中国科学: 信息科学 2016 年 第 46 卷 第 7 期: 819–833
SCIENTIA SINICA Informationis
《中国科学》杂志社
SCIENCE CHINA PRESS
论 文

基于多源共享因子的多张量填充

张骁, 胡清华, 廖士中*

天津大学计算机科学与技术学院, 天津 300350
* 通信作者. E-mail: szliao@tju.edu.cn

收稿日期: 2016–03–09; 接受日期: 2016–03–28
国家自然科学基金重点项目 (批准号: 61432011) 和国家自然科学基金 (批准号: 61170019) 资助项目

摘要 张量填充在数据挖掘、机器学习、生物信号处理等领域有着广泛的应用. 现有的张量填充方法多在低秩假设的前提下对单独的张量进行填充, 然而由于张量数据的复杂结构, 张量填充的精度通常较低. 为此, 研究不同来源多个张量同时填充的方法. 首先, 利用 Tucker 分解将多源张量填充问题转换为最小二乘问题. 然后, 假设不同来源的张量在共享模式上具有共同的信息, 为 Tucker 分解构造共享的因子矩阵集, 提取多源张量在共享模式上的共同潜在结构, 进而建立基于共享因子的多源张量填充 (SF-MTC) 方法. 最后, 利用非线性共轭梯度法和奇异值分解 (SVD) 快速求解 Tucker 分解的因子矩阵集及核心张量, 完成同时对多个张量的填充, 并进一步分析了 SF-MTC 的计算复杂度. 在人工及实际数据集上的实验结果表明, 所提出的 SF-MTC 能提高张量填充的求解效率, 并在具有相关性的多源张量数据集上得到更高的填充精度.

关键词 张量填充 Tucker 分解 共享因子 非线性共轭梯度法 奇异值分解

1 引言

高阶张量为高维数据的表示及挖掘提供了便捷的方式 [1,2]. 由于数据采集过程中的信息损失等因素, 张量数据常常缺失部分元素. 如何利用张量已知元素对缺失部分的值进行估计称为张量填充问题. 张量填充现已成功应用于视觉数据、时空数据、神经影像数据的处理中 [3~5].

近年来, 低秩矩阵填充得到了广泛的关注 [6,7]. 其中, 核范数常用于近似矩阵的秩函数, 据此可将低秩矩阵填充问题转化为凸优化的问题 [8]. Tomioka 等 [9] 通过在每个模式上将张量展开为矩阵, 利用多个矩阵核范数的加权和近似张量的秩函数, 建立了低秩张量填充的凸优化形式. 针对张量填充的张量核范数框架, Liu 等 [5] 提出了 3 种有效的优化算法 SiLRTC, FaLRTC 和 HaLRTC. 此外, 张量分解由于能够精确的表示张量数据的潜在结构及不同模式上的相关性, 已经成为填充不完整张量数据的重要工具 [10]. Acar 等 [11] 利用张量的 CANDECOMP/PARAFAC(CP) 分解构造了带权最小二乘问题, 并应用到低秩张量的填充问题中. 另外一种常用的张量分解 — Tucker 分解, 也被引入到低秩张量填充的带权最小二乘问题中 [12], 该方法在输入张量的秩不明确的情形下仍可以获得较好的填充效果.

引用格式: 张骁, 胡清华, 廖士中. 基于多源共享因子的多张量填充. 中国科学: 信息科学, 2016, 46: 819–833, doi: 10.1360/N112016-00049

对于单个张量数据, 大多数的张量填充方法在低秩假设下, 利用已知元素和未知元素之间的低秩结构对未知元素的值进行估计. 在低秩假设的基础上, 引入更多的先验信息可以进一步提升张量填充的性能 [13]. 对于多源张量数据, 可以利用多个张量的潜在相似性结构获得更多的信息, 进而提升张量填充的性能. Narita 等 [13] 利用多个数据集的相似性构造 Laplacian 矩阵, 提出了张量填充的两种正则化方法. Acar 等 [14] 利用对多源数据的联合分析 (joint analysis), 利用 CP 分解对一个张量和与其相关的矩阵同时进行分解, 达到了提升张量填充精度的目的.

本文在张量低秩假设的前提下, 进一步假设多源张量数据在某些模式上具有相关性, 并在这些模式上设置共享因子, 探索多张量之间的相关性, 进而应用带权最小二乘目标函数来建立多源张量填充方法, 可同步分解并填充多源相关张量.

