



中国科技期刊卓越行动计划项目入选期刊

控制与决策

CONTROL AND DECISION



一种基于模特征的增量式张量Tucker分解方法

渠超洋, 韩建军

引用本文:

渠超洋, 韩建军. 一种基于模特征的增量式张量Tucker分解方法[J]. *控制与决策*, 2024, 39(7): 2431–2437.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2023.1003>

您可能感兴趣的其他文章

Articles you may be interested in

基于矩阵的双论域模糊概率粗糙集增量更新算法

Incremental updating of fuzzy probability rough sets over two universes based on matrix method

控制与决策. 2021, 36(3): 553–564 <https://doi.org/10.13195/j.kzyjc.2019.0692>

基于局部信息增量与MPLS的质量相关故障检测方法

Quality-related fault detection method based on local information increment and MPLS

控制与决策. 2021, 36(7): 1647–1654 <https://doi.org/10.13195/j.kzyjc.2019.1402>

基于HI-DD-AdaBoost.RT的锂离子动力电池SOH预测

Prediction of Li-ion battery SOH based on HI-DD-AdaBoost.RT

控制与决策. 2021, 36(3): 686–692 <https://doi.org/10.13195/j.kzyjc.2019.0764>

基于分类特征约束变分伪样本生成器的类增量学习

Class incremental learning based on variational pseudo-sample generator with classification feature constraints

控制与决策. 2021, 36(10): 2475–2482 <https://doi.org/10.13195/j.kzyjc.2020.0228>

一种基于池计算的宽度学习系统

A broad learning system based on reservoir computing

控制与决策. 2021, 36(9): 2203–2210 <https://doi.org/10.13195/j.kzyjc.2019.1729>

一种基于模特征的增量式张量 Tucker 分解方法

渠超洋, 韩建军[†]

(华中科技大学 计算机科学与技术学院, 武汉 430000)

摘要: 随着数据量的爆炸式增长, 边缘计算在大数据处理中的作用愈加重要. 现实应用中产生的数据通常建模表示成高阶增量式张量的形式, 增量式张量 Tucker 分解是一种高效挖掘高阶海量数据中隐藏信息的方法. 针对传统增量式张量分解忽视张量模特征对分解过程的影响、分解结果不能较好保留原始数据特征的问题, 提出一种基于模特征的增量式张量 Tucker 分解方法 ITTDMC (incremental tensor tucker decomposition based on mode characteristics). 首先, 用模长增量决定增量因子矩阵更新顺序, 以此降低更新顺序带来的重构误差; 其次, 根据模熵变化比决定增量因子矩阵更新权重, 使分解结果更准确保留各模特征; 然后, 将过往时刻的模特征和更新参数记录在指导张量中, 遇到模特征相似的增量数据时直接使用来指导张量中参数的更新, 避免重复计算, 降低时间开销; 最后, 在合成和真实数据集上进行大量的实验, 实验结果表明 ITTDMC 在模特征明显的数据集上能显著降低 (最高可达 29%) 增量式张量的重构误差.

关键词: 大数据; 增量式张量; Tucker 分解; 模特征; 模长增量; 模熵

中图分类号: TP301

文献标志码: A

DOI: 10.13195/j.kzyjc.2023.1003

引用格式: 渠超洋, 韩建军. 一种基于模特征的增量式张量 Tucker 分解方法 [J]. 控制与决策, 2024, 39(7): 2431-2437.

An incremental tensor tucker decomposition method based on model characteristics

QU Chao-yang, HAN Jian-jun[†]

(School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430000, China)

Abstract: With the explosive growth of data volume, edge computing plays an increasingly important role in big data processing. In general, the data generated by real applications is modelled and represented as high-order incremental tensors. Recently, the incremental tensor tucker decomposition is deemed as an efficient approach to extract the information inherent in those high-order massive data. As the traditional incremental tensor decomposition usually ignores the influence of tensor model characteristics on the decomposition process, it is rather difficult for the decomposition results to preserve the overall characteristics of the original data. To address these issues, we propose an incremental tensor tucker decomposition method ITTDMC (incremental tensor tucker decomposition based on mode characteristics) based on mode characteristics. First, the update order of the increment factor matrix is determined by the increase of mode length, for reducing the reconstruction error caused by the update order. Next, the update weight of the incremental factor matrix is computed according to the changing ratio of the mode entropy, such that the decomposition results enable to maintain the characteristics of each module more accurately. Furthermore, the previous model characteristics and update parameters are recorded in a guide tensor. When the incremental data with similar model characteristics needs to be processed, the corresponding update parameters of the guide tensor can be directly employed to reduce the computational costs. Extensive experiment results on the synthetic and real data sets exhibit that the ITTDMC can greatly reduce the reconstruction error of incremental tensor (up to 29%) for those data sets with strong model characteristics.

Keywords: big data; incremental tensor; Tucker decomposition; mode characteristics; mode length increment; mode entropy

0 引言

张量分解作为具有强大数据压缩和信息挖掘能力的高阶数据处理方法^[1], 常用于边缘计算进行模型

压缩和相关性分析^[2], 通过优化数据处理算法在边缘设备性能受限时提高边缘计算的性能. 此外, 张量分解在信号处理^[3]、计算机视觉^[4]、图像压缩、知识发

收稿日期: 2023-07-18; 录用日期: 2023-11-21.

