots \times I_{N-1} \times I_N^t}$, 则定义 $\mathcal{Y}^\tau = \mathcal{X}^t(:, \dots, :, \tau) \in \mathbb{R}^{I_1 \times \dots \times I_{N-1} \times I_N}$ 表示第 τ 个时序切片, 且 $1 \leq \tau \leq I_N^t$ 。当 $I_N^t - I_N^{t-1} = 1$ 时, 第 t 时刻的增量张量为时序切片; 当 $I_N^t - I_N^{t-1} > 1$ 时, 第 t 时刻的增量张量为时序切块, 均可记为 $\mathcal{Y}^\Delta \in \mathbb{R}^{I_1 \times \dots \times I_{N-1} \times (I_N^t - I_N^{t-1})}$ 。因此, 流张量 \mathcal{X}^t 可看作由一系列时序切片 $\{\mathcal{Y}^\tau\}_{\tau=1}^{I_N^t}$ 构成, 或者由 \mathcal{X}^{t-1} 和时序切块 $\mathcal{Y}^\Delta = \{\mathcal{Y}^\tau\}_{\tau=I_N^{t-1}+1}^{I_N^t}$ 构成。不失一般性, 将时序切片和时序切块统称为时序增量, 即存在 $I_N^t - I_N^{t-1} = J$, 并将时序增量 \mathcal{Y}^Δ 简记为 \mathcal{Y}^t 。换句话说, 流张量 \mathcal{X}^t 可以通过即将到来的时序增量 \mathcal{Y}^t 附加到之前沿时变维度的第 $t-1$ 时刻观测张量 \mathcal{X}^{t-1} 上的方式连接得到, 即

$$\begin{cases} \mathcal{X}^t = \mathcal{X}^{t-1} \boxplus_N \mathcal{Y}^t \\ I_N^t = I_N^{t-1} + J \end{cases} \quad (2)$$

其中, \boxplus_N 表示张量沿 N 阶模态的连接。

一般来说, 时序增量 \mathcal{Y}^t 可以表示为

$$\mathcal{Y}^t = \mathcal{L}^t + \sigma \mathcal{N}^t \quad (3)$$

其中, \mathcal{L}^t 表示时序增量的低秩成分, \mathcal{N}^t 表示噪声张量, σ 表示噪声强度。低秩项 \mathcal{L}^t 可以用向量、矩阵、核张量等形式表示, 能够应用于静态和时变场景。综上所述, 单向流张量跟踪问题可形式化表达如下: 在第 t 时刻给定时序增量 \mathcal{Y}^t 和 \mathcal{X}^{t-1} 的先前估计 (核张量或张量因子), 则需要及时跟踪 $\mathcal{X}^t = \mathcal{X}^{t-1} \boxplus_N \mathcal{Y}^t$ 得到新的估计。

1.3 自适应 TT 分解模型

在单向流模型中, 实时张量 \mathcal{X}^t 在含噪声环境下的 TT 分解形式可以表示为

$$\mathcal{X}^t = \mathcal{G}_1^t \times_2^1 \mathcal{G}_2^t \times_3^1 \dots \times_N^1 \mathcal{G}_N^t + \sigma \mathcal{N}^t \quad (4)$$

假设 TT 核 $\{\mathcal{G}_n\}_{n=1}^{N-1}$ 为静态或变化缓慢, 则可根据式 (2) 和式 (3) 得出时序低秩项 $\mathcal{L}^t = \mathcal{G}_1^t \times_2^1 \mathcal{G}_2^t \times_3^1 \dots \times_N^1 \mathcal{G}_N^t$, $\mathcal{G}_N^t \in \mathbb{R}^{I_{N-1} \times J}$ 表示第 N 个 TT 核张量对应的时序增量因子矩阵, 且 \mathcal{G}_N^t 在 $J=1$ 的情况下表示 $r_{N-1} \times 1$ 的向量, 故可用 g_N^t 表示时序向量。由式 (3) 可知, 时序增量 \mathcal{Y}^t 可改写为 $\mathcal{Y}^t = \mathcal{G}_1^t \times_2^1 \mathcal{G}_2^t \times_3^1 \dots \times_N^1 \mathcal{G}_N^t + \sigma \mathcal{N}^t$ 。为了增加自适应 TT 分解模型的时变表征能力, 在非时序 TT 核张量的基础上增加时变项, 即 $\{\mathcal{G}_k^t\}_{k=1}^{N-1} = \{\mathcal{G}_k^{t-1}\}_{k=1}^{N-1} + \varepsilon \{\mathcal{D}_k^t\}_{k=1}^{N-1}$ 。其中, ε 表示时变因子; $\{\mathcal{D}_k^t\}_{k=1}^{N-1}$ 表示时变项, 且与 $\{\mathcal{G}_k^t\}_{k=1}^{N-1}$ 维度一致。当 $N=3$ 时, TT 张量流分解的示意如图 1 所示。从图 1 可以看出, 第 N 个 TT 核张量 $\mathcal{G}_N^t \in \mathbb{R}^{r_{N-1} \times I_N^t}$ 可以由 \mathcal{G}_N^{t-1} 和 \mathcal{G}_N^t 构成, 即 $\mathcal{G}_N^t = [\mathcal{G}_N^{t-1} \quad \mathcal{G}_N^t]$ 。

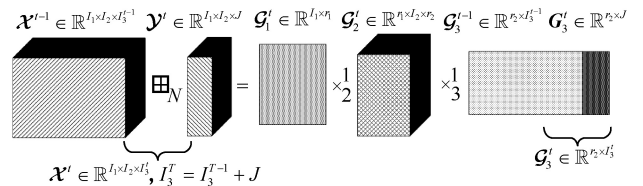


图 1 TT 张量流分解示意

从理论上分析时序 $\mathcal{G}_N^t = [\mathcal{G}_N^{t-1} \quad \mathcal{G}_N^t]$ 的推导过程。令 $\mathcal{U}^t = \mathcal{G}_1^t \times_2^1 \dots \times_{N-1}^1 \mathcal{G}_{N-1}^t$ 表示解析张量,