2 预备知识

本节给出张量代数中的一些符号及定义. 为了便于区分, 将向量和矩阵分别定义为小写和大写的黑体字母, 如向量 $\boldsymbol{x}$、矩阵 $\boldsymbol{X}$. 高阶张量用花体表示, 如 $\boldsymbol{\mathcal{X}}$. 对于一个 N 阶张量 $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, 其 $(i_1, i_2, \ldots, i_N)$ 元素为 $x_{i_1,i_2,\ldots,i_N}$.

定义1 (张量积及张量) 令 $\boldsymbol{x} \in V$, $\boldsymbol{y} \in W$, 其中 $V \subseteq \mathbb{R}^n$, $W \subseteq \mathbb{R}^m$ 为向量空间. $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 的张量积记为 $\boldsymbol{x} \otimes \boldsymbol{y}$, 其中 $(\boldsymbol{x} \otimes \boldsymbol{y})_{ij} = x_i y_j$. N 阶张量 $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ 为 N 个向量空间的张量积中的一个元素, 即

$$\boldsymbol{\mathcal{X}} = \sum \boldsymbol{u}_1 \otimes \boldsymbol{u}_2 \otimes \cdots \otimes \boldsymbol{u}_N,$$

其中, $\boldsymbol{u}_i \in \mathbb{R}^{I_i}$, $i = 1, 2, \ldots, N$, 称 N 阶张量的每一维为一个模式.

定义2 (n- 式展开及 n- 秩) 令 $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, 其 n- 式向量为 $\mathbb{R}^{I_n}$ 空间中的一个元素, 可通过变化 $\boldsymbol{\mathcal{X}}$ 的指标 i_n 并固定其他指标得到. 张量 $\boldsymbol{\mathcal{X}}$ 的 n- 式展开 (矩阵化) 记为矩阵 $\boldsymbol{\mathcal{X}}_{(n)} \in \mathbb{R}^{I_n \times L_n}$, 其列向量为 $\boldsymbol{\mathcal{X}}$ 的 n- 式向量, 其中

$$L_n = \prod_{k \in \{1,2,\ldots,n-1,n+1,\ldots,N\}} I_k.$$

相应地, $\boldsymbol{\mathcal{X}}_{(n)}$ 的秩被称为 $\boldsymbol{\mathcal{X}}$ 的 n- 秩, 记为 $\mathrm{rank}_n(\boldsymbol{\mathcal{X}})$. 对于 $n = 1, 2, \ldots, N$, 若令 $r_n = \mathrm{rank}_n(\boldsymbol{\mathcal{X}})$, 则称 $\boldsymbol{\mathcal{X}}$ 为秩 -$(r_1, r_2, \ldots, r_n)$ 张量.

定义3 (n- 式积) 张量 $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ 与矩阵 $\boldsymbol{U} \in \mathbb{R}^{J \times I_n}$ 的 n- 式积记为 $\boldsymbol{\mathcal{X}} \times_n \boldsymbol{U}$, 其定义为

$$(\boldsymbol{\mathcal{X}} \times_n \boldsymbol{U})_{(n)} = \boldsymbol{U} \boldsymbol{\mathcal{X}}_{(n)} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N}.$$

定义4 (Tucker 分解) Tucker 分解将张量 $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ 分解为一个核心张量 $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ 在每个模式上的 n- 式积:

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \cdots \times_N \boldsymbol{U}_N,$$

其中, $\{\boldsymbol{U}_n\}$ 为因子矩阵 (通常是正交的). 此外, 若 $\boldsymbol{\mathcal{X}}$ 为秩 -$(r_1, r_2, \ldots, r_N)$ 张量, 则 $J_k = r_k$, $k = 1, 2, \ldots, N$.

定义5 (CANDECOMP/PARAFAC (CP) 分解) 张量 $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ 的 CP 分解可表示为如下形式:

$$\boldsymbol{\mathcal{X}} = [\![\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \ldots, \boldsymbol{U}^{(N)}]\!] = \sum_{r=1}^{R} \boldsymbol{u}_r^{(1)} \otimes \boldsymbol{u}_r^{(2)} \otimes \cdots \otimes \boldsymbol{u}_r^{(N)},$$

其中, $\boldsymbol{u}_r^{(n)} \in \mathbb{R}^{I_n}$, $\boldsymbol{U}^{(n)} = [\boldsymbol{u}_1^{(n)}, \boldsymbol{u}_2^{(n)}, \ldots, \boldsymbol{u}_R^{(n)}]$. 若 $\boldsymbol{u}_r^{(n)}$ 为单位向量, 则 CP 分解可表示为

$$\boldsymbol{\mathcal{X}} = [\![\boldsymbol{\lambda};\ \boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \ldots, \boldsymbol{U}^{(N)}]\!] = \sum_{r=1}^{R} \lambda_r \boldsymbol{u}_r^{(1)} \otimes \boldsymbol{u}_r^{(2)} \otimes \cdots \otimes \boldsymbol{u}_r^{(N)},$$

其中, $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_R]^{\mathrm{T}}$.