责任编辑: 邓庆绪.

[†]通讯作者. E-mail: jsonhan@hust.edu.cn.

现^[5]等领域也有较为广泛的应用。

在云边端协同环境中,边缘节点采集到的通常是海量的增量式数据^[6-8]。为减轻云端节点压力,可由边缘设备负责数据收集和张量分解的任务,将蕴含数据特征的分解结果传输给云端节点;云端节点负责根据张量分解结果进行数据分析以及数据共享。

由于张量分解结果保留原始数据特征的程度直接决定其应用价值,保留原始数据特征的程度越大,云端数据分析的结果可靠性越高。因此,云边端协同中增量式张量分解的优化目标应为提高分解结果保留原始数据特征程度,采用重构张量与原始张量的误差作为度量标准,误差越小,分解结果价值越高。

针对实时增长的张量数据,全量式张量分解会产生较大的存储成本和时间开销^[9]。增量式张量分解摒弃了全量式张量分解每次重新计算的弊端,利用前一时间点的张量分解结果和新增数据计算最新的分解结果^[10],降低了计算复杂度,极大地提高了时间效率。

最初对增量式张量的研究局限于流张量^[11-16],此类算法无法处理多模爆炸性增长的张量数据。文献[17-19]将增量式张量分解拓展到多模增长领域,但都面临误差较大的问题。文献[20]在指定数据集上准确率更高。文献[21]通过递归更新因子矩阵降低重构误差。这些改进算法都忽略了数据特征的影响,误差改进效果较差。另外,当前许多基于特征信息的张量分解算法存在较大的局限性。文献[22-23]仅针对全量式张量,文献[24]需要集成辅助生物信息,文献[25]需要数据集有少量显性数据。且基于数据特征的增量式张量分解方法忽略了过往时刻的数据特征,导致时间开销巨大。

本文的主要工作如下:

1) 提出一种基于模特征的增量式张量 Tucker 分解方法(incremental tensor tucker decomposition based on mode characteristics, ITTDMC)。该方法借助模特征指导增量式张量的分解过程,显著降低了重构误差,更有利于挖掘数据的隐藏信息。

2) 用模长增量决定增量因子矩阵更新顺序。增量因子矩阵的依次更新带来的重构误差不可避免,但可通过比较增量因子矩阵大小决定更新顺序,以此尽量降低此部分重构误差。

3) 利用模熵变化比决定增量因子矩阵更新权重。增量数据导致的模熵变化比能合理反映各模对分解结果的影响,以其作为增量因子矩阵的更新权重使每模更准确反映原始数据的特征。

4) 用指导张量记录过往时刻的模长增量和更新

权重。当出现模长增量相似的增量数据时直接复用指导张量中更新权重,减少重复计算。

1 理论基础

1.1 张量基本操作定义

本节主要介绍与本文相关的张量基本操作^[26-27]。

定义1(张量矩阵化) N 阶张量 $\chi \in \mathbb{R}^{I_1 \times \dots \times I_N}$, $I_n \in \mathbb{N}^*$ 表示 χ 在第 n 模上的长度。 χ 在模 n 上的矩阵化结果为 $X_{(n)}$, 二者元素一一对应。 $\chi(i_1, i_2, \dots, i_N)$ 模 n 展开后对应 $X_{(n)}$ 中的 $X_{(n)}(i_n, j)$, j 由 χ 中下标确定,具体计算如下:

$$j = \sum_{k=1, k \neq n}^N (i_k - 1)j_k + 1, j_k = \prod_{m=1, m \neq n}^{k-1} I_m. \quad (1)$$

定义2(张量矩阵模积) N 阶张量 $\chi \in \mathbb{R}^{I_1 \times \dots \times I_N}$, 矩阵 $A \in \mathbb{R}^{M \times I_n}$, χ 与 A 在第 n 模上的模态积 $y = \chi \times_n A$, $y \in \mathbb{R}^{I_1 \times \dots \times M \times \dots \times I_N}$. y 中元素具体计算如下:

$$y(i_1 \dots m \dots i_N) = \sum_{i_n=1}^{I_n} \chi(i_1 \dots i_n \dots i_N) A(m, i_n). \quad (2)$$

1.2 张量 Tucker 分解

Tucker 分解核心思想是将张量表示为核心张量和各模上因子矩阵乘积的形式。对于 N 阶张量 $\chi \in \mathbb{R}^{I_1 \times \dots \times I_N}$, 核心张量 $G \in \mathbb{R}^{R_1 \times \dots \times R_N}$ 代表不同模间的交互程度, $R_i \in \mathbb{N}^*$ 表示 χ 在模 i 上的秩, 因子矩阵 $U_i \in \mathbb{R}^{I_i \times R_i}$ 代表第 i 模上的主成分。 χ 的 Tucker 分解表示为

$$\chi \approx G \times_1 U_1 \cdots \times_N U_N = [G; U_1, \dots, U_N]. \quad (3)$$

2 基于模特征的增量式张量分解方法

基于模特征的增量式张量分解方法 ITTDMC 主要由两部分组成:提取模特征完善指导张量和用指导张量指导增量式张量分解。下面将介绍本算法需要提取的特征以及指导张量的作用,并分析模特征发挥作用的方式。

