

基于高阶累积量张量分解的联合盲源分离算法

季策, 刘明欣

(东北大学 计算机科学与工程学院, 辽宁 沈阳 110169)

摘要: 提出一种基于高阶累积量张量分解的联合盲源分离(JBSS)算法, 该算法可以从多组数据集的观测信号中恢复出源信号。首先通过计算多组数据集观测信号的高阶互累积量张量, 利用累积量张量潜在的对角结构, 将JBSS问题转化为高阶张量CP分解(CPD)问题。接下来, 通过张量列分解(TTD)将高阶张量分解为由不高于3阶的多个互连的核张量组成的简单张量网络, 由此将高阶CPD问题转化为多个3阶CPD问题。最后, 根据TTD与CPD之间的关系, 在多次3阶CPD之后, 通过依次对因子矩阵进行重新排序与缩放得到多数据集的混合矩阵, 进而实现对源信号的分离。实验结果表明, 该算法具有较快的运行速度。

关键词: 联合盲源分离; 张量列分解; CP分解; 高阶累积量

中图分类号: TN 911.7

文献标志码: A

文章编号: 1005-3026(2024)01-0026-07

Joint Blind Source Separation Algorithm Based on Decomposition of Higher-Order Cumulant Tensors

Ji Ce, Liu Ming-xin

(School of Computer Science & Engineering, Northeastern University, Shenyang 110169, China. Corresponding author: LIU Ming-xin, E-mail: dqrlmx@163.com)

Abstract: A joint blind source separation (JBSS) algorithm based on decomposition of high-order cumulant tensors is proposed. The algorithm recovers the source signals from the observation signals of multiset data. Firstly, the higher-order cross-cumulant tensors of observation signals of multiset data are calculated. Due to the potential diagonal structures of cumulant tensors, the JBSS problem can be transformed into canonical polyadic decomposition (CPD) of a higher-order tensor. Next, by tensor train decomposition (TTD), the higher-order tensor is decomposed into a simple tensor network composed of a set of interconnected core tensors of orders not higher than 3. The CPD of a higher-order tensor thereby is transformed into a set of CPDs of order-3 tensors. Finally, according to the links between TTD and CPD, after several CPDs of order-3 tensors, the mixed matrices of multi-dataset can be obtained by reordering and rescaling the factor matrices sequentially, resulting in the separation of the source signals. Simulation results show that the proposed algorithm operated at a faster speed.

Key words: joint blind source separation; tensor train decomposition; canonical polyadic decomposition; higher-order cumulant

有关同一目标或物理现象的信息可以用不同类型的仪器、测量技术, 在不同的条件下, 通过多个实验或对象获得。每一种信息来源或形式, 可称为一种模态, 并与一个数据集相关联。由于客观现象的丰富特征, 仅通过单一模态很难对其

进行完整地了解。因此, 对多个数据集进行融合分析的多模态方法得到了发展, 并应用于生物识别、人机交互、脑科学、医疗诊断、遥感等领域。传统的盲源分离是针对单数据集信号进行分析。联合盲源分离(joint blind source separation, JBSS)

收稿日期: 2022-08-02

基金项目: 中央高校基本科研业务费专项资金资助项目(N2116015, N2116020)。

作者简介: 季策(1969-), 女, 辽宁沈阳人, 东北大学副教授。

是一种多模态融合方法,在已知多数据集观测信号的前提下,对多个数据集的混合矩阵和源信号进行辨识.每一种模态都给整体带来了新的信息,数据集之间的关联本身也是新的信息.这些新的信息在分析模型中引入了新的约束,从而增强了模型的唯一性、可解释性和稳健性^[1].

以往的 JBSS 方法多是基于矩阵分析的算法,近些年,基于张量分解的 JBSS 算法得到了发展^[2].张量是多维数组,相较矩阵算法,张量算法可以更好地利用数据的高维信息^[3-4].与矩阵分解相比,张量分解的唯一性条件更宽松,即使是在源信号多于传感器的欠定情况也有可能满足唯一分解的条件.在 JBSS 问题中,多个数据集是面向同一现象的多组观测值,张量分解可以提供一个自然的框架,将观测的多样性对应于张量的结构.因此,高阶张量可以被视为联合分析同一现象的多组观测值的工具^[5].

通过低维数据的重排,计算高阶统计量以及将时域、空域、频域等的数据自然地叠加可以从低维原始数据创建高阶数据张量,通过张量分解可以将高阶张量表示为其各个分量的数学运算的形式^[6-7].通常的张量 CP 分解(canonical polyadic decomposition, CPD)算法,其时间复杂度会随着张量阶数呈指数增长. Phan 等和 Zniyed 等^[7-8]利用文献[9]中介绍的张量列分解(tensor train decomposition, TTD)算法,将一个大规模的高阶 CPD 问题转化为多个小规模 3 阶 CPD 问题以降低计算成本.目前,CPD 和 TTD 算法多应用于特征降维、图像补全、神经网络的模型压缩以及量子物理等领域^[4,7].

鉴于以上分析,本文提出了一种基于高阶累积量张量分解(TT-CPD)的算法,将 TTD 算法应用于 JBSS 问题中.首先,通过计算多数据集观测信号的高阶互累积量将 JBSS 问题转化为高阶张量的 CPD 问题;接下来通过 TTD 对原始张量进行预压缩;之后利用 TTD 与 CPD 的对应关系估计出多数据集的混合矩阵,进而恢复出源信号.相比于文献[7-8],本文利用不相同的置换矩阵论证了对各个因子矩阵进行重新排序的过程.另一方面,相较基于 Givens 旋转的一类算法^[10-11],本文的算法不需要限定混合矩阵为正交矩阵,省去了预白化的过程.此外,对称张量算法只适用于单数据集的盲源分离问题^[12],而本文的算法是针对非对称张量做分解,因而适用于多数据集的混合矩阵估计.