$\{\mathbf{g}_n^{t-1}\}_{n=1}^N$ 表示在第 $t-1$ 时刻张量 \mathbf{x}^{t-1} 对应的 TT 核集合。根据 TT 核 $\{\mathbf{g}_n\}_{n=1}^{N-1}$ 为静态或变化缓慢的三阶张量的假设, 可得出 $\mathbf{u}^t \approx \mathbf{u}^{t-1}$, 即可认为在 $t-1$ 时刻和 t 时刻的解析张量 \mathbf{u} 是近似相等的。因此, 可将第 t 时刻的张量 \mathbf{x}^t 表示为

$$\begin{aligned} \mathbf{x}^t &= \mathbf{u}^t \times_N^1 \mathbf{g}_N^t = \mathbf{x}^{t-1} \boxtimes_N \mathbf{y}^t = \\ &(\mathbf{u}^{t-1} \times_N^1 \mathbf{g}_N^{t-1}) \boxtimes_N (\mathbf{u}^t \times_N^1 \mathbf{g}_N^t) \approx \\ &\mathbf{u}^t \times_N^1 \left(\begin{bmatrix} \mathbf{g}_N^{t-1} & \mathbf{g}_N^t \end{bmatrix} \right) \end{aligned} \quad (5)$$

由图 1 和式(5)可知, 只要估计出时序增量对应的因子矩阵 $\mathbf{G}_N^t \in \mathbb{R}^{r_{N-1} \times J}$, 就可以得到第 t 时刻的核张量 $\mathbf{g}_N^t \in \mathbb{R}^{r_{N-1} \times (I_N^{t-1} + J)}$, 而不必重复对 $\mathbf{g}_N^{t-1} \in \mathbb{R}^{r_{N-1} \times I_N^{t-1}}$ 进行计算。当数据流规模比较大时, 对 \mathbf{g}_N^t 整体进行估计存在计算复杂度高、存储要求大的缺点。因此, 仅对时序增量对应的增量因子矩阵 \mathbf{G}_N^t 进行估计的策略适用于大规模信号流的场景。该策略不仅降低了数据处理规模, 而且分解后的核张量可采用分布式的存储方式。

2 算法说明

2.1 构建目标函数

由于在线数据流固有的时变性和非平稳性, 传统批处理 TT 分解方法在计算复杂度和数据存储方面没有优势, 故需要研究一种能避免重复计算且有效跟踪的自适应高阶数据流分析方法。在自适应 TT 分解框架中, 核张量的估计是研究重点。不考虑噪声的情况下, 估计 TT 核张量 $\{\mathbf{g}_n^t\}_{n=1}^N$ 的广义目标函数可表示为

$$\arg \min_{\{\mathbf{g}_n^t\}_{n=1}^N} \left\| \mathbf{x}^t - \mathbf{g}_1^t \times_2^1 \mathbf{g}_2^t \times_3^1 \cdots \times_N^1 \mathbf{g}_N^t \right\|_F^2 \quad (6)$$

根据式(2)和自适应策略^[17,20]可以将式(6)目标函数改写为指数权重最小二乘的形式^[18,21], 表示为

$$\begin{aligned} \arg \min_{\{\mathbf{g}_n^t\}_{n=1}^N} & \left\{ \sum_{i=1}^t \rho^{t-i} \left\| \mathbf{y}^i - \mathbf{g}_1^i \times_2^1 \mathbf{g}_2^i \times_3^1 \cdots \times_N^1 \mathbf{g}_N^i \right\|_F^2 + \right. \\ & \left. \mu \sum_{k=1}^{N-1} \left\| \mathbf{g}_k^t - \mathbf{g}_k^{t-1} \right\|_F^2 \right\} \end{aligned} \quad (7)$$

其中, $\rho \in (0,1]$ 表示遗忘因子, 旨在降低观测数据

间距离的影响, 并提高在动态环境中的跟踪速度; μ 表示正则化参数, 旨在控制相邻的连续时刻对应 TT 核的时变程度。当 $\rho=1$ 且 $\mu=0$ 时, 式(6)和式(7)中的 2 个目标函数是等价的。

2.2 核张量的学习策略

由于待求解的目标函数是一个非凸问题, 因此可利用块坐标下降学习策略交替/递归更新核张量。将原目标函数分解为 2 个子问题, 即分别对时序和非时序 TT 核张量进行求解。对每个子问题求解都需要在迭代时固定一部分参数, 只针对一个变量进行优化, 其余变量保持不变, 然后交替求解, 最终收敛得到最优解。因此, 基于 BCD 学习框架, 核张量的计算过程可以分为两部分。

1) 估计时序 TT 核 \mathbf{g}_N^t : 在已知前 $t-1$ 时刻的全部 TT 核 $\{\mathbf{g}_n^{t-1}\}_{n=1}^N$ 情况下, 估计第 t 时刻的时序 TT 核 $\mathbf{g}_N^t = \begin{bmatrix} \mathbf{g}_N^{t-1} & \mathbf{g}_N^t \end{bmatrix}$, 其核心是计算 \mathbf{G}_N^t 。2) 更新非时序 TT 核 $\{\mathbf{g}_k^t\}_{k=1}^{N-1}$: 在给定 $\{\mathbf{g}_k^{t-1}\}_{k=1}^{N-1}$ 和 \mathbf{g}_N^t 的情况下, 并行更新第 t 时刻的前 $N-1$ 个 TT 核。

根据 $\mathbf{u}^t \approx \mathbf{u}^{t-1}$ 可以得出, 增量因子矩阵 \mathbf{G}_N^t 的更新准则为

$$\mathbf{G}_N^t = \arg \min_{\mathbf{G}_N \in \mathbb{R}^{r_{N-1} \times J}} \left\| \mathbf{y}^t - \mathbf{u}^t \times_N^1 \mathbf{G}_N \right\|_F^2 \quad (8)$$

在此基础上, 除 \mathbf{g}_N^t 外的其余时序 TT 核张量 $\{\mathbf{g}_k^t\}_{k=1}^{N-1}$ 的更新准则可以表示为

$$\mathbf{g}_k^t = \arg \min_{\{\mathbf{g}_k\}_{k=1}^{N-1}} \sum_{i=1}^t \rho^{t-i} \left\| \mathbf{y}^i - \mathbf{A}_k^{t-1} \times_k^1 \mathbf{g}_k \times_{k+1}^1 \mathbf{B}_k^i \right\|_F^2 \quad (9)$$

其中, \mathbf{A} 和 \mathbf{B} 为虚拟张量, 分别定义为

$$\begin{cases} \mathbf{A}_k^{t-1} = \mathbf{g}_1^{t-1} \times_2^1 \cdots \times_{k-1}^1 \mathbf{g}_{k-1}^{t-1} \\ \mathbf{B}_k^i = \mathbf{g}_{k+1}^{i-1} \times_{k+2}^1 \cdots \times_{N-1}^1 \mathbf{g}_{N-1}^{i-1} \times_N^1 \mathbf{g}_N^i \end{cases} \quad (10)$$