2.1 提取模特征完善指导张量

ITTDMC 采用模长增量和模熵变化比作为模特征指导增量式张量分解,将过往时刻的模特征记录在指导张量中以复用之前的计算结果。提取模特征的过程分为4个步骤:增量式子张量划分,计算模长增量,查找指导张量和完善指导张量。

2.1.1 增量式子张量划分

与时刻 $t-1$ 的 N 阶张量 χ^{t-1} 相比,时刻 t 的 N 阶张量 χ^t 可能在各个模都有增长。将 χ^t 沿着各模按 χ^{t-1} 的模长切分,图1给出了当 N 为3时对 χ^t 划分的

结果. 每个子张量都由一个长度为 N 的0、1序列组成的下标唯一标识, 下标全为0的子张量与 χ^{t-1} 相同, 剩余子张量为增量子张量.

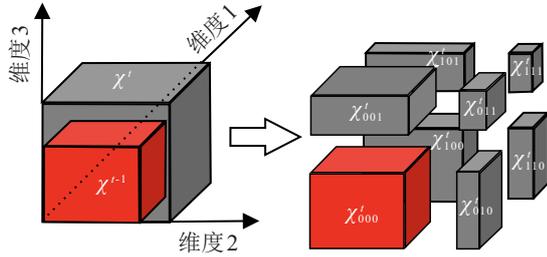


图1 3阶增量张量划分

根据下标中1的个数, 将增量子张量分为两类:

$$C_1 = \{\chi_{10\dots 0}^t, \chi_{01\dots 0}^t, \dots, \chi_{00\dots 1}^t\},$$

$$C_2 = \{\chi_{11\dots 0}^t, \chi_{10\dots 1}^t, \dots, \chi_{01\dots 1}^t, \chi_{11\dots 1}^t\}.$$

其中: C_1 中增量子张量下标中只有一个1, 此类增量子张量只与对应模增量因子矩阵的计算有关, 用于初始化增量因子矩阵; C_2 中增量子张量在多个模上均有增量, 用于更新增量因子矩阵. 用 sub_i 表示参与更新第 i 模增量因子矩阵的增量子张量集合, 有

$$sub_i = \{\chi_{0\dots 1\dots 0}^t, \dots, \chi_{1\dots 1\dots 1}^t\}.$$

不难发现 sub_i 中增量子张量下标第 i 位一定为1. C_2 中增量子张量出现在多个 sub_i 中, 说明 C_2 中增量子张量参与多个增量因子矩阵的更新.

2.1.2 计算模长增量

增量式张量各模的模长增量客观反映了该模的变化程度. 时刻 $t-1$ 与时刻 t 张量第 i 模的长度分别用 len_i^{t-1} 和 len_i^t 表示, 张量时刻 t 第 i 模的模长增量 $inclen_i^t$ 可由下式获得:

$$inclen_i^t = len_i^t - len_i^{t-1}. \quad (4)$$

2.1.3 查找指导张量

维护一个过往时刻张量各模模长增量和增量因子矩阵更新权重的3阶指导张量. 该张量的3个模分别为: 时间-特征参数-模. 计算完整特征参数前查找指导张量判断是否与过往某时刻增量数据相似.

对于 N 阶张量, 指导张量中下标为 (t_0, k, i) 的值表示时刻 t_0 第 k 个特征参数在第 i 个模上的值. 记录在指导张量中的特征参数共有 2^N 个: 1个模长增量, $2^N - 1$ 个增量子张量在更新增量因子矩阵时的更新权重. 指导张量在时间模针对某时刻 t_0 的切片是一个大小为 $2^N \times N$ 的二维矩阵, 表1展示了该矩阵对应的表结构, 第1列为特征参数名称, 第2行为模名

称, 其余部分与切片矩阵中值一一对应, 表示特征参数在某个模上的取值. 表中更新权重 $\alpha_{i_1 i_2 \dots i_N}^t$ 表示用 $\chi_{i_1 i_2 \dots i_N}^t$ 更新第 i 模增量因子矩阵时的权重.

表1 指导张量在时刻 t_0 的切片

特征参数	模1	模2	模 i	模 N
模长增量	$inclen_1^{t_0}$	$inclen_2^{t_0}$	$inclen_i^{t_0}$	$inclen_N^{t_0}$
$\chi_{10\dots 0}^t$ 对应更新权重	$\alpha_{10\dots 0}^1$	$\alpha_{10\dots 0}^2$	$\alpha_{10\dots 0}^i$	$\alpha_{10\dots 0}^N$
$\chi_{01\dots 0}^t$ 对应更新权重	$\alpha_{01\dots 0}^1$	$\alpha_{01\dots 0}^2$	$\alpha_{01\dots 0}^i$	$\alpha_{01\dots 0}^N$
$\chi_{i_1 \dots i_N}^t$ 对应更新权重	$\alpha_{i_1 \dots i_N}^1$	$\alpha_{i_1 \dots i_N}^2$	$\alpha_{i_1 \dots i_N}^i$	$\alpha_{i_1 \dots i_N}^N$
$\chi_{11\dots 1}^t$ 对应更新权重	$\alpha_{11\dots 1}^1$	$\alpha_{11\dots 1}^2$	$\alpha_{11\dots 1}^i$	$\alpha_{11\dots 1}^N$