定义6 (Hadamard 积) 张量 $\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ 的 Hadamard 积为对应元素的乘积, 记为 $\boldsymbol{\mathcal{X}} \cdot \boldsymbol{\mathcal{Y}}$, 即

$$(\boldsymbol{\mathcal{X}} \cdot \boldsymbol{\mathcal{Y}})_{i_1, i_2, \ldots, i_N} = x_{i_1, i_2, \ldots, i_N} y_{i_1, i_2, \ldots, i_N}.$$

定义7 (内积及 Frobenius 范数) $\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}}$ 的内积为其对应元素乘积之和, 即

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

相应地, 张量 $\boldsymbol{\mathcal{X}}$ 的 Frobenius 范数为 $\|\boldsymbol{\mathcal{X}}\|_F^2 = \langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}} \rangle$.

对于 N 阶张量 $\boldsymbol{\mathcal{X}}$, 本文规定以下张量运算符号:

$$\mathcal{P}(\boldsymbol{\mathcal{X}}, \{\boldsymbol{U}_n\}^N) = \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \times_3 \cdots \times_N \boldsymbol{U}_N, \quad \mathcal{P}^{\mathrm{T}}(\boldsymbol{\mathcal{X}}, \{\boldsymbol{U}_n\}^N) = \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{U}_1^{\mathrm{T}} \times_2 \boldsymbol{U}_2^{\mathrm{T}} \times_3 \cdots \times_N \boldsymbol{U}_N^{\mathrm{T}}.$$

3 张量分解及填充方法

对于部分位置不可观测的 N 阶张量 $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, 基于张量的 CP 分解, Acar 等 [11, 15] 提出了带权重的张量填充方法 (CP-WOPT):

$$\min_{\{\boldsymbol{U}^{(n)}\}} \left\| \boldsymbol{\mathcal{W}}_\Omega \cdot \left[\boldsymbol{\mathcal{X}} - [\![\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \ldots, \boldsymbol{U}^{(N)}]\!] \right] \right\|_F^2,$$

其中, $\boldsymbol{U}^{(n)} \in \mathbb{R}^{I_n \times R}$, $\boldsymbol{\mathcal{W}}_\Omega$ 为与张量 $\boldsymbol{\mathcal{X}}$ 规模相同的非负权重张量:

$$(w_\Omega)_{i_1, i_2, \ldots, i_N} = \begin{cases} 1, & \text{若 } x_{i_1, i_2, \ldots, i_N} \in \Omega, \\ 0, & \text{若 } x_{i_1, i_2, \ldots, i_N} \in \Omega^C, \end{cases}$$

其中, 指标集 Ω 表示观测到的张量元素的位置, Ω^C 表示 Ω 的补集.

通过对多源数据的联合分析, Acar 等 [14] 利用张量的 CP 分解给出了矩阵、张量的联合填充方法 CMTF-OPT:

$$\min_{\{\boldsymbol{U}^{(n)}\}, \boldsymbol{V}} \left\| \boldsymbol{\mathcal{W}}_\Omega \cdot \left[\boldsymbol{\mathcal{X}} - [\![\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \ldots, \boldsymbol{U}^{(N)}]\!] \right] \right\|_F^2 + \|\boldsymbol{Y} - \boldsymbol{U}^{(n)} \boldsymbol{V}^{\mathrm{T}}\|_F^2.$$

对于张量 $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ 和矩阵 $\boldsymbol{Y} \in \mathbb{R}^{I_n \times M}$, 该方法通过共享第 n 个模式来挖掘二者的相关性信息, 并同时对张量和矩阵进行分解及填充.