此外, 核张量更新问题的假设条件为: 1) TT 核 $\{\mathbf{g}_k\}_{k=1}^{N-1}$ 在 $t-1$ 和 t 这 2 个连续时刻的改变很小, 即 $\{\mathbf{g}_k^{t-1}\}_{k=1}^{N-1} \approx \{\mathbf{g}_k^t\}_{k=1}^{N-1}$; 2) TT 秩向量 $\{r_i\}_{i=0}^N$ 已知, 且为时不变序列, 其中 $r_0 = r_N = 1$ 。

2.3 具体步骤

2.3.1 估计时序 TT 核 \mathbf{g}_N^t

在给定时序张量 \mathbf{y}^t 和解析张量 \mathbf{u}^{t-1} 的情况下,

\mathbf{G}'_N 可在式(8)的基础上加正则项的方式解决, 即

$$\mathbf{G}'_N = \arg \min_{\mathbf{G}_N \in \mathbb{R}^{r_{N-1} \times J}} \left\| \mathcal{Y}' - \mathbf{U}^{t-1} \times_N^1 \mathbf{G}_N \right\|_F^2 + \eta \left\| \mathbf{G}_N \right\|_F^2 \quad (11)$$

其中, $\arg \min_{\mathbf{G}_N \in \mathbb{R}^{r_{N-1} \times J}} \left\| \mathcal{Y}' - \mathbf{U}^{t-1} \times_N^1 \mathbf{G}_N \right\|_F^2$ 旨在计算第 t 时刻对应观测与估计的时序切片间最小化残差, $\eta > 0$ 表示正则化经验参数, $\eta \left\| \mathbf{G}_N \right\|_F^2$ 旨在避免不适定的计算过程。为了便于分析, 式(11)可改写为矩阵-向量表示的形式。

$$\mathbf{G}'_N = \arg \min_{\mathbf{G}_N \in \mathbb{R}^{r_{N-1} \times J}} \left\| \mathbf{Y}' - \mathbf{U}^{t-1} \mathbf{G}_N \right\|_2^2 + \eta \left\| \mathbf{G}_N \right\|_F^2 \quad (12)$$

其中, $\mathbf{Y}' \in \mathbb{R}^{I_1 \cdots I_{N-1} \times J}$ 和 $\mathbf{U}^{t-1} \in \mathbb{R}^{I_1 \cdots I_{N-1} \times r_{N-1}}$ 分别表示 \mathcal{Y}' 和 \mathbf{U}^{t-1} 的展开矩阵。

因此, 可以将式(12)中矩阵 \mathbf{G}_N 的 J 个列分解为相关的 J 个切片, 表示为正则最小二乘的形式, 即

$$\mathbf{G}'_N(:, j) = \mathbf{g}'_j = \arg \min_{\mathbf{g}_j \in \mathbb{R}^{r_{N-1} \times 1}} \left\| \mathbf{y}'_j - \mathbf{U}^{t-1} \mathbf{g}_j \right\|_2^2 + \eta \left\| \mathbf{g}_j \right\|_2^2 \quad (13)$$

其中, $\mathbf{g}'_j \in \mathbb{R}^{r_{N-1} \times 1}$ 表示矩阵 $\mathbf{G}'_N \in \mathbb{R}^{r_{N-1} \times J}$ 的第 j 个列向量, $\mathbf{y}'_j = \mathbf{Y}'(:, j) \in \mathbb{R}^{I_1 \cdots I_{N-1} \times 1}$ 表示矩阵 $\mathbf{Y}' \in \mathbb{R}^{I_1 \cdots I_{N-1} \times J}$ 的第 j 个列向量, $j \in \{1, \dots, J\}$ 。则式(13)的封闭解可以表示为

$$\mathbf{G}'_N(:, j) = \mathbf{g}'_j = \left[(\mathbf{U}^{t-1})^T \mathbf{U}^{t-1} + \eta \mathbf{I}_{r_{N-1}} \right]^{-1} (\mathbf{U}^{t-1})^T \mathbf{y}'_j \quad (14)$$

在此基础上, 时序 TT 核张量 \mathcal{G}'_N 可以通过 $\mathcal{G}'_N = [\mathcal{G}'_N^{-1} \quad \mathcal{G}'_N]$ 不断学习更新。每次仅计算与时变增量数据对应的 \mathcal{G}'_N 即可更新 \mathcal{G}'_N , 而不必重复计算 \mathcal{G}'_N^{-1} , 避免了大规模在线高维数据的爆炸式增长带来的计算复杂度和存储压力。

2.3.2 依次计算非时序 TT 核 $\{\mathcal{G}'_k\}_{k=1}^{N-1}$

根据当前的时序 TT 核 \mathcal{G}'_N , 通过最小化式(15)依次更新剩余的 $N-1$ 个非时序 TT 核 $\{\mathcal{G}'_k\}_{k=1}^{N-1}$, 表示为

$$\mathcal{G}'_k = \arg \min_{\{\mathcal{G}'_k\}_{k=1}^{N-1}} \left\{ \sum_{i=1}^t \rho^{t-i} \left\| \mathcal{Y}' - \mathbf{A}_k^{t-1} \times_k^1 \mathcal{G}'_k \times_{k+1}^1 \mathcal{B}_k \right\|_F^2 + \mu \left\| \mathcal{G}'_k - \mathcal{G}'_k^{t-1} \right\|_F^2 \right\} \quad (15)$$

其中, $\rho \in (0, 1]$ 表示遗忘因子, μ 表示正则化参数。为方便计算, 将式(15)改写为矩阵形式, 可表示为

$$\mathbf{G}'_k = \arg \min_{\mathbf{G}_k \in \mathbb{R}^{I_k \times r_{k-1}}} \left\{ \sum_{i=1}^t \rho^{t-i} \left\| \mathbf{Y}_k^i - \mathbf{G}_k \left(\mathbf{B}_k^i \otimes \mathbf{A}_k^{t-1} \right) \right\|_F^2 + \mu \left\| \mathbf{G}_k - \mathbf{G}_k^{t-1} \right\|_F^2 \right\} \quad (16)$$