模长增量反映增量数据对该模影响程度, 可近似认为模长增量相近时, 增量数据也相似. 为复用指导张量数据, 对每次增量数据按时间模遍历指导张量, 判断当前时刻模长增量 $inclen_i^t$ 是否与过往某时间点 t_0 模长增量 $inclen_i^{t_0}$ 相似. 判断标准如下:

$$(|inclen_i^t - inclen_i^{t_0}|)^2 \leq \theta (inclen_i^{t_0})^2, \quad (5)$$

其中 θ 为固定的模长增量相似因子. 如果两时刻在所有模上模长增量都满足式(5), 则认为它们的增量数据相似. 若存在时刻 t_0 与当前时刻的增量数据相似, 则直接使用时刻 t_0 的更新权重, 无需重新计算.

2.1.4 完善指导张量

若在2.1.3节中发现 $inclen_i^t$ 与之前任一时刻的模长增量都不相似, 则需将当前时刻的特征参数以二维参数矩阵形式添加到指导张量中.

ITDDMC采用模熵变化比作为计算更新权重的指标. 数据集中各模的熵变化值反应此模上值分布情况变化, 各模上熵变化值之比反映了不同模变化对最终分解结果影响的程度. 更新权重计算过程如下:

1) 计算各模熵变化量. 若时刻 t 数据集第 i 模上数据去重后集合为 S , 则张量该模上熵为

$$entropy_i^t = - \sum_{j=1}^{len_i^t} p_i^t(S_j) \log_2 p_i^t(S_j), \quad (6)$$

其中 $p_i^t(S_j)$ 表示 S 中的第 j 个数据在第 i 模上出现的频率. 模熵变化量为当前时刻张量第 i 模熵与前一时刻张量第 i 模熵差值的绝对值, 即

$$incentropy_i^t = |entropy_i^t - entropy_i^{t-1}|. \quad (7)$$

2) 计算各个增量子张量中模熵变化值之比. 对于在多个模上都有增量(下标中有不止一个1)的增量子张量, 采用模熵变化比表示其在不同增量因子矩阵的计算过程中发挥的不同作用. 对于 C_2 中的增量子张量 $\chi_{i_1 i_2 \dots i_N}^t$, 其在第 j 模上的熵变化比为

$$\psi_{i_1 \dots i_N}^j = \frac{\text{incentropy}_j^t \times i_j}{\sum_{k=1}^N \text{incentropy}_{i_k}^t \times i_k}. \quad (8)$$

其中: $\psi_{i_1 \dots i_N}^j$ 表示 \mathbb{C}_2 中 $\chi_{i_1 \dots i_N}^t$ 在第 j 模上的熵变化比; $\psi_{i_1 \dots i_N}^j$ 分母是增量子张量下标中所有值为1的模对应的模熵变化值之和,若增量子张量下标在第 i 模上值为1,则分子是第 i 模的熵变化值,否则为0.

3) 计算增量子张量更新增量因子矩阵时的权重. 采用不同增量子张量针对某个模的模熵变化比比值作为更新对应模增量因子矩阵的权重. 更新第 j 模增量因子矩阵时需使用 \mathbb{C}_1 中第 j 个增量子张量 $\chi_{0 \dots 1 \dots 0}^t$ 和 sub_j 中所有增量子张量,这些增量子张量更新增量因子矩阵时的更新权重计算方式如下:

$$\alpha_{0 \dots 1 \dots 0}^j = \frac{1}{1 + \sum_{(i_1 \dots i_N) \in \phi_j} \psi_{i_1 \dots i_N}^j},$$

$$\alpha_{i_1 \dots i_N}^j = \frac{\psi_{i_1 \dots i_N}^j}{1 + \sum_{(i_1 \dots i_N) \in \phi_j} \psi_{i_1 \dots i_N}^j}. \quad (9)$$

其中: ϕ_j 表示 sub_j 中所有增量子张量的 N 维下标集合; $\alpha_{0 \dots 1 \dots 0}^j$ 表示 \mathbb{C}_1 中只在第 j 模上增长的增量子张量在更新第 j 模的增量因子矩阵时所占的权重; $\alpha_{i_1 \dots i_N}^j$ 表示 sub_j 中的增量子张量所得到的增量因子矩阵在更新第 j 模的增量因子矩阵时所占的权重. 由式(8)可得,若 $\chi_{0 \dots 1 \dots 0}^t$ 表示只在第 j 模上增长的增量子张量,则 $\psi_{0 \dots 1 \dots 0}^j$ 的值为1,式(9)分子分母中1由此而来. 得出更新权重后,将更新权重与2.1.2节中的模长增量组成二维矩阵添加到指导张量中.

2.2 指导张量指导增量式张量分解

得到指导张量后,根据指导张量中特征参数指导增量式张量分解. 此过程主要分为初始化增量因子矩阵、更新增量因子矩阵、合并增量因子矩阵和计算核心张量4个步骤.