1 问题模型

多数据集 JBSS 问题的模型如下:

$$\mathbf{x}^{(m)}(t) = \mathbf{A}^{(m)}\mathbf{s}^{(m)}(t) + \mathbf{n}^{(m)}(t). \quad (1)$$

式中: $m = 1, 2, \dots, M$; $\mathbf{x}^{(m)}(t) \in \mathbf{R}^{I_m}$ 为第 m 个数据集在 t 时刻的 I_m 个观测信号向量,是受噪声 $\mathbf{n}^{(m)}(t)$ 干扰的源信号的瞬时线性混合; $\mathbf{s}^{(m)}(t) \in \mathbf{R}^R$ 为第 m 个数据集在 t 时刻的 R 个源信号向量; $\mathbf{A}^{(m)} \in \mathbf{R}^{I_m \times R}$ 表示第 m 个数据集的混合矩阵.在 JBSS 中通常预先假设源信号为非高斯、非平稳信号,且源信号具有组内独立性和组间相关性,即当 $a \neq b$ 时, $s_a^{(m)}(t)$ 和 $s_b^{(m)}(t)$ 统计独立,当 $a = b$ 时, $s_a^{(m)}(t)$ 和 $s_b^{(m)}(t)$ 统计相关^[2].

盲源分离问题中存在排序模糊,可以由置换矩阵 $\mathbf{\Pi}$ 来表示.在 JBSS 问题中,多数据集信号往往是同一目标经由不同观测方式所得,存在组间相关性.所以分离出的多组源信号的排序应该是对齐的,即表示排序模糊的置换矩阵 $\mathbf{\Pi}$ 应当是相同的.

理论推导时,先忽略噪声项,计算 K 组 M 个数据集观测信号的高阶互累积量张量,可得到分解结构^[13-14]:

$$\begin{aligned} \mathbf{X}^{(k)} = \text{Cum} & \left[\mathbf{x}^{(1)}(t_k), \mathbf{x}^{(2)}(t_k), \dots, \mathbf{x}^{(M)}(t_k) \right] = \\ & \text{Cum} \left[\mathbf{s}^{(1)}(t_k), \mathbf{s}^{(2)}(t_k), \dots, \mathbf{s}^{(M)}(t_k) \right] \times \\ & \mathbf{A}^{(1)} \times \mathbf{A}^{(2)} \times \dots \times \mathbf{A}^{(M)} = \\ & \mathbf{D}^{(k)} \times \mathbf{A}^{(1)} \times \mathbf{A}^{(2)} \times \dots \times \mathbf{A}^{(M)}. \end{aligned} \quad (2)$$

式中: $k = 1, 2, \dots, K$; $\text{Cum}[\cdot]$ 表示计算累积量; $\mathbf{X}^{(k)} \in \mathbf{R}^{I_1 \times I_2 \times \dots \times I_M}$ 为 M 阶观测信号张量; $\mathbf{D}^{(k)} \in \mathbf{R}^{R \times R \times \dots \times R}$ 为 M 阶源信号张量.由 JBSS 的先验假设可得 $\mathbf{D}^{(k)}$ 为对角张量,即当且仅当 $r_1 = r_2 = \dots = r_M$ 时,有 $\mathbf{D}^{(k)}(r_1, r_2, \dots, r_M) \neq 0$. \times_m 表示张量的模式 m 积,定义为

$$(\mathbf{X} \times_m \mathbf{A})_{i_1, i_2, \dots, i_{m-1}, j, i_{m+1}, \dots, i_N} = \sum_{i_m=1}^{I_m} \mathbf{x}_{i_1, i_2, \dots, i_m, \dots, i_N} \mathbf{a}_{j, i_m}. \quad (3)$$

K 组 M 阶观测信号张量 $\mathbf{X}^{(k)}$ 按顺序堆叠后形成的 $M+1$ 阶张量 $\tilde{\mathbf{X}}$,其 CPD 结构为

$$\tilde{\mathbf{X}} = [\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \dots, \mathbf{A}^{(M)}, \mathbf{A}^{(M+1)}] = \mathbf{I}_{M+1, R} \times \mathbf{A}^{(1)} \times \mathbf{A}^{(2)} \times \dots \times \mathbf{A}^{(M+1)}. \quad (4)$$

$\tilde{\mathbf{X}}$ 中的元素可以表示为

$$\tilde{\mathbf{X}}(i_1, i_2, \dots, i_M, k) = \sum_{r=1}^R \mathbf{A}^{(1)}(i_1, r) \mathbf{A}^{(2)}(i_2, r) \dots \mathbf{A}^{(M)}(i_M, r) \mathbf{A}^{(M+1)}(k, r). \quad (5)$$

其中: $\mathbf{I}_{M+1, R}$ 表示尺寸为 R 的 $M+1$ 阶单位张量; $\mathbf{A}^{(M+1)}$ 有 K 行,每一行对应着 $\mathbf{D}^{(k)}$ 的主对角元素,即 $\mathbf{D}^{(k)}(r, r, \dots, r) = \mathbf{A}^{(M+1)}(k, r)$.混合矩阵 \mathbf{A} 的列数,即

源信号个数对应着张量秩 R . 式(4)表明只要对已获取的 $M+1$ 阶张量 \tilde{X} 做秩为 R 的 CPD, 求出的前 M 个因子矩阵即是 M 个数据集的混合矩阵. 由于 CPD 在秩超过各个模式的维数, 即 $R > \max(I_1, I_2, \dots, I_M, I_{M+1})$ 的情况下也有可能满足唯一性条件, 所以此算法也可应用于方程组欠定, 即源的数量大于传感器数量的情形. 此外, 因子矩阵 $A^{(1)}, A^{(2)}, \dots, A^{(M+1)}$ 不必是列满秩^[1].

为避免直接进行 CPD 时算法的时间复杂度随张量阶数呈指数增长, 本文的算法首先通过 TTD 对原始张量进行预压缩以降低计算成本, 之后使用一个从核张量到因子矩阵的精确转换方法来估计 CPD 的因子矩阵. TTD 这一步骤可以通过 TT 秩为 (R, R, \dots, R) 的 TT-SVD 算法实现, 对于 N 阶张量, 该算法需要执行 $N-1$ 次截断奇异值分解去估计 N 个核张量^[15-16].