4 多源张量填充方法

Filipović 等 [12] 提出了基于 Tucker 分解的低 n- 秩张量填充方法 (Tuck-WOPT), 该方法在张量 $\boldsymbol{\mathcal{X}}$ 的 n- 秩估计为 $(r_1, r_2, \ldots, r_N)$ 的情形下, 可表示为

$$\min_{\boldsymbol{\mathcal{G}}, \{\boldsymbol{U}_n\}} \left\| \boldsymbol{\mathcal{W}}_\Omega \cdot \left[\boldsymbol{\mathcal{X}} - \mathcal{P}(\boldsymbol{\mathcal{G}}, \{\boldsymbol{U}_n\}^N) \right] \right\|_F^2,$$

其中, 因子矩阵 $\boldsymbol{U}_n \in \mathbb{R}^{I_n \times r_n}$. 该方法可以利用非线性共轭梯度法快速求解.

首先, 针对 M 个来源的相关张量 $\boldsymbol{\mathcal{X}}_k \in \mathbb{R}^{I_1^{(k)} \times I_2^{(k)} \times \cdots \times I_{N_k}^{(k)}}$, 给出基于 Tucker 分解的多源张量填充方法:

$$\begin{aligned} \min_{\{\boldsymbol{\mathcal{F}}_k\},\{\boldsymbol{\mathcal{G}}_k\},\{\boldsymbol{U}_n^{(k)}\}} & \sum_{k=1}^{M} \left\| \boldsymbol{\mathcal{F}}_k - \mathcal{P}(\boldsymbol{\mathcal{G}}_k, \{\boldsymbol{U}_n^{(k)}\}^{N_k}) \right\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{\mathcal{W}}_\Omega^{(k)} \cdot \boldsymbol{\mathcal{F}}_k = \boldsymbol{\mathcal{W}}_\Omega^{(k)} \cdot \boldsymbol{\mathcal{X}}_k, \quad k = 1, 2, \ldots, M, \end{aligned} \tag{1}$$

其中, 因子矩阵 $\boldsymbol{U}_n^{(k)} \in \mathbb{R}^{I_n^{(k)} \times r_n}$.

假设 M 个待填充相关张量在前 D 个模式上具有共享的因子矩阵集 $\{\boldsymbol{V}_d\}_{d=1}^{D}$, 即

$$\boldsymbol{U}_d^{(1)} = \boldsymbol{U}_d^{(2)} = \cdots = \boldsymbol{U}_d^{(M)} = \boldsymbol{V}_d, \quad d = 1, 2, \ldots, D,$$

在此情形下, 第 k 个张量的因子矩阵集包含 D 个共享因子矩阵和 $N_k - D$ 个非共享因子矩阵, 记为

$$\{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k} = \{\underbrace{\boldsymbol{V}_1, \boldsymbol{V}_2, \ldots, \boldsymbol{V}_D}_{\text{共享因子矩阵}}, \underbrace{\boldsymbol{U}_{D+1}^{(k)}, \boldsymbol{U}_{D+2}^{(k)}, \ldots, \boldsymbol{U}_{N_k}^{(k)}}_{\text{非共享因子矩阵}}\}.$$

因此, 对于 M 个相关张量

$$\boldsymbol{\mathcal{X}}_k \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_D \times I_{D+1}^{(k)} \times \cdots \times I_{N_k}^{(k)}},$$

由式 (1) 可得到如下具有 D 个共享因子的多源张量填充 (SF-MTC) 方法:

$$\begin{aligned} \min_{\{\boldsymbol{\mathcal{F}}_k\},\{\boldsymbol{\mathcal{G}}_k\},\{\boldsymbol{V}_n\},\{\boldsymbol{U}_n^{(k)}\}} & \sum_{k=1}^{M} \left\| \boldsymbol{\mathcal{F}}_k - \mathcal{P}(\boldsymbol{\mathcal{G}}_k, \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}) \right\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{\mathcal{W}}_\Omega^{(k)} \cdot \boldsymbol{\mathcal{F}}_k = \boldsymbol{\mathcal{W}}_\Omega^{(k)} \cdot \boldsymbol{\mathcal{X}}_k, \quad k = 1, 2, \ldots, M, \end{aligned} \tag{2}$$

其中, 共享因子矩阵

$$\boldsymbol{V}_n = [\boldsymbol{v}_{1,n}, \boldsymbol{v}_{2,n}, \ldots, \boldsymbol{v}_{r_n,n}] \in \mathbb{R}^{I_n \times r_n},$$