2.2.1 初始化增量因子矩阵

用 \mathbb{C}_1 初始化增量因子矩阵 U_i' , 计算如下:

$$U_i' = (\chi_{0 \dots 1 \dots 0}^t)_{(i)} ((U_N^{t-1})^T \otimes \dots \otimes (U_{i+1}^{t-1})^T \otimes (U_{i-1}^{t-1})^T \otimes \dots \otimes (U_1^{t-1})^T) (G_{(i)}^{t-1})^\dagger, \quad (10)$$

其中 $\chi_{0 \dots 1 \dots 0}^t$ 表示 \mathbb{C}_1 中只在第 i 模上有增长的增量子张量. 由式(10)可得,初始化 U_i' 的过程中 U_i' 彼此间互不影响,此过程与顺序无关.

2.2.2 更新增量因子矩阵

用 \mathbb{C}_2 更新 U_i' 以得到完整增量因子矩阵. 在更新 U_i' 时,需固定其他增量因子矩阵,用 sub_i 中增量子张

量、其他增量因子矩阵和上一时刻因子矩阵 U_i^{t-1} 完成 U_i' 的更新. 由于更新后的 U_i' 参与其他增量因子矩阵更新过程,必须考虑增量因子矩阵更新顺序.

U_i' 带来的重构误差与其所包含数据量的大小正相关,为减缓重构误差累积速度,应按照 U_i' 包含数据量大小的非递减顺序更新 U_i' . U_i' 列固定,行数为 inclen_i^t , U_i' 包含数据量大小 v_i^t 的计算如下:

$$v_i^t = \text{inclen}_i^t \times R_i. \quad (11)$$

其中: R_i 是模 i 因子矩阵列数,也是核心张量在第 i 模的长度,此值在运行之初确定,保持不变. 假设 v_i^t 的大小关系为 $v_1^t < \dots < v_i^t < \dots < v_N^t$, 则增量因子矩阵更新顺序为 $U_1', \dots, U_i', \dots, U_N'$.

按照更新顺序依次更新增量因子矩阵. 更新 U_i' 时只需用到在第 i 模上有增量的增量子张量,即 sub_i 中增量子张量. 对于 sub_i 中每一增量子张量,计算其用于更新 U_i' 的增量因子矩阵,计算方法如下

$$U_j'' = \begin{cases} U_j^{t-1}, & \chi_{i_1 \dots i_N}^t \in \text{sub}_j; \\ U_j', & \text{otherwise.} \end{cases}$$

$$U_{i_1 \dots i_N, i}' = (\chi_{i_1 \dots i_N}^t)_{(i)} \{ (U_N'')^T \otimes \dots \otimes (U_{i+1}'')^T \otimes (U_{i-1}'')^T \otimes \dots \otimes (U_1'')^T \} (G_{(i)}^{t-1})^\dagger. \quad (12)$$

其中: $U_{i_1 \dots i_N, i}'$ 为 $\chi_{i_1 \dots i_N}^t$ 用于更新 U_i' 的增量因子矩阵, $\chi_{i_1 \dots i_N}^t \in \text{sub}_i$.

得到所有用于更新 U_i' 的增量因子矩阵后,根据指导张量中对应权重更新 U_i' , 更新公式如下:

$$U_i' = \alpha_{0 \dots 1 \dots 0}^i U_i' + \sum_{(i_1 \dots i_N) \in \phi_i} \alpha_{i_1 \dots i_N}^i U_{i_1 \dots i_N, i}'. \quad (13)$$

其中: ϕ_i 表示 sub_i 中所有增量子张量的 N 维下标集合; $\alpha_{0 \dots 1 \dots 0}^i$, $\alpha_{i_1 \dots i_N}^i$ 分别表示初始化的 U_i' 和 $U_{i_1 \dots i_N, i}'$ 在最终增量因子矩阵中所占权重,二者均可由指导张量在相似时刻的切片矩阵或2.1.4节中计算结果获得. 其他增量因子矩阵更新过程与 U_i' 类似.

2.2.3 合并因子矩阵

当前时刻的因子矩阵 U_i^t 由 U_i^{t-1} 和 U_i' 按列拼接组成. 但此时 U_i^t 可能不满足酉矩阵的形式,需对其进行 MGS 处理得到列正交的因子酉矩阵.

2.2.4 计算核心张量

将式(2)优化为各个子张量与部分因子矩阵转置模乘的形式. 按照 U_i^{t-1} 和 U_i' 的行数将列正交的 U_i^t 按行分为 $U_{i,0}^t$ 和 $U_{i,1}^t$. 核心张量的计算过程如下:

$$G^t =$$

$$\sum_{(i_1 \dots i_N) \in \phi} \chi_{i_1 \dots i_N}^t \times_1 (U_{1,i_1}^t)^\top \dots \times_N (U_{N,i_N}^t)^\top = G^{t-1} \times_1 \{(U_{1,0}^t)^\top U_1^{t-1}\} \dots \times_N \{(U_{N,0}^t)^\top U_N^{t-1}\} + \sum_{(i_1 \dots i_N) \in \phi'} \chi_{i_1 \dots i_N}^t \times_1 (U_{1,i_1}^t)^\top \dots \times_N (U_{N,i_N}^t)^\top. \quad (14)$$

其中: ϕ 表示由0,1组成的所有 N 维下标集合, ϕ' 表示 ϕ 中去除值全为0的下标后剩余下标集合.