2 采取顺序调整方案的 TT-CPD 算法

2.1 TTD 与 CPD 的对应关系

假设秩为 R 的 N 阶张量 Y 存在本质上唯一的 CPD, 同时 Y 也可以通过 TTD 被表示成 TT 秩为 (R, R, \dots, R) 的形式, 如式(6)所示. 张量表达式也可以表示为张量图的形式^[3], 如图 1 所示.

$$Y = [A^{(1)}, A^{(2)}, \dots, A^{(N)}] = G_1 \cdot G_2 \cdots G_{N-1} \cdot G_N. \quad (6)$$

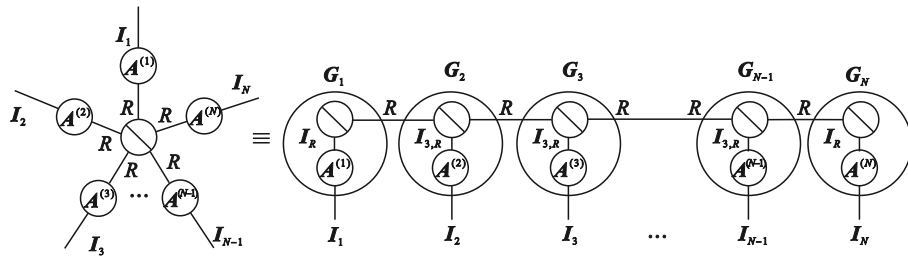


图 1 张量的 CPD 与 TTD

Fig. 1 CPD and TTD of tensor

当 $(2 \leq n \leq N-1)$ 时, 3 阶核张量 G_n 可以进行张量秩为 R 的 3 阶 CPD, 其第 2 个因子矩阵对应着 $A^{(n)} = \tilde{G}_n^T Q_{N-1}^{-T}$, 由此将 TTD 与 CPD 建立了联系^[3,8](图 2).

将 N 阶张量的 CPD 写成模式积的形式:

$$Y = I_{N,R} \times_1 A^{(1)} \times_2 A^{(2)} \times \cdots \times_N A^{(N)}. \quad (7)$$

可通过 TTD 将尺寸为 R 的 N 阶单位张量 $I_{N,R}$ 分解为互连的单位矩阵 I_R 与 3 阶单位张量 $I_{3,R}$, 用张量列缩并的形式表示为

$$I_{N,R} = I_R \cdot I_{3,R} \cdot I_{3,R} \cdots I_R. \quad (8)$$

将其代入式(7)并与式(6)对比可得

$$\left. \begin{aligned} G_1 &= A^{(1)}; \\ G_n &= I_{3,R} \times_2 A^{(n)}, 2 \leq n \leq N-1; \\ G_N &= A^{(N)T}. \end{aligned} \right\} \quad (9)$$

此处的核张量 G_n 是稀疏的, 其第 i 个垂直切片矩阵是对角矩阵, 对角元素为 $A^{(n)}$ 的第 i 行, 即

$$G_n(:, i, :) = \text{Diag}(A^{(n)}(i, :)). \quad (10)$$

图 2 CPD 与 TTD 的对应关系

Fig. 2 Links between TTD and CPD

在实际进行 TTD 运算时, 所得的 TT 核张量并不一定具有对角结构. 为得到通用的 TT 张量结构, 需要考虑到张量 TTD 的非唯一性. 令 Q_n 为任意 $R \times R$ 的可逆矩阵, 使 G_n 后乘 Q_n 且 G_{n+1} 前乘 Q_n^{-1} , 这改变了核张量, 但张量的 TTD 保持不变^[7]. 如式(11)所示:

$$\begin{aligned} Y &= G_1 \cdot G_2 \cdots G_{N-1} \cdot G_N = \\ &G_1 \cdot Q_1 \cdot Q_1^{-1} \cdot G_2 \cdot Q_2 \cdots Q_{N-1}^{-1} \cdot G_N \cdot Q_N \cdot \\ &Q_N^{-1} \cdot G_{N+1} \cdot Q_{N+1} \cdots Q_{N-1}^{-1} \cdot G_N = \\ &\tilde{G}_1 \cdot \tilde{G}_2 \cdots \tilde{G}_{N-1} \cdot \tilde{G}_N. \end{aligned} \quad (11)$$

其中:

$$\left. \begin{aligned} \tilde{G}_1 &= G_1 Q_1 = A^{(1)} Q_1; \\ \tilde{G}_n &= Q_{n-1}^{-1} \cdot G_n \cdot Q_n = \\ &I_{3,R} \times_1 Q_{n-1}^{-1} \times_2 A^{(n)} \times_3 Q_n^T, 2 \leq n \leq N-1; \\ \tilde{G}_N &= Q_{N-1}^{-1} G_N = Q_{N-1}^{-1} (A^{(N)})^T. \end{aligned} \right\} \quad (12)$$

从式(12)可以看出, 所有 3 阶 TT 核张量都有一个 3 阶 CPD 结构, 其第 2 个因子矩阵对应着张量 Y 的 CPD 的因子矩阵 $A^{(1)}$ 与 $A^{(N)}$, 则可以通过 $A^{(1)} = \tilde{G}_1^T Q_1^{-T}$ 与 $A^{(N)} = \tilde{G}_N^T Q_{N-1}^T$ 得到.

2.2 TTD 到 CPD 的转换

根据 2.1 节提到的 TTD 与 CPD 的对应关系,

对 N 阶张量 \mathbf{Y} 进行 TTD 之后,为得到 \mathbf{Y} 的 CPD, 本文的算法需要对 $N-2$ 个 3 阶核张量 \mathbf{G}_n (图 3) 分别作 CPD 以求出 \mathbf{G}_n 的第二个因子矩阵 $\mathbf{A}^{(n)}$. CPD 固有的排序模糊使得 $N-2$ 次 3 阶 CPD 可能分别对应着不同的置换矩阵 (图 4), 若直接取出 \mathbf{G}_n 的第二个因子矩阵会出现排序错位. 因此, 本文采取顺序调整的方案对各个因子矩阵进行重新排序和缩放.