其中, $\mathbf{G}'_k \in \mathbb{R}^{I_k \times r_{k-1}}$ 和 $\mathbf{Y}_k^i \in \mathbb{R}^{I_k \times I_1 \cdots I_{k-1} I_{k+1} \cdots I_{N-1}}$ 分别为 \mathcal{G}'_k 和 \mathcal{Y}' 的展开矩阵, 令 $\mathbf{W}_k^i = \mathbf{B}_k^i \otimes \mathbf{A}_k^{t-1}$, 则 $\mathbf{A}_k^{t-1} \in \mathbb{R}^{r_{k-1} \times I_1 \cdots I_{k-1}}$ 和 $\mathbf{B}_k^i \in \mathbb{R}^{I_k \times I_{k+1} \cdots I_{N-1}}$ 分别是 \mathcal{A}_k^{t-1} 和 \mathcal{B}_k^i 的展开矩阵。

由于前 $N-1$ 个 TT 核的矩阵形式 $\mathbf{G}'_k \in \mathbb{R}^{I_k \times r_{k-1}}$ 中, I_k ($k \in \{1, \dots, N-1\}$) 是固定不变的, 则可以采取与计算 \mathbf{G}'_N 类似的步骤, 通过估计 \mathbf{G}'_k 对应的每个行向量 $\mathbf{g}'_{k,m} = \mathbf{G}'_k(m, :) \in \mathbb{R}^{1 \times r_{k-1}}$ ($m \in \{1, \dots, I_k\}$), 进而得到 $N-1$ 个 \mathbf{G}'_k 。将式(16)改写为 \mathbf{G}'_k 第 m 个行向量 $\mathbf{g}'_{k,m}$ 的目标函数, 可以表示为

$$\mathbf{g}'_{k,m} = \arg \min_{\mathbf{g}_{k,m} \in \mathbb{R}^{1 \times r_{k-1}}} \left\{ \sum_{i=1}^t \rho^{t-i} \left\| \mathbf{y}_{k,m}^i - \mathbf{g}_{k,m} \mathbf{W}_k^i \right\|_2^2 + \mu \left\| \mathbf{g}_{k,m} - \mathbf{g}_{k,m}^{t-1} \right\|_2^2 \right\} \quad (17)$$

其中, $\mathbf{y}_{k,m}^i = \mathbf{Y}_k^i(m, :)$ 。令式(17)的梯度为 0, 可推导出 $\mathbf{g}'_{k,m}$, 即

$$\mathbf{g}'_{k,m} = \left(\mu \mathbf{I}_{r_{k-1}} + \mathbf{S}'_{k,m} \right)^{-1} \left[\mu \left(\mathbf{g}_{k,m}^{t-1} \right)^T + \mathbf{R}'_{k,m} \right]^T \quad (18)$$

其中, 解析矩阵分别为 $\mathbf{S}'_{k,m} = \sum_{i=1}^t \rho^{t-i} \mathbf{W}_k^i \left(\mathbf{W}_k^i \right)^T$, $\mathbf{R}'_{k,m} = \sum_{i=1}^t \rho^{t-i} \mathbf{W}_k^i \mathbf{y}_{k,m}^i{}^T$ 。此外, 矩阵 $\mathbf{S}'_{k,m}$ 和 $\mathbf{R}'_{k,m}$ 通过递归学习的方式更新, 则学习准则可以表示为 $\mathbf{S}'_{k,m} = \rho \mathbf{S}'_{k,m}^{t-1} + \mathbf{W}_k^t \left(\mathbf{W}_k^t \right)^T$ 和 $\mathbf{R}'_{k,m} = \rho \mathbf{R}'_{k,m}^{t-1} + \mathbf{W}_k^t \mathbf{y}_{k,m}^t{}^T$ 。

根据式(18), $\mathbf{g}'_{k,m}$ 的更新规则可以表示为

$$\mathbf{g}'_{k,m} = \mathbf{g}_{k,m}^{t-1} + \left(\Delta \mathbf{y}'_{k,m} \mathbf{W}_k^{t-1} + \rho \mu \Delta \mathbf{g}'_{k,m} \right) \cdot \left(\mu \mathbf{I}_{r_{k-1}} + \mathbf{S}'_{k,m} \right)^{-T} \quad (19)$$

其中, $\Delta \mathbf{y}'_{k,m} = \mathbf{y}'_{k,m} - \mathbf{g}_{k,m}^{t-1} \mathbf{W}_k^t$, $\Delta \mathbf{g}'_{k,m} = \mathbf{g}'_{k,m} - \mathbf{g}_{k,m}^{t-2}$ 。

因此, 根据低空间复杂度的递归准则同步更新矩阵 \mathbf{G}'_k , 可以表示为

$$\mathbf{G}'_k = \mathbf{G}'_k^{t-1} + \left[\left(\Delta \mathbf{Y}_k^t \mathbf{W}_k^t \right)^T + \rho \mu \Delta \mathbf{G}'_k \right] \cdot \left(\mu \mathbf{I}_{r_{k-1}} + \mathbf{S}'_k \right)^{-T} \quad (20)$$

其中, $\Delta \mathbf{Y}_k^t = \mathbf{Y}_k^t - \mathbf{G}'_k^{t-1} \mathbf{W}_k^t$, $\Delta \mathbf{G}'_k = \mathbf{G}'_k^{t-1} - \mathbf{G}'_k^{t-2}$ 。然

后, 将式(20)得到的矩阵转变为三阶张量的形式, 即可恢复核张量 $\{\mathbf{g}_k^t\}_{k=1}^{N-1} \in \mathbb{R}^{r_{k-1} \times I_k \times r_k}$, 其中 $r_0 = 1$ 。将第 t 次迭代更新后的非时序 TT 核张量 $\{\mathbf{g}_k^t\}_{k=1}^{N-1}$ 作为第 $t+1$ 次迭代时估计时序 TT 核张量 \mathbf{g}_N^{t+1} 的输入, 依次估计出所有 TT 核张量。综上所述, 所提的基于单向流模型的自适应 TT 学习算法估计核张量步骤如算法 1 所示。

算法 1 核张量估计步骤

输入 时序张量 \mathcal{Y}^t 、解析张量 \mathcal{U}^{t-1} 、TT 秩 $\{r_i\}_{i=0}^N$ 、遗忘因子 ρ 、正则化参数 μ 和 η