3 实验分析

本章对 ITTDMC 算法进行实验测试, 并与已有的增量式张量分解方法对比.

3.1 实验设置

实验的相关设置包括实验环境、数据集、比较算法和评价算法的性能指标.

3.1.1 实验环境

实验程序由 python3.8 编写. 实验测试环境为一台浪潮英信服务器, 服务器的配置为: 英特尔至强 CPU 处理器 Silver 4210R、32GB 内存和 2.2T 硬盘.

3.1.2 数据集

用于实验的数据集具体情况如表2所示.

表2 实验测试数据集统计信息

数据集	各模模长	初始时刻各模模长	初始时刻张量秩
Ersatz	(158, 328, 400)	(76, 180, 00)	(3, 3, 3)
MovieLens	(370, 1388, 1000)	(183, 583, 500)	(38, 197, 204)
Air Quality	(182, 327, 253, 998)	(84, 177, 122, 500)	(34, 64, 42, 170)
Book	(1257, 660, 800)	(1249, 653, 400)	(347, 134, 129)

3.1.3 对比算法

1) eOTD^[19]: 增量式张量 Tucker 分解通用方法, 按照固定的顺序和比例更新增量因子矩阵.

2) RI-NTD^[21]: 使用递归策略更新因子矩阵, 通过牺牲时间降低了一定的重构误差.

3) ITTDMC-OO: 只改变模更新顺序的 ITTDMC 算法, 用于验证 ITTDMC 在增量因子矩阵更新顺序和更新权重两方面的修改对降低重构误差均有效果.

3.1.4 性能指标

1) 重构误差^[19]. 张量分解是 NP-hard 问题^[21], 求其最优解较为困难且现实意义不大. 采用重构误差作为度量标准以得到次优解, 不仅计算方便且更能反应与原始数据相似度. 时刻 t 的重构张量 $\hat{\chi}^t$ 可由式 (2) 得到. 重构误差 $\text{error}(\hat{\chi}^t, \chi^t)$ 表示 $\hat{\chi}^t$ 与时刻 t 真实张量 χ^t 之间的误差, 误差越小, 分解结果越能保留原始数据信息, $\text{error}(\hat{\chi}^t, \chi^t)$ 计算方式如下:

$$\text{error}(\hat{\chi}^t, \chi^t) = \frac{\|\chi^t - \hat{\chi}^t\|_F^2}{\|\chi^t\|_F^2}. \quad (15)$$

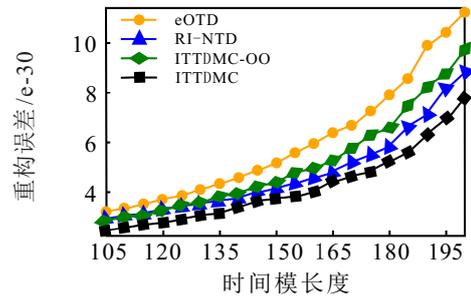
2) 运行时间. 根据增量数据和前一时刻分解结果获得当前时刻分解结果所用时间. 运行时间越小越能满足实时性要求.

3.2 实验结果及分析

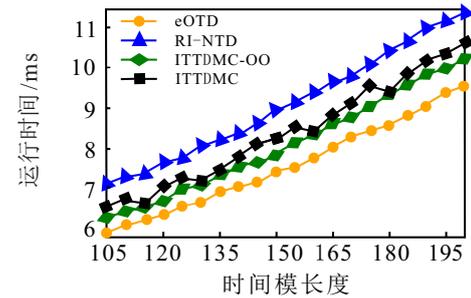
本节将展示并分析 4 种算法在不同数据集上的测试结果.

3.2.1 合成数据集

图2给出了 4 种算法在合成数据集 Ersatz 上随着时间模模长的增长重构误差和运行时间的变化情况.



(a) 重构误差随时间模长度的变化



(b) 运行时间随时间模长度的变化

图2 合成数据集 Ersatz 上的实验结果

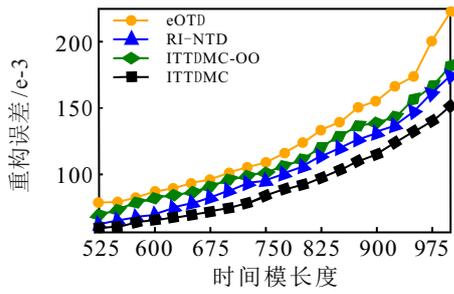
综合分析图2可得: Ersatz 数据集每次增量数据在各模上均有明显增长, 模长增量和模熵变化值能较好反映模特征变化情况, ITTDMC-OO 和 ITTDMC 误差降低效果明显. 由图2(b)可知, 当出现相似增量数据时, 无需重复计算更新权重, ITTDMC 的运行时间降至与 ITTDMC-OO 相差无几.

Ersatz 数据集上实验结果表明, 当模特征变化明显时, 改变更新顺序和更新权重均效果较好. ITTDMC 在消耗较少时间的同时降低了更多重构误差.