假设 N 阶秩 R 张量 \mathbf{Y} 具有本质上唯一的 CPD, 如式 (6) 所示. 由于排序与缩放模糊, 对张量 \mathbf{Y} 进行 TTD 之后, TT 核张量的 CPD 结构如式 (13) 所示:

$$\begin{aligned} \mathbf{G}_1 &= \mathbf{A}^{(1)} \mathbf{A}_{11} \Pi_1 (\mathbf{Q}_1^T \mathbf{A}_{12} \Pi_1)^T, \\ \mathbf{G}_n &= \mathbf{I}_{3,R} \times_1 \mathbf{L}_n \times_2 \mathbf{B}^{(n)} \times_3 \mathbf{R}_n = \\ & \mathbf{I}_{3,R} \times_1 \mathbf{Q}_{n-1}^{-1} \mathbf{A}_{n1} \Pi_n \times_2 \mathbf{A}^{(n)} \mathbf{A}_{n2} \Pi_n \times_3 \mathbf{Q}_n^T \mathbf{A}_{n3} \Pi_n, \\ \mathbf{G}_N &= \mathbf{Q}_{N-1}^{-1} \mathbf{A}_{N1} \Pi_N (\mathbf{A}^{(N)} \mathbf{A}_{N2} \Pi_N)^T. \end{aligned} \quad (13)$$

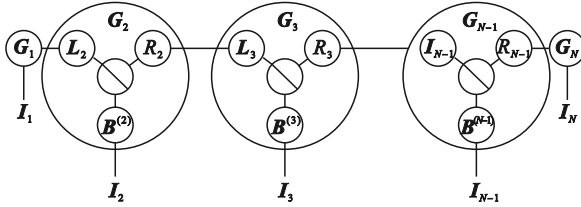


图 3 核张量的 CPD 结构
Fig. 3 CPD of core tensors

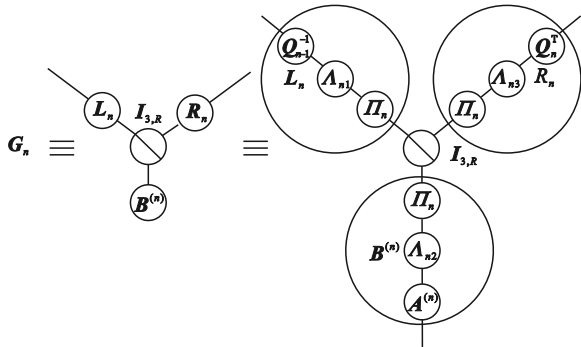


图 4 排序与缩放模糊的张量图表示
Fig. 4 Tensor graph of permutation and scaling ambiguities

其中: Π_n 表示第 n 个核张量的 3 阶 CPD 所对应的置换矩阵, 表示排序模糊; \mathbf{A} 是对角矩阵, 表示缩放模糊, 并且有

$$\left. \begin{aligned} \mathbf{A}_{n1} \mathbf{A}_{n2} \mathbf{A}_{n3} &= \mathbf{I}_R, 2 \leq n \leq N-1; \\ \mathbf{A}_{11} \mathbf{A}_{12} &= \mathbf{A}_{N1} \mathbf{A}_{N2} = \mathbf{I}_R. \end{aligned} \right\} \quad (14)$$

结合式 (13) 与图 3, 图 4 可以得出

$$\left. \begin{aligned} \mathbf{B}^{(0)} &= \mathbf{G}_1 \mathbf{L}_2 = \mathbf{A}^{(0)} \mathbf{A}_{11} \Pi_1 (\mathbf{Q}_1^T \mathbf{A}_{12} \Pi_1)^T \mathbf{Q}_1^{-1} \mathbf{A}_{21} \Pi_2 = \\ & \mathbf{A}^{(0)} \mathbf{A}_{21} \Pi_2; \\ \mathbf{B}^{(n)} &= \mathbf{A}^{(n)} \mathbf{A}_{n2} \Pi_n, 2 \leq n \leq N-1; \\ \mathbf{B}^{(N)} &= \mathbf{G}_N^T \mathbf{R}_{N-1} = \mathbf{A}^{(N)} \mathbf{A}_{N2} \Pi_N (\mathbf{Q}_{N-1}^{-1} \mathbf{A}_{N1} \Pi_N)^T \times \\ & \mathbf{Q}_{N-1}^T \mathbf{A}_{N-1,3} \Pi_{N-1} = \mathbf{A}^{(N)} \mathbf{A}_{(N-1),3} \Pi_{N-1}. \end{aligned} \right\} \quad (15)$$

N 个因子矩阵 $\mathbf{B}^{(n)}$ 是 $\mathbf{A}^{(n)}$ 经过不同的置换矩阵 Π_n 重新排序后所得, 出现了排序错位的现象. 因此, 在对 $N-2$ 个 3 阶核张量 \mathbf{G}_n 分别作 CPD 之后, 需要对因子矩阵进行调整, 使其具有相同的排序. 这需要利用到 \mathbf{G}_n 与 \mathbf{G}_{n+1} 连接处的矩阵 \mathbf{R}_n 和 \mathbf{L}_{n+1} (图 5).