输出 时序 TT 核张量 \mathbf{g}_N^t 和非时序 TT 核张量 $\{\mathbf{g}_k^t\}_{k=1}^{N-1}$

初始化 非时序 TT 核张量 $\{\mathbf{g}_k^t\}_{k=1}^{N-1}$

for $t = 1, 2, 3, \dots$

1) 估计时序 TT 核张量 \mathbf{g}_N^t

for $j = 1 : J$

根据式(14)计算第 j 个切片对应的因子矩阵 $\mathbf{G}_N^t(:, j)$;

end for

根据 $\mathbf{g}_N^t = [\mathbf{g}_N^{t-1} \quad \mathbf{G}_N^t]$ 得到时序 TT 核 \mathbf{g}_N^t

2) 更新非时序 TT 核张量 $\{\mathbf{g}_k^t\}_{k=1}^{N-1}$

for $k = 1 : N-1$

根据式(19)计算第 k 个核张量展开矩阵的第 m 个行向量 $\mathbf{g}_{k,m}^t$

根据式(20)更新展开矩阵 \mathbf{G}_k^t

将 \mathbf{G}_k^t 转变为三阶张量的形式, 即

$$\{\mathbf{g}_k^t\}_{k=1}^{N-1} \in \mathbb{R}^{r_{k-1} \times I_k \times r_k}$$

end for

3) 将更新后的非时序 TT 核张量 $\{\mathbf{g}_k^t\}_{k=1}^{N-1}$

作为第 $t+1$ 次迭代时步骤 1) 的输入, 重复上述步骤直至迭代终止。

end for

为了分析所提算法的计算复杂度, 不妨假设张量的前 $N-1$ 维具有超对称结构, 即 $\{I_n\}_{n=1}^{N-1} = I$, 且 TT 秩 $\{r_i\}_{i=1}^{N-1} = r$ 。估计时序 TT 核张量 \mathbf{g}_N^t 和非时序 TT 核张量 $\{\mathbf{g}_k^t\}_{k=1}^{N-1}$ 的计算复杂度可以分别表示为 $\mathcal{O}(JI^{N-1}r^2)$ 和 $\mathcal{O}((N-1)I^{N-1}r^4)$, 则所提算法的计算复杂度可以表示为 $\mathcal{O}(I^{N-1}r^2 \max\{J, (N-1)r^2\})$ 。综

上所述, 在张量维度 N 和 TT 秩 r 固定的情况下, 时序增量 J 的大小将决定算法的整体计算复杂度。此外, 相较于传统批处理 TT 分解需要计算 N 阶数据^[3,16], 所提算法仅处理 $N-1$ 阶时序增量, 具有计算复杂度和存储要求低的优点。

3 实验分析

3.1 模拟数据仿真

为了验证所提算法对高阶张量数据流的自适应分解/张量跟踪能力, 考虑在非平稳环境下分别从增量大小、不同 TT 秩、噪声强度和时变强度等角度出发对所提算法展开仿真实验。模拟仿真采用随机生成的四阶张量流数据, 记为 $\mathcal{X}^t \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4}$, TT 秩记为 $\{r_1, r_2, r_3\}$, 时序增量可以表示为 $\mathcal{Y}^t = \mathbf{g}_1^t \times_2 \mathbf{g}_2^t \times_3 \mathbf{g}_3^t \times_4 \mathbf{G}_4^t + \sigma \mathcal{N}^t$ 。其中, $\mathbf{g}_1^t \in \mathbb{R}^{I_1 \times r_1}$ 、 $\mathbf{g}_2^t \in \mathbb{R}^{r_1 \times I_2 \times r_2}$ 、 $\mathbf{g}_3^t \in \mathbb{R}^{r_2 \times I_3 \times r_3}$ 分别表示非时序 TT 核张量, $\mathbf{G}_4^t \in \mathbb{R}^{r_3 \times J}$ 表示时序 TT 核张量对应的解析矩阵。此外, 在非时序 TT 核张量上添加时变项可以分别表示为 $\mathbf{g}_1^t = \mathbf{g}_1^{t-1} + \varepsilon \mathbf{D}_1^t$ 、 $\mathbf{g}_2^t = \mathbf{g}_2^{t-1} + \varepsilon \mathbf{D}_2^t$ 、 $\mathbf{g}_3^t = \mathbf{g}_3^{t-1} + \varepsilon \mathbf{D}_3^t$, 并采用相对误差准则评估张量重构的准确性, 即

$$\xi(\mathcal{X}, \hat{\mathcal{X}}) = \frac{\|\mathcal{X} - \hat{\mathcal{X}}\|_F}{\|\mathcal{X}\|_F} \quad (21)$$

其中, $\hat{\mathcal{X}}$ 表示 \mathcal{X} 的估计。相对误差准则 ξ 越小, 说明算法对张量流的跟踪、自适应分解、重构能力越强, 反之亦然。此外, 所有仿真均在 MATLAB2019a 上进行, 计算机配置为 Intel(R) Core(TM) i7-10710U CPU @1.10 GHz 1.61 GHz, RAM 为 16 GB。

令四阶张量的维度为 [10 15 20 1000], 且假设最后一个维度时变, 可称为时序点数, TT 秩固定为 [5 5 5], 噪声强度固定为 $\sigma = 10^{-3}$, 时变强度固定为 $\varepsilon = 10^{-3}$, 增量大小分别为 $J = \{7, 5, 3, 1\}$, 并在时序点数为 550 时增加突变干扰, 即此刻的突变强度为 1。不失一般性, 鉴于参数选取不是本文研究的重点内容, 故设定所提算法的遗忘因子 $\rho = 0.5$, 正则化参数 $\eta = 1$, $\mu = 1$, 且仿真结果均是算法迭代 30 次的平均值。非平稳条件下, 增量大小对算法性能的影响如图 2 所示。由图 2 可知, 算法在 $J = 1$ 时比在 $J = \{7, 5, 3\}$ 时的相对误差收敛速度更快且稳态误差更小。随

随着时序增量的增加，张量分解相对误差收敛速度变慢、稳态误差也随之变大，说明所提算法在处理时序切块方面的性能还有待提升。此外，所提算法表现出对张量切片较强的学习能力，符合自适应算法对数据逐点处理的要求，故其余实验均采用张量切片 $J=1$ 作为时序增量。