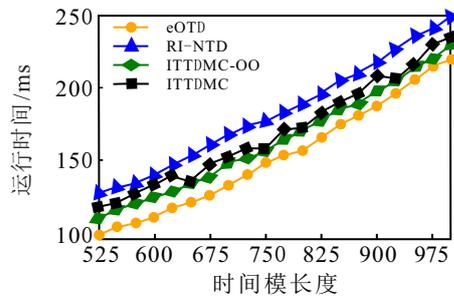
3.2.2 真实数据集

在真实数据集上进行测试分析. 图3和图4分别展示了 MovieLens、Air Quality 数据集上的实验结果.

由图3和图4可知, 与合成数据集相比, 真实数据集模长更长, 秩更大, 各模间相关性较弱, 所以重构误

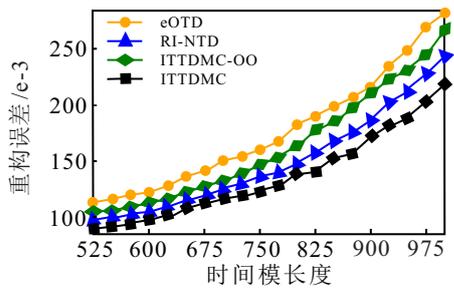


(a) 重构误差随时间模长度的变化

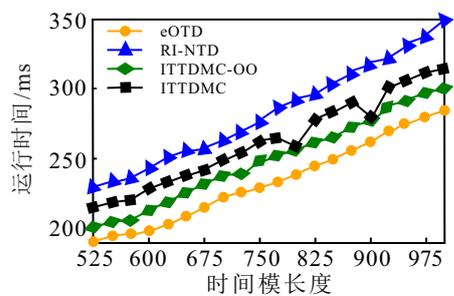


(b) 运行时间随时间模长度的变化

图3 MovieLens数据集上实验结果



(a) 重构误差随时间模长度的变化



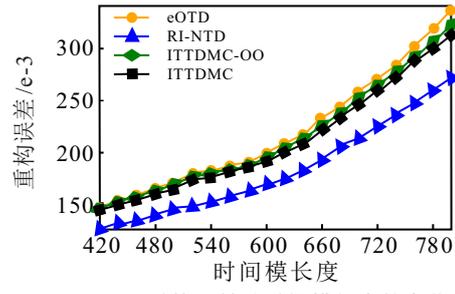
(b) 运行时间随时间模长度的变化

图4 Air Quality数据集上实验结果

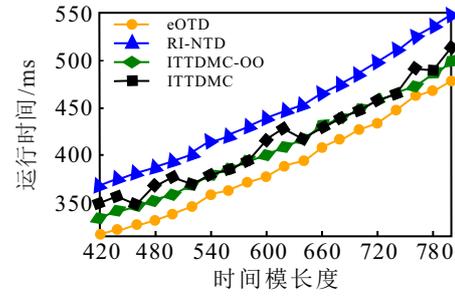
差较大,运行时间较长.

对比图3、图4可得: Air Quality和MovieLens每次增量数据带来的模特征变化较为显著, ITTDMC-OO和ITTDMC明显降低了重构误差; Air Quality作为四模张量数据,相似模长增量的判断条件更多,使用指导张量中过往时刻参数的概率降低.

引入Book数据集对比分析算法在模长增量较小数据集上的性能,图5为Book数据集上实验结果.分析图5可得:Book数据集增量数据只在时间模有明显增长,从增量数据中提取的模特征意义不大,ITTDMC-OO和ITTDMC降低误差的效果较差.



(a) 重构误差随时间模长度的变化



(b) 运行时间随时间模长度的变化

图5 Book数据集上实验结果

真实数据集上的实验结果表明:模特征变化较大时,ITTDMC-OO性能较好;模特征变化较小时,ITTDMC-OO虽然牺牲了更多的时间但误差降低效果微弱.

4 结论

本文针对传统增量式张量分解误差较大的问题,提出了一种基于张量模特征的增量式张量Tucker分解方法ITTDMC.该算法提取模特征指导增量式张量的解过程,分别用模长增量和模熵变化比指导增量因子矩阵更新顺序和更新权重.为避免模特征相似时的重复计算,用指导张量存储过往时刻模长增量和更新权重,若模长增量与指导张量中过往某时刻相似,则复用更新权重,提高时间效率;实验结果表明,当模特征变化较大时,ITTDMC通过牺牲少量时间可显著降低重构误差.未来工作中,将重点关注如何在模特征变化微弱时降低重构误差.

参考文献(References)

- [1] Chang S Y, Wu H C. Tensor Kalman filter and its applications[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(6): 6435-6448.
- [2] Wang X K, Yang L T, Ren L, et al. A tensor-based computing and optimization model for intelligent edge services[J]. IEEE Network, 2022, 36(1): 40-44.
- [3] Aidini A, Tsagkatakis G, Tsakalides P. Tensor decomposition learning for compression of multidimensional signals[J]. IEEE Journal of Selected Topics in Signal Processing, 2021, 15(3): 476-490.
- [4] Minh H P, Xuan N N, Thai S T. TT-ViT: Vision transformer compression using tensor-train