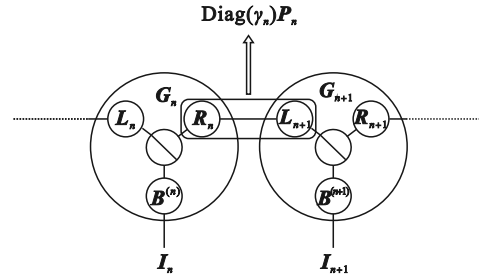


图 5 相邻的核张量
Fig. 5 Adjacent core tensors

$$\begin{aligned} \mathbf{R}_n^T \mathbf{L}_{n+1} &= (\mathbf{Q}_n^T \mathbf{A}_{n3} \Pi_n)^T \mathbf{Q}_{n+1}^{-1} \mathbf{A}_{n+1,1} \Pi_{n+1} = \\ & \Pi_n^T \mathbf{A}_{n3} \mathbf{A}_{n+1,1} \Pi_{n+1} = \\ & \Pi_n^T \mathbf{A}_{n3} \mathbf{A}_{n+1,1} \Pi_n \Pi_n^T \Pi_{n+1} = \text{Diag}(\gamma_n) \mathbf{P}_n. \end{aligned} \quad (16)$$

其中, $\text{Diag}(\gamma_n) = \Pi_n^T \mathbf{A}_{n3} \mathbf{A}_{n+1,1} \Pi_n$, $\mathbf{P}_n = \Pi_n^T \Pi_{n+1}$. 通过矩阵 $\text{Diag}(\gamma_n) \mathbf{P}_n$ 的各行中的主元及主元的位置, 可以确定缩放列向量 γ_n 和置换矩阵 \mathbf{P}_n [7]. 因子矩阵 $\mathbf{B}^{(n+1)}$ 表示 $\mathbf{A}^{(n+1)}$ 的列按照置换矩阵 Π_{n+1} 排序, 根据置换矩阵的性质, 若 $\mathbf{B}^{(n+1)}$ 右乘 $\mathbf{P}_n^T = \Pi_{n+1}^T \Pi_n$, 则 $\mathbf{B}^{(n+1)}$ 变为因子矩阵 $\mathbf{A}^{(n+1)}$ 的列按照 Π_n 排序, 如式 (17) 所示:

$$\mathbf{B}^{(n+1)} \mathbf{P}_n^T = \mathbf{A}^{(n+1)} \mathbf{A}_{n+1,2} \Pi_{n+1} \Pi_{n+1}^T \Pi_n = \mathbf{A}^{(n+1)} \mathbf{A}_{n+1,2} \Pi_n. \quad (17)$$

利用 \mathbf{P}_n^T 将 \mathbf{G}_{n+1} 的因子矩阵重新排序使之具有和 \mathbf{G}_n 的因子矩阵相同的排序, 将缩放列向量吸收进 \mathbf{G}_{n+1} 的第二个因子矩阵, 可以得到张量 \mathbf{G}_{n+1} 的新的 CPD 形式:

$$\begin{aligned} \tilde{\mathbf{G}}_{n+1} &= [\mathbf{L}_{n+1} \mathbf{P}_n^T \text{Diag}(\mathbf{1}_R / \gamma_n), \\ & \mathbf{B}^{(n+1)} \mathbf{P}_n^T \text{Diag}(\gamma_n), \mathbf{R}_{n+1} \mathbf{P}_n^T] = \\ & [\tilde{\mathbf{L}}_{n+1}, \tilde{\mathbf{B}}^{(n+1)}, \tilde{\mathbf{R}}_{n+1}]. \end{aligned} \quad (18)$$

式中: “/” 表示对应位置的元素相除; $\mathbf{1}_R$ 表示元素都为 1 的列向量; $\text{Diag}(\mathbf{I}_R) = \mathbf{I}_R$. 此时有 $\mathbf{R}_n^T \tilde{\mathbf{L}}_{n+1} = \mathbf{R}_n^T \mathbf{L}_{n+1} \mathbf{P}_n^T \text{Diag}(\mathbf{I}_R / \gamma_n) = \mathbf{I}_R$.

因子矩阵 $\mathbf{B}^{(3)}, \mathbf{B}^{(4)}, \dots, \mathbf{B}^{(N-1)}$ 的排序可以被依次纠正为 $\tilde{\mathbf{B}}^{(3)}, \tilde{\mathbf{B}}^{(4)}, \dots, \tilde{\mathbf{B}}^{(N-1)}$. $\tilde{\mathbf{B}}^{(0)}, \tilde{\mathbf{B}}^{(2)}$ 和 $\tilde{\mathbf{B}}^{(N)}$, 可以通过式 (19) 得到

$$\left. \begin{aligned} \tilde{\mathbf{B}}^{(0)} &= \mathbf{B}^{(1)} = \mathbf{G}_1 \mathbf{L}_2 = \mathbf{A}^{(0)} \mathbf{A}_{21} \Pi_2, \\ \tilde{\mathbf{B}}^{(2)} &= \mathbf{B}^{(2)} = \mathbf{A}^{(2)} \mathbf{A}_{22} \Pi_2, \\ \tilde{\mathbf{B}}^{(N)} &= \mathbf{G}_N^T \tilde{\mathbf{R}}_{N-1} = \mathbf{G}_N^T \mathbf{R}_{N-1} \mathbf{P}_{N-2}^T. \end{aligned} \right\} \quad (19)$$

从第二个核张量 \mathbf{G}_2 开始起, 计算出 $\mathbf{R}_2^T \mathbf{L}_3 = \text{Diag}(\gamma_2) \mathbf{P}_2$, 对 \mathbf{G}_3 进行重新排序和缩放可得

$$\begin{aligned} \tilde{\mathbf{B}}^{(3)} &= \mathbf{B}^{(3)} \mathbf{P}_2^T \text{Diag}(\gamma_2) = \\ & \mathbf{A}^{(3)} \mathbf{A}_{32} \mathbf{P}_3 (\mathbf{P}_3^T \mathbf{P}_2) (\mathbf{P}_2^T \mathbf{A}_{23} \mathbf{A}_{31} \mathbf{P}_2) = \\ & \mathbf{A}^{(3)} \mathbf{A}_{32} \mathbf{A}_{23} \mathbf{A}_{31} \mathbf{P}_2 = \mathbf{A}^{(3)} \tilde{\mathbf{A}}_{32} \mathbf{P}_2. \end{aligned} \quad (20)$$

继续按顺序对 $\mathbf{G}_4, \mathbf{G}_5, \dots, \mathbf{G}_{N-1}$ 进行调整, 当 $3 \leq n \leq N-1$ 时, 有 $\tilde{\mathbf{B}}^{(n)} = \mathbf{A}^{(n)} \tilde{\mathbf{A}}_{n2} \mathbf{P}_2$. 最后计算出 $\tilde{\mathbf{B}}^{(N)} = \mathbf{A}^{(N)} \mathbf{A}_{(N-1)3} \mathbf{P}_2$. 此时 N 个因子矩阵都有了相同的排序, 都是按照置换矩阵 \mathbf{P}_2 排序.