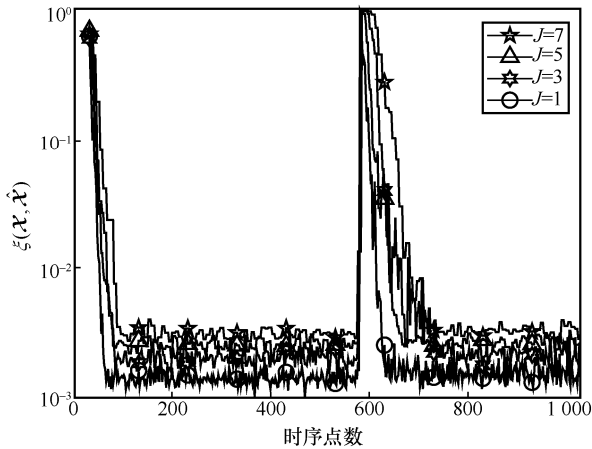


图 2 增量大小对算法性能的影响

固定时序增量为时序切片，选取 TT 秩分别为 $[2\ 2\ 2]$ 、 $[4\ 4\ 4]$ 、 $[6\ 6\ 6]$ 和 $[8\ 8\ 8]$ ，其余实验条件和参数不变。TT 秩对算法性能的影响如图 3 所示。由图 3 可知，当 TT 秩为 $[8\ 8\ 8]$ 时，所提算法在时序突变后出现了难以收敛的情况，说明较大的 TT 秩会限制 TT 分解的低秩表征能力。此外，当 TT 秩较小时，算法虽然收敛速度快但稳态误差不理想波动较大。因此，综合考虑收敛速度和稳态误差两方面因素，TT 秩的选取不宜过大或过小，需根据张量最小维度值选择，故令模拟数据仿真实验中的 TT 秩为 $[5\ 5\ 5]$ 。

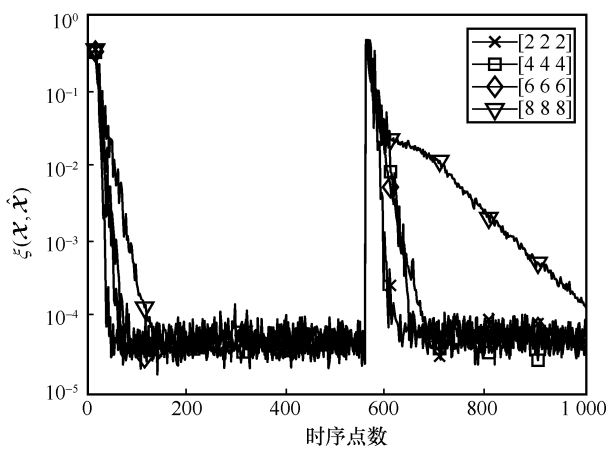


图 3 TT 秩对算法性能的影响

为了验证所提算法的抗噪性，分别选取噪声强度 $\sigma = \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ 进行鲁棒性实验，其余仿真条件和参数不变。噪声强度对算法性能的影响如图 4 所示。由图 4 可知，选取的 4 个噪声强度对算法的收敛速度影响不大，其收敛曲线基本一致。但当 $\sigma = 10^{-1}$ 时，算法的稳态误差明显比其他噪声强度大，说明在噪声强度为 $\sigma = \{10^{-4}, 10^{-3}, 10^{-2}\}$ 时所提算法表现出了较好的鲁棒性。此外，所提算法对较大的噪声强度敏感，当噪声强度较小时算法性能变化不大，并且不同噪声强度下算法在突变前收敛速度也比较接近。

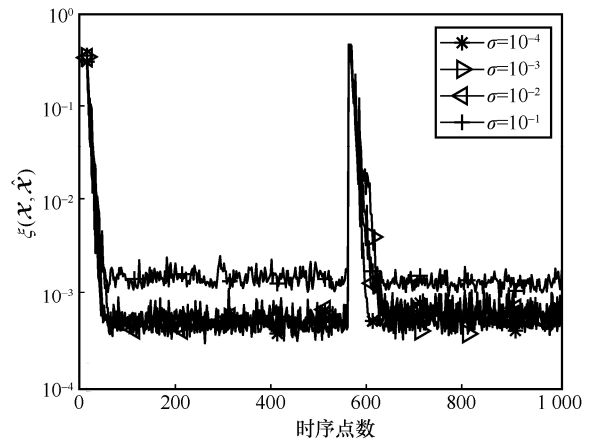


图 4 噪声强度对算法性能的影响

固定实验的其余仿真条件和参数，分别选取时变强度 $\varepsilon = \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ 验证所提算法性能。时变强度对算法性能的影响如图 5 所示。由图 5 可知，时变强度越大，算法的稳态误差也越大，且在突变干扰后收敛速度主要受强时变的影响，如 $\varepsilon = 10^{-1}$ 。当 $\varepsilon = \{10^{-4}, 10^{-3}\}$ 时，算法的性能比较理想，因此不断提升算法的抗时变能力是自适应张量分解研究的优化方向。

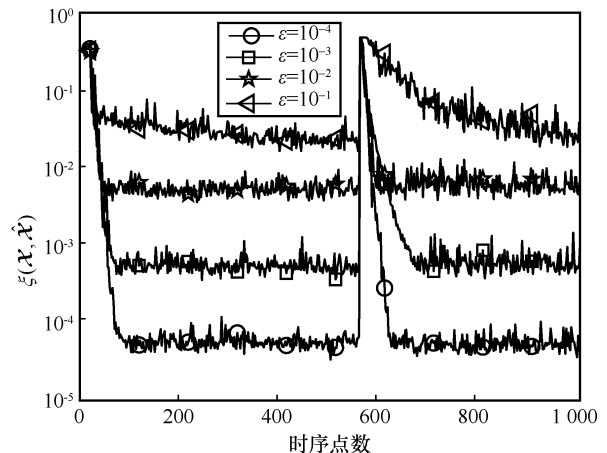


图 5 时变强度对算法性能的影响

上述实验分别从增量大小、TT 秩选取、噪声强度和时变强度几个方面验证了所提算法的性能,接下来将 4 种自适应张量分解方法作为对比算法,分别为 PARAFAC-RLST^[17]、PARAFAC-SDT^[17]、TeCPSGD^[18]和 SOAP^[20],来分析随机生成的在线三阶张量。其中,张量维度为[150 200 1200],噪声强度和时变强度均固定为 10^{-3} ,增量 $J=1$,在时序点数为 600 时增加突变干扰,且对比算法的秩均为 10,所提算法的 TT 秩为[10 10]。5 种算法估计张量流的相对误差对比如图 6 所示。由图 6 可知,PARAFAC-RLST 和 PARAFAC-SDT 具有相似的收敛速度和稳态误差,突变后的稳态误差均不理想,但突变前后的收敛速度较快。TeCPSGD 收敛速度最慢,且稳态误差较大,算法收敛困难。SOAP 在突变前表现出与 PARAFAC-RLST 和 PARAFAC-SDT 相似的算法性能,但突变后相对误差几乎不变,说明 SOAP 方法不适用于存在突变的数据分析或者信号处理应用。相较于对比算法,所提算法在突变前后均表现出了良好的收敛速度和稳态误差。