- decomposition[C]. Computational Collective Intelligence. Hammamet, 2022: 755-767.
- [5] Jia T, Yang Y X, Lu X, et al. Link prediction based on tensor decomposition for the knowledge graph of COVID-19 antiviral drug[J]. Data Intelligence, 2022, 4(1): 134-148.
- [6] 朴勇, 江贺, 王秀坤. 基于张量的XML相似度计算方法[J]. 控制与决策, 2016, 31(9): 1711-1714. (Piao Y, Jiang H, Wang X K. Tensor-based approach to XML similarity calculation[J]. Control and Decision, 2016, 31(9): 1711-1714.)
- [7] 代伟, 南静. 随机神经网络增量构造学习方法研究进展[J]. 控制与决策, 2023, 38(8): 2231-2242. (Dai W, Nan J. Recent advances in incremental learning methods for random weight neural network[J]. Control and Decision, 2023, 38(8): 2231-2242.)
- [8] 苑红星, 卓雪雪, 竺德, 等. 基于矩阵的混合型邻域决策粗糙集增量式更新算法[J]. 控制与决策, 2022, 37(6): 1621-1631. (Yuan H X, Zhuo X X, Zhu D, et al. Incremental updating algorithms of neighborhood decision-theoretic rough set model for hybrid data based on matrix[J]. Control and Decision, 2022, 37(6): 1621-1631.)
- [9] Ahn D, Jang J G, Kang U. Time-aware tensor decomposition for sparse tensors[C]. Proceedings of the 8th International Conference on Data Science and Advanced Analytics. Porto, 2021: 1-2.
- [10] Song Q Q, Huang X, Ge H C, et al. Multi-aspect streaming tensor completion[C]. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Halifax, 2017: 435-443.
- [11] 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. San Francisco, 2016: 1375-384.
- [12] Sun J M, Tao D C, Papadimitriou S, et al. Incremental tensor analysis: Theory and applications[J]. ACM Transactions on Knowledge Discovery from Data, 2008, 2(3): 1-37.
- [13] Yu R, Cheng D H, Liu Y. Accelerated online low-rank tensor learning for multivariate spatio-temporal streams[C]. Proceedings of the 32nd International Conference on International Conference on Machine Learning. Lille, 2015: 238-247.
- [14] Wang X K, Wang W, Yang L T, et al. A distributed HOSVD method with its incremental computation for big data in cyber-physical-social systems[J]. IEEE Transactions on Computational Social Systems, 2018, 5(2): 481-492.
- [15] Zeng C, Ng M K. Incremental CP tensor decomposition by alternating minimization method[J]. SIAM Journal on Matrix Analysis and Applications, 2021, 42(2): 832-858.
- [16] 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.
- [17] Wang P M, Yang L T, Qian G W, et al. HO-OTSVD: A novel tensor decomposition and its incremental decomposition for cyber-physical-social networks (CPSN)[J]. IEEE Transactions on Network Science and Engineering, 2020, 7(2): 713-725.
- [18] Yang K Y, Gao Y J, Shen Y F, et al. DisMASTD: An efficient distributed multi-aspect streaming tensor decomposition[C]. Proceedings of the 37th International Conference on Data Engineering (ICDE). Athens, 2021: 1080-1091.
- [19] Xiao H P, Wang F, Ma F L, et al. eOTD: An efficient online tucker decomposition for higher order tensors[C]. 2018 IEEE International Conference on Data Mining. Singapore, 2018: 1326-1331.
- [20] Biswas D, Azharul Hasan K M, Zarnaz Z. Progressive recommendation by incremental tensor factorization[C]. The 25th International Conference on Computer and Information Technology. Cox's Bazar, 2022: 483-488.
- [21] Zdunek R, Fona K. Incremental nonnegative tucker decomposition with block-coordinate descent and recursive approaches[J]. Symmetry, 2022, 14(1): 113-120.
- [22] Li M L, Candan K S, Sapino M L. GTT: Guiding the tensor train decomposition[C]. Similarity Search and Applications. Cham: Springer International Publishing, 2020: 187-202.
- [23] Li M L, Candan K S, Sapino M L. GTT: Leveraging data characteristics for guiding the tensor train decomposition[J]. Information Systems, 2022, 108: 102047.
- [24] Huang F, Yue X, Xiong Z K, et al. Tensor decomposition with relational constraints for predicting multiple types of microRNA-disease associations[J]. Briefings in Bioinformatics, 2021, 22(3): 140-148.
- [25] Zhao J L, Zheng S D, Huo H, et al. Fast weighted CP decomposition for context-aware recommendation with explicit and implicit feedback[J]. Expert Systems with Applications, 2022, 204: 117591.
- [26] Mørup M. Applications of tensor(multiway array) factorizations and decompositions in data mining[J]. WIREs Data Mining and Knowledge Discovery, 2011, 1(1): 24-40.
- [27] Bernardi A, Brachat J, Comon P, et al. General tensor decomposition, moment matrices and applications[J]. Journal of Symbolic Computation, 2013, 52: 51-71.

作者简介

渠超洋(1999—), 男, 硕士生, 从事大数据处理的研究, E-mail: m202173616@hust.edu.cn;

韩建军(1972—), 男, 教授, 博士生导师, 从事并行与分布式计算、嵌入式实时系统等研究, E-mail: jasonhan@hust.edu.cn.