经过 $N-2$ 次 3 阶张量 CPD, 对 3 阶 CP 张量的因子矩阵进行重新排序和缩放之后, 最终得到高阶张量 \mathbf{Y} 的秩为 R 的 CPD, 如式 (21) 所示:

$$\mathbf{Y} = [\tilde{\mathbf{B}}^{(1)}, \tilde{\mathbf{B}}^{(2)}, \dots, \tilde{\mathbf{B}}^{(N)}]. \quad (21)$$

在本文中, 3 阶张量的 CPD 采用交替最小二乘法 (ALS) 实现^[4]. 每次运算时, 固定 2 个因子矩阵不变, 只优化 1 个因子矩阵, 通过交替更新的方式估计出 3 个因子矩阵.

根据式 (4), 用上述算法对 $M+1$ 阶观测信号张量 $\tilde{\mathbf{X}}$ 做分解, 求出前 M 个因子矩阵就是要求的 M 个数据集的混合矩阵. 将 M 组观测信号左乘 M 个因子矩阵的广义逆即可恢复出 M 组源信号.

3 实验结果

仿真实验是在一台笔记本上运行, 其配置如下: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz 2.59 GHz; 16 GB 内存; 64 位 Windows 10 操作系统; Matlab R2020b.

3.1 张量分解算法性能对比

随机生成 M 个尺寸为 $I \times R$ 的混合矩阵 $\mathbf{A}^{(m)}$ 与 K 个尺寸为 R 的 M 阶对角信号张量 $\mathbf{D}^{(k)}$, $\mathbf{A}^{(m)}$ 与 $\mathbf{D}^{(k)}$ 中的元素都服从标准正态分布, 根据式 (2) 构造 K 个 M 阶观测信号张量 $\mathbf{X}^{(k)}$. 然后将 K 个 M 阶张量 $\mathbf{X}^{(k)}$ 堆叠为一个 $M+1$ 阶张量.

首先, 在无噪声的情况下, 对比张量分解算法的运行时间. 本文所用的 TT-CPD 算法通过 TTD 对原始张量进行了预压缩, CP-ALS 算法直接对原始张量进行 CPD^[4], CP-FastALS 算法是在 CP-ALS 算法的基础上经过改进的速度更快的算法^[17].

固定张量个数 $K=10$, 选定所有 ALS 算法的误差限为 10^{-10} , 最大迭代次数为 1 000. 在表 1 中, 固定混合矩阵的尺寸 $I=R=5$, 改变数据集个数 M , 即固定张量的尺寸与秩, 改变张量阶数. 在表 2 中, 固定数据集个数 $M=7$ 与张量秩 $R=4$, 改变张量尺寸 I . 仿真给出的运行时间是 50 次蒙特卡洛独立实验取平均值所得.

表 1 数据集个数不同时的算法运行时间

Table 1 Running time of algorithms under different numbers of datasets

(I, R, M)	TT-CPD	CP-ALS	CP-FastALS
(5, 5, 6)	0.452 5	0.580 8	0.090 8
(5, 5, 7)	0.505 9	1.754 1	0.148 0
(5, 5, 8)	0.628 5	3.074 8	0.346 7
(5, 5, 9)	1.410 2	11.071 6	1.837 2

表 2 张量尺寸不同时的算法运行时间

Table 2 Running time of algorithms under different sizes of tensors

(I, R, M)	TT-CPD	CP-ALS	CP-FastALS
(5, 4, 7)	0.366 7	0.808 8	0.143 7
(7, 4, 7)	0.438 9	6.014 0	0.509 6
(9, 4, 7)	1.380 8	15.958 6	2.374 0
(11, 4, 7)	5.889 4	41.351 8	9.488 2

由表 1 和表 2 可以看出, 随着数据集个数 M 的增加, CP-ALS 相较 TT-CPD 的运行时间的倍数也在显著增加. 当数据集个数增长到 9 时, TT-CPD 的表现优于 CP-FastALS, 说明数据集个数越多, TT-CPD 相对 CP-FastALS 越有优势, 这是因为 TT-CPD 可以通过 TTD 的步骤将高阶张量分解问题转化为多个低阶张量分解问题. 另外, 对于大尺寸的张量, TT-CPD 表现优于 CP-FastALS.

接下来评估算法在有噪声时的鲁棒性, 固定张量个数 $K=10$, 混合矩阵尺寸 $I=R=3$ 与数据集个数 $M=7$. 带噪声项的目标张量按照文献 [18] 生成, 如式 (22) 所示:

$$\hat{\mathbf{X}}^{(k)} = \mathbf{X}^{(k)} + \sigma^{(k)} \mathbf{E}^{(k)}, k = 1, 2, \dots, K. \quad (22)$$

其中, 误差张量 $\mathbf{E}^{(k)}$ 的元素服从标准正态分布, 通过系数 $\sigma^{(k)}$ 可以调整噪声强度进而调整信噪比:

$$\text{SNR} = 10 \lg \left(\frac{\|\mathbf{X}^{(k)}\|^2}{\|\sigma^{(k)} \mathbf{E}^{(k)}\|^2} \right). \quad (23)$$

将 K 个带噪声项的 M 阶张量 $\hat{\mathbf{X}}^{(k)}$ 堆叠为一个 $M+1$ 阶张量, 使用文献 [18-19] 所提及的性能指标式 (24) 来衡量张量分解算法的精度:

$$\text{PI}(\mathbf{G}^{(1)}, \mathbf{G}^{(2)}, \dots, \mathbf{G}^{(M)}) = \frac{1}{M} \{ \text{pi}(\mathbf{G}^{(1)}) + \text{pi}(\mathbf{G}^{(2)}) + \dots + \text{pi}(\mathbf{G}^{(M)}) \}. \quad (24)$$