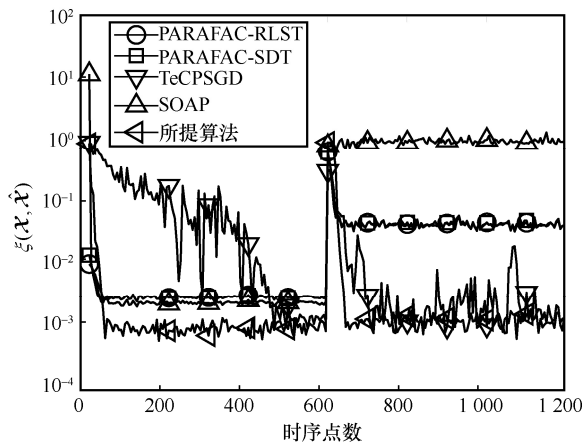


图 6 5 种算法估计张量流的相对误差对比

不同算法处理模拟数据的平均运算时间如图 7 所示。由图 7 可知,PARAFAC-RLST 平均运算时间最长,其次分别为 TeCPSGD、SOAP 和 PARAFAC-SDT,所提算法平均运算时间最短,说明 BCD 学习策略中仅处理时序增量的方式在算法运算时间方面具有优势。综上所述,不同于对比算法仅能处理三阶张量,所提算法能够处理任意阶数的张量数据。此外,综合考虑相对误差指标和平均运算时间可知,所提算法更新非时序 TT 核张量的并行学习策略不仅能够缩短算法运算时间,而且可以获取稳健的估计性能。

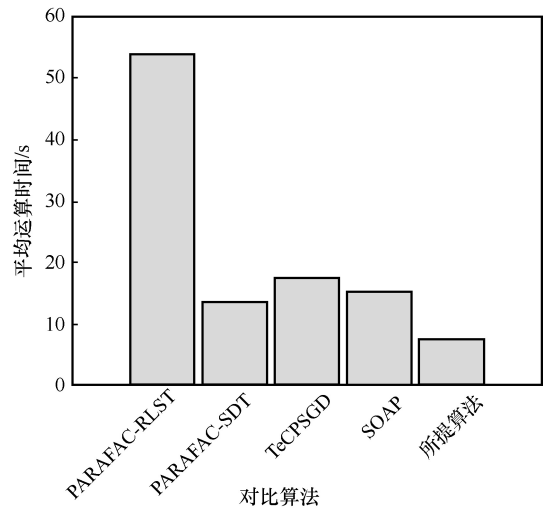


图 7 不同算法处理模拟数据的平均运算时间

3.2 实测数据实验

将视频信号^[24]作为实测数据验证算法张量重构性能,选取数据库中名为‘eli_wave1’的视频作为自适应张量算法的输入信号。其中,视频属性包括:数据时长为 5 s,帧率为 25 frame/s,帧的高度为 144,帧的宽度为 180,每个像素的位数为 24,视频格式为 RGB24,则视频流中的帧数为 125。为了更公平地验证算法性能,将每个视频帧数据对应的 RGB 图像转换为灰度图像,故输入信号的维度为 $144 \times 180 \times 125$ 。将视频帧作为时序增量,不同算法处理视频数据的平均相对误差如图 8 所示。由图 8 可知,PARAFAC-RLST 收敛速度较快,但相对误差较大;PARAFAC-SDT 收敛速度最慢,在 90 帧左右才收敛;所提算法性能明显优于对比算法,其次是 SOAP 和 TeCPSGD。对比图 6 可知,实测数据的时变强度小,因而所有算法的性能曲线均波动小且平滑,且算法的相对误差明显变差。

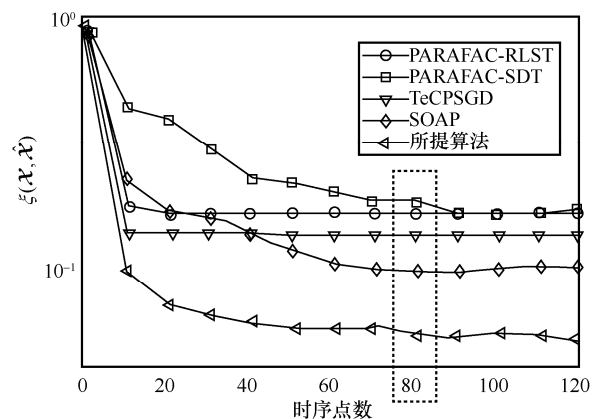


图 8 不同算法处理视频数据的平均相对误差

不失一般性, 选取各算法重构后张量的第 80 帧进行可视化, 人体运动部分以虚线椭圆形标注, 可视化灰度图如图 9 所示。所提算法重构的可视化灰度图与原始视频帧的灰度图最接近, 基本可以分辨出人体轮廓。而 SOAP 算法仅能识别出人体位置, 难以辨别动作状态, 其余 3 种对比算法可视化灰度图较差。图 9 的可视化结果与图 8 中虚线方框对应, 即当相对误差 $\xi > 10^{-1}$ 时, 自适应张量分解算法重构的张量切片可视化不理想, 难以识别人体动作, 不能应用于人体运动图像/视频实时分析。此外, 所提算法在实测视频数据的实验中表现出了优于对比算法的张量切片重构能力。

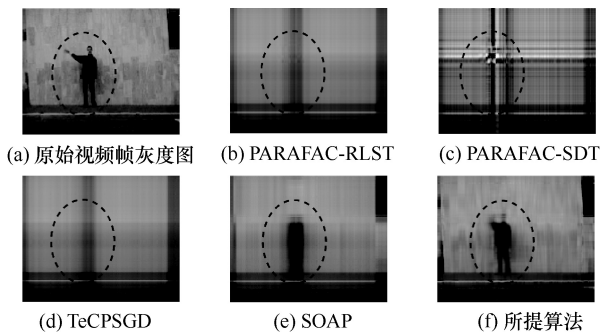


图 9 第 80 帧的可视化灰度图

4 结束语

由于泛在的实时信号/数据大多呈现高维结构, 将给传统批处理张量分解方法和受阶数限制的自适应张量分析方法带来挑战, 因此本文研究了一种基于单向流模型的自适应 TT 学习算法。通过推导单向流时序增量对 TT 核更新的影响, 将核张量分为时序和非时序两类, 并构造相应的目标函数利用 BCD 学习策略分别估计更新。实验表明, 所提方法能够处理三阶和四阶的模拟数据, 在平均相对误差和平均运算时间方面优于对比算法, 且在视频实时分析研究中重构张量数据具有一定的意义。此外, 在现有工作基础上研究基于多向流模型的自适应张量学习算法将是下一步重点探究方向。