其中:

$$\text{pi}(\mathbf{G}^{(m)}) = \frac{1}{2R(R-1)} \left[\sum_{i=1}^R \left(\frac{\sum_{j=1}^R |g_{ij}^{(m)}|^2}{\max_s |g_{is}^{(m)}|^2} - 1 \right) + \sum_{j=1}^R \left(\frac{\sum_{i=1}^R |g_{ij}^{(m)}|^2}{\max_s |g_{sj}^{(m)}|^2} - 1 \right) \right]. \quad (25)$$

式中, $\mathbf{G}^{(m)} = (\hat{\mathbf{A}}^{(m)})^\dagger \mathbf{A}^{(m)}$, 其中 $(\hat{\mathbf{A}}^{(m)})^\dagger$ 表示第 m 个估计的混合矩阵的广义逆矩阵. 性能指标 PI 的值是非负的, 当矩阵 \mathbf{G} 中任意非零元素所在的行和列只有一个非零元素, 即矩阵 \mathbf{G} 等于一个对角矩阵与一个置换矩阵的乘积时, 该性能指标的值等于 0. PI 的值越小, 则张量分解算法的精度越高.

图 6 显示了 3 种算法在不同噪声时的 PI 值, 曲线中的每个点由 300 次蒙特卡洛独立实验取平均值所得.

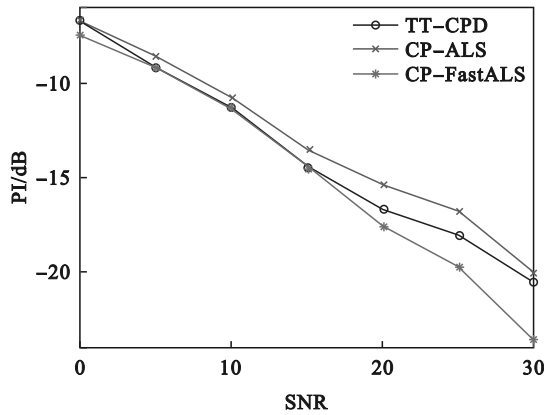


图 6 不同 SNR 时算法的对比

Fig. 6 Comparison of algorithms under different SNR

可以看出, 在 SNR 较小时, TT-CPD 与 CP-FastALS 精度相当. 若是在 SNR 不大, 数据集个数多, 混合矩阵尺寸大的情况, 选择 TT-CPD 更有优势.

接下来, 令 $I=R=3, M=4, 6, 8$, 测试在数据集个数 M 不同时 TT-CPD 算法的性能, 曲线中的每个点为 300 次蒙特卡洛独立实验平均值, 如图 7 所示.

在数据集个数越多时 TT-CPD 算法的精确度越好. 这是因为在 JBSS 中, 数据集个数增加就相当于从更多的角度观测源信号, 从而获得了更多的信息, 分离的结果也就更精确.

3.2 多数据集音频源信号的分离

本节的实验模拟了 TT-CPD 算法在 JBSS 问题中的应用. 取 4 段时长为 10 s、采样频率为 16 000 Hz

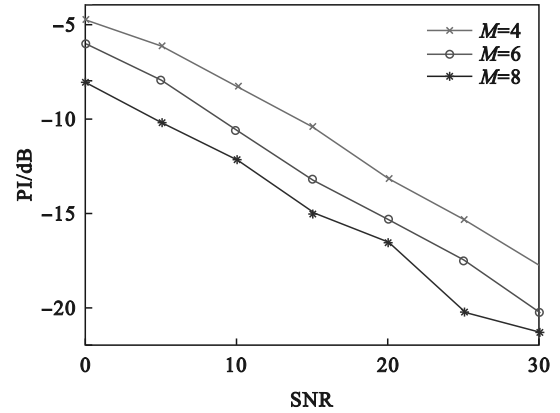


图 7 数据集个数不同时 TT-CPD 算法的性能

Fig. 7 Performance of TT-CPD algorithm under different numbers of datasets

的真实音频信号作为第 1 个数据集的源信号 s_1 , 如图 8a 所示. 通过文献 [20] 所提的式 (26) 再生成另外 $(M-1)$ 组源信号. 随机生成 M 个 $I \times 4$ 的混合矩阵, 其中的元素都服从标准正态分布. 然后通过式 (1) 构造无噪声时的 M 组观测信号, 利用 TT-CPD 算法对 M 组源信号做分离.

$$\left. \begin{aligned} s_1 &= [s_1^{(1)}, s_1^{(2)}, \dots, s_1^{(R)}]^\top, \\ s_2 &= [s_2^{(1)}, s_2^{(2)}, \dots, s_2^{(R)}]^\top = \\ & s_1 \cdot (\text{unifrnd}(0, 1, s_1)), \\ & \vdots \\ s_M &= [s_M^{(1)}, s_M^{(2)}, \dots, s_M^{(R)}]^\top = \\ & s_1 \cdot (\text{unifrnd}(0, 1, s_1)). \end{aligned} \right\} \quad (26)$$

式中: s_m 表示第 m 个数据集的源信号; $s_m^{(r)}$ 表示 s_m 的第 r 个分量且各分量相互独立; $\text{unifrnd}(0, 1, s_1)$ 生成与 s_1 大小相同的数组, 并且数组的每一个元素服从区间 $[0, 1]$ 上的均匀分布. 通过式 (26) 可以生成具有组间相关性的多数据集源信号.

取 $M=I=4$, 根据 4 组观测信号分离 4 组源信号. 图 8 显示了第 1 组源信号、混合信号及分离信号的波形图. 表 3 展示了 4 组源信号与 4 组分离信号的相关系数. 实验结果表明本文所提的算法具有良好的分离精度.

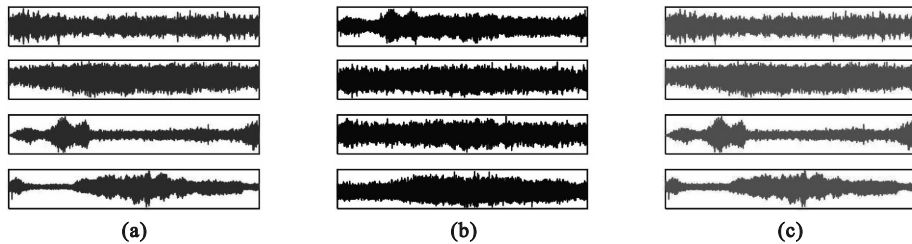


图 8 第 1 组信号的波形图

Fig. 8 Waveforms of signals of group 1
(a)—源信号; (b)—混合信号; (c)—分离信号.

表3 4组源信号与分离信号的相关系数
Table 3 Correlation coefficients of source signals and separated signals of 4 groups

信号组	$s^{(1)}$	$s^{(2)}$	$s^{(3)}$	$s^{(4)}$
1	0.998 4	0.998 9	0.997 2	0.986 3
2	0.999 5	0.998 1	0.997 2	0.987 8
3	0.999 6	0.996 5	0.999 2	0.992 2
4	0.998 6	0.996 6	0.996 1	0.995 3

4 结 语

本文提出一种基于高阶累积量张量分解的JBSS算法. 通过计算多数据集信号的高阶互累积量张量, 利用其潜在的对角结构将JBSS问题转化为高阶张量CPD问题. 张量分解算法, 特别是TTD在信号处理中的应用属于比较新的研究领域. 本文利用TTD与CPD的对应关系将高阶张量CPD问题转化为一组3阶的CPD问题以降低计算成本. 实验结果表明本文算法能够较好地多数据集信号进行分离.

参考文献:

- [1] Lahat D, Adali T, Jutten C. Multimodal data fusion: an overview of methods, challenges, and prospects [J]. *Proceedings of the IEEE*, 2015, 103(9): 1449-1477.
- [2] 龚晓峰, 毛蕾, 林秋华, 等. 基于四阶累积量张量联合对角化的多数据集联合盲源分离[J]. *电子与信息学报*, 2019, 41(3): 509-515.
(Gong Xiao-feng, Mao Lei, Lin Qiu-hua, et al. Joint blind source separation based on joint diagonalization of fourth-order cumulant tensors [J]. *Journal of Electronics & Information Technology*, 2019, 41(3): 509-515.)
- [3] 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.
- [4] Kolda T G, Bader B W. Tensor decompositions and applications [J]. *SIAM Review*, 2009, 51(3): 455-500.
- [5] Sorber L, van Barel M, de Lathauwer L. Structured data fusion [J]. *IEEE Journal of Selected Topics in Signal Processing*, 2015, 9(4): 586-600.
- [6] Cichocki A, Mandic D, de Lathauwer L, et al. Tensor decompositions for signal processing applications: from two-way to multiway component analysis [J]. *IEEE Signal Processing Magazine*, 2015, 32(2): 145-163.
- [7] Phan A H, Cichocki A, Oseledets I, et al. Tensor networks for latent variable analysis: higher order canonical polyadic decomposition [J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2020, 31(6): 2174-2188.
- [8] Zniyed Y, Boyer R, de Almeida A L F, et al. High-order tensor estimation via trains of coupled third-order CP and Tucker decompositions [J]. *Linear Algebra and Its Applications*, 2020, 588: 304-337.
- [9] Oseledets I V. Tensor-train decomposition [J]. *SIAM Journal on Scientific Computing*, 2011, 33(5): 2295-2317.
- [10] Cardoso J F, Souloumiac A. Blind beamforming for non-Gaussian signals [J]. *IEE Proceedings F: Radar and Signal Processing*, 1993, 140(6): 362.
- [11] Miao J F, Cheng G H, Cai Y F, et al. Approximate joint singular value decomposition algorithm based on givens-like rotation [J]. *IEEE Signal Processing Letters*, 2018, 25(5): 620-624.
- [12] De Lathauwer L, de Moor B, Vandewalle J. Independent component analysis and (simultaneous) third-order tensor diagonalization [J]. *IEEE Transactions on Signal Processing*, 2001, 49(10): 2262-2271.
- [13] 张贤达. 时间序列分析: 高阶统计量方法 [M]. 北京: 清华大学出版社, 1996: 1-24.
(Zhang Xian-da. Time series analysis: higher-order statistics method [M]. Beijing: Tsinghua University Press, 1996: 1-24.)
- [14] Xia J, Cheng G H, Miao J F. An adaptive nonunitary joint diagonalization algorithm of high-order tensors [C] // 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI). Beijing, 2018: 1-4.
- [15] Phan A H, Cichocki A, Uschmajew A, et al. Tensor networks for latent variable analysis. part I: algorithms for tensor train decomposition [EB/OL]. (2016-07-29) [2022-07-02]. 2016: arXiv: 1609.09230. <https://arxiv.org/abs/1609.09230.pdf>.
- [16] Phan A H, Cichocki A, Uschmajew A, et al. Tensor networks for latent variable analysis: novel algorithms for tensor train approximation [J]. *IEEE Transactions on Neural Networks and Learning Systems*, 2020, 31(11): 4622-4636.
- [17] Phan A H, Tichavský P, Cichocki A. Fast alternating LS algorithms for high order Candecomp/Parafac tensor factorizations [J]. *IEEE Transactions on Signal Processing*, 2013, 61(19): 4834-4846.
- [18] Miao J F, Cheng G H, Li W R, et al. A unitary joint diagonalization algorithm for nonsymmetric higher-order tensors based on Givens-like rotations [J]. *Numerical Linear Algebra with Applications*, 2020, 27(3): e2291.
- [19] Gong X F, Wang X L, Lin Q H. Generalized non-orthogonal joint diagonalization with LU decomposition and successive rotations [J]. *IEEE Transactions on Signal Processing*, 2015, 63(5): 1322-1334.
- [20] Zou L, Wang Z J, Chen X, et al. Underdetermined joint blind source separation based on tensor decomposition [C] // 2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE). Vancouver, 2016: 1-4.