参考文献:

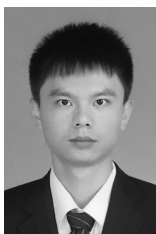
[1] KOLAJO T, DARAMOLA O, ADEBIYI A. Big data stream analysis: a systematic literature review[J]. *Journal of Big Data*, 2019, 6(1): 1-30.
 [2] BAHRI M, BIFET A, GAMA J, et al. Data stream analysis: foundations, major tasks and tools[J]. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2021, 11(3): 1405.
 [3] BENGUA J A, PHIEN H N, TUAN H D, et al. Efficient tensor com-

pletion for color image and video recovery: low-rank tensor train[J]. *IEEE Transactions on Image Processing*, 2017, 26(5): 2466-2479.
 [4] SIDIROPOULOS N D, LATHAUWER L D, FU X, et al. Tensor decomposition for signal processing and machine learning[J]. *IEEE Transactions Signal Processing*, 2017, 65(13): 3551-3582.
 [5] 马宝泽, 李国军, 向翠玲, 等. 基于张量分析的欠定混合矩阵估计算法[J]. *通信学报*, 2022, 43(11): 35-43.
 MA B Z, LI G J, XIANG C L, et al. Underdetermined mixing matrix estimation algorithm based on tensor analysis[J]. *Journal on Communications*, 2022, 43(11): 35-43.
 [6] THANH L T, DAO N T A, DUNG N V, et al. Multi-channel EEG epileptic spike detection by a new method of tensor decomposition[J]. *Journal of Neural Engineering*, 2020, 17(1): 016023.
 [7] CONG F, LIN Q H, KUANG L D, et al. Tensor decomposition of EEG signals: a brief review[J]. *Journal of Neuroscience Methods*, 2015, 248: 59-69.
 [8] NION D, SIDIROPOULOS N D. Tensor algebra and multidimensional harmonic retrieval in signal processing for MIMO radar[J]. *IEEE Transactions on Signal Processing*, 2010, 58(11): 5693-5705.
 [9] CHEN H Y, AHMAD F, VOROBYOV S, et al. Tensor decompositions in wireless communications and MIMO radar[J]. *IEEE Journal of Selected Topics in Signal Processing*, 2021, 15(3): 438-453.
 [10] NAKATSUJI M, ZHANG Q P, LU X H, et al. Semantic social network analysis by cross-domain tensor factorization[J]. *IEEE Transactions on Computational Social Systems*, 2017, 4(4): 207-217.
 [11] FERNANDES S, FANAEE-T H, GAMA J. Tensor decomposition for analysing time-evolving social networks: an overview[J]. *Artificial Intelligence Review*, 2021, 54(4): 2891-2916.
 [12] HARSHMAN R A. Foundations of the PARAFAC procedure: models and conditions for an explanatory multimodal factor analysis[J]. *UC-LA Working Papers in Phonetics*, 1970, 16: 1-84.
 [13] TUCKER L R. Some mathematical notes on three-mode factor analysis[J]. *Psychometrika*, 1966, 31(3): 279-311.
 [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] CICHOCKI A, PHAN A H, ZHAO Q, et al. Tensor networks for dimensionality reduction and large-scale optimization: part 2 applications and future perspectives[J]. *Foundations and Trends in Machine Learning*, 2017, 9(6): 431-673.
 [16] OSELEDETS I V. Tensor-train decomposition[J]. *SIAM Journal on Scientific Computing*, 2011, 33(5): 2295-2317.
 [17] NION D, SIDIROPOULOS N D. Adaptive algorithms to track the PARAFAC decomposition of a third-order tensor[J]. *IEEE Transactions on Signal Processing*, 2009, 57(6): 2299-2310.
 [18] MARDANI M, MATEOS G, GIANNAKIS G B. Subspace learning and imputation for streaming big data matrices and tensors[J]. *IEEE Transactions on Signal Processing*, 2015, 63(10): 2663-2677.
 [19] ZHOU S, VINH N X, BAILEY J, et al. Accelerating online CP decompositions for higher order tensors[C]//*Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM Press, 2016: 1375-1384.
 [20] NGUYEN V D, ABED M K, TRUNG N L. Second-order optimization

based adaptive PARAFAC decomposition of three-way tensors[J]. Digital Signal Processing, 2017, 63:100-111.

- [21] KASAI H. Fast online low-rank tensor subspace tracking by CP decomposition using recursive least squares from incomplete observations[J]. Neurocomputing, 2019, 347: 177-190.
- [22] LIU H Z, YANG L T, GUO Y M, et al. An incremental tensor-train decomposition for cyber-physical-social big data[J]. IEEE Transactions on Big Data, 2021, 7(2): 341-354.
- [23] WANG X K, YANG L T, WANG Y H, et al. ADTT: a highly efficient distributed tensor-train decomposition method for IIoT big data[J]. IEEE Transactions on Industrial Informatics, 2021, 17(3): 1573-1582.
- [24] KUEHNE H, JHUANG H, GARROTE E, et al. HMDB: a large video database for human motion recognition[C]//Proceedings of 2011 International Conference on Computer Vision. Piscataway: IEEE Press, 2011: 2556-2563.

[作者简介]



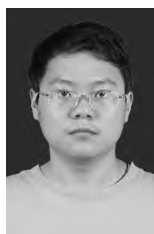
马宝泽 (1990-)，男，河北廊坊人，博士，重庆邮电大学讲师，主要研究方向为盲源分离、信道辨识、数据分析、人工智能等。



李国军 (1978-)，男，四川资阳人，博士，重庆邮电大学教授、博士生导师，主要研究方向为复杂恶劣环境超视距无线通信与网络。



邢隆 (1979-)，男，四川南充人，重庆邮电大学高级工程师，主要研究方向为通信侦察与通信对抗。



叶昌荣 (1989-)，男，重庆人，博士，重庆邮电大学副教授，主要研究方向为短波通信、信号处理、异构链路融合组网等。