非共享因子矩阵

$$\boldsymbol{U}_n^{(k)} = [\boldsymbol{u}_{1,n}^{(k)}, \boldsymbol{u}_{2,n}^{(k)}, \ldots, \boldsymbol{u}_{r_n,n}^{(k)}] \in \mathbb{R}^{I_n^{(k)} \times r_n}.$$

若因子矩阵为列正交的, 则利用文献 [16] 中的定理 4.2, 可将关于因子矩阵集 $\{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}$、核心张量 $\{\boldsymbol{\mathcal{G}}_k\}$ 和 $\{\boldsymbol{\mathcal{F}}_k\}$ 的最小化问题 (2) 转化为关于 $\{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}$ 和 $\{\boldsymbol{\mathcal{F}}_k\}$ 的最大化问题:

$$\begin{aligned} \max_{\{\boldsymbol{\mathcal{F}}_k\},\{\boldsymbol{V}_n\},\{\boldsymbol{U}_n^{(k)}\}} & \sum_{k=1}^{M} \left\| \mathcal{P}^{\mathrm{T}}(\boldsymbol{\mathcal{F}}_k, \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}) \right\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{\mathcal{W}}_\Omega^{(k)} \cdot \boldsymbol{\mathcal{F}}_k = \boldsymbol{\mathcal{W}}_\Omega^{(k)} \cdot \boldsymbol{\mathcal{X}}_k, \quad k = 1, 2, \ldots, M. \end{aligned} \tag{3}$$

令核心张量为超对角张量 (即仅有 $(i, i, \ldots, i)$ 元素为非零的), 若张量 $\{\boldsymbol{\mathcal{X}}_k\}$ 的 n- 秩为 $(R, R, \ldots, R)$, 则张量的 Tucker 分解退化为 CP 分解 [4]. 此时, SF-MTC 可以转化为其 CP 分解形式:

$$\min_{\{\boldsymbol{\mathcal{F}}_k\},\{\boldsymbol{\mathcal{C}}_k\},\{\boldsymbol{V}_n\},\{\boldsymbol{U}_n^{(k)}\}} \sum_{k=1}^{M} \left\| \boldsymbol{\mathcal{F}}_k - \mathcal{P}(\boldsymbol{\mathcal{C}}_k, \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}) \right\|_F^2$$

$$\text{s.t.} \qquad \boldsymbol{\mathcal{W}}_{\Omega}^{(k)} \cdot \boldsymbol{\mathcal{F}}_k = \boldsymbol{\mathcal{W}}_{\Omega}^{(k)} \cdot \boldsymbol{\mathcal{X}}_k, \quad k = 1, 2, \ldots, M,$$

其中, 因子矩阵 $\widetilde{\boldsymbol{U}}_n^{(k)}$ 的列向量为单位向量. 核心张量 $\boldsymbol{\mathcal{C}}_k$ 的 $(i, i, \ldots, i)$ 元素为 c_i^k. 因此, 填充后的张量 $\{\hat{\boldsymbol{\mathcal{X}}}_k\}$ 可表示为 CP 分解的形式:

$$\hat{\boldsymbol{\mathcal{X}}}_k = [\![\boldsymbol{c}^{(k)};\ \boldsymbol{V}_1, \ldots, \boldsymbol{V}_D, \boldsymbol{U}_{D+1}^{(k)}, \ldots, \boldsymbol{U}_{N_k}^{(k)}]\!] = \sum_{r=1}^{R} c_r^{(k)} \boldsymbol{v}_{r,1} \otimes \cdots \otimes \boldsymbol{v}_{r,D} \otimes \boldsymbol{u}_{r,D+1}^{(k)} \otimes \cdots \otimes \boldsymbol{u}_{r,N_k}^{(k)},$$

其中, $\boldsymbol{c}^{(k)} = (c_1^{(k)}, c_2^{(k)}, \ldots, c_R^{(k)})^{\mathrm{T}}$.

5 优化算法

本节将优化问题 (2) 分解为多个子问题, 采用交替投影的方法分别求解 $\{\boldsymbol{\mathcal{F}}_k\}$、核心张量 $\{\boldsymbol{\mathcal{G}}_k\}$ 及因子矩阵 $\{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}$. 对于 N 阶张量 $\boldsymbol{\mathcal{X}}$, 求解过程中需在 $N-1$ 个模式上作矩阵投影, 故定义如下张量运算:

$$\mathcal{P}_n(\boldsymbol{\mathcal{X}}, \{\boldsymbol{U}_n\}^N) = \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{U}_1 \times_2 \boldsymbol{U}_2 \cdots \times_{n-1} \boldsymbol{U}_{n-1} \times_{n+1} \boldsymbol{U}_{n+1} \cdots \times_N \boldsymbol{U}_N,$$
$$\mathcal{P}_n^{\mathrm{T}}(\boldsymbol{\mathcal{X}}, \{\boldsymbol{U}_n\}^N) = \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{U}_1^{\mathrm{T}} \times_2 \boldsymbol{U}_2^{\mathrm{T}} \cdots \times_{n-1} \boldsymbol{U}_{n-1}^{\mathrm{T}} \times_{n+1} \boldsymbol{U}_{n+1}^{\mathrm{T}} \cdots \times_N \boldsymbol{U}_N^{\mathrm{T}}.$$

5.1 求解过程

首先, 利用文献 [17] 中的高阶 SVD (HOSVD) 算法初始化核心张量 $\{\boldsymbol{\mathcal{G}}_k\}$ 及因子矩阵 $\{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}$.

计算 $\{\boldsymbol{\mathcal{F}}_k\}$: 由式 (2) 关于 $\{\boldsymbol{\mathcal{F}}_k\}$ 的一阶 KKT 条件可得

$$\boldsymbol{\mathcal{F}}_k = \mathcal{P}(\boldsymbol{\mathcal{G}}_k, \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}),$$

因此, 由约束条件 $\boldsymbol{\mathcal{W}}_{\Omega}^{(k)} \cdot \boldsymbol{\mathcal{F}}_k = \boldsymbol{\mathcal{W}}_{\Omega}^{(k)} \cdot \boldsymbol{\mathcal{X}}_k$ 可得

$$\boldsymbol{\mathcal{F}}_k = \boldsymbol{\mathcal{W}}_{\Omega^C}^{(k)} \cdot \mathcal{P}(\boldsymbol{\mathcal{G}}_k, \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}) + \boldsymbol{\mathcal{W}}_{\Omega}^{(k)} \cdot \boldsymbol{\mathcal{X}}_k. \tag{4}$$

计算非共享因子集 $\{\boldsymbol{U}_i^{(k)}\}_{i=D+1}^{N_k}$: 对于固定的 $\{\boldsymbol{\mathcal{F}}_k\}$ 和共享因子矩阵集 $\{\boldsymbol{V}_d\}_{d=1}^{D}$, 最大化问题 (3) 可表示为关于非共享因子矩阵集 $\{\boldsymbol{U}_i^{(k)}\}_{i=D+1}^{N_k}$ 的优化问题:

$$\max_{\{\boldsymbol{U}_i^{(k)}\}_{i=D+1}^{N_k}} \left\| \mathcal{P}^{\mathrm{T}}(\boldsymbol{\mathcal{F}}_k, \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}) \right\|_F^2. \tag{5}$$

假设 $\{\boldsymbol{U}_i^{(k)}\}_{i=D+1, i\neq n}^{N_k}$ 也为固定的, 则最大化问题 (5) 可转化为 $N_k - D$ 个关于因子矩阵 $\boldsymbol{U}_n^{(k)}$ 的优化问题:

$$\max_{\boldsymbol{U}_n^{(k)}} \left\| \boldsymbol{U}_n^{(k)\mathrm{T}} \left[\mathcal{P}_n^{\mathrm{T}}(\boldsymbol{\mathcal{F}}_k, \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}) \right]_{(n)} \right\|_F^2. \tag{6}$$

问题 (6) 可以利用 SVD 求解:

$$\left[\mathcal{P}_n^{\mathrm{T}}(\boldsymbol{\mathcal{F}}_k, \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}) \right]_{(n)} = \boldsymbol{P}_n^{(k)} \boldsymbol{\Lambda}_n^{(k)} \boldsymbol{Q}_n^{(k)\mathrm{T}}, \quad \boldsymbol{U}_n^{(k)} = \boldsymbol{P}_n^{(k)},$$

其中,

$$\boldsymbol{P}_n^{(k)} \in \mathbb{R}^{I_n \times r_n}, \quad \boldsymbol{Q}_n^{(k)} \in \mathbb{R}^{\prod_{i=1, i\neq n}^{N_k} r_i \times r_n},$$

且

$$(\boldsymbol{P}_n^{(k)})^{\mathrm{T}}\boldsymbol{P}_n^{(k)} = (\boldsymbol{Q}_n^{(k)})^{\mathrm{T}}\boldsymbol{Q}_n^{(k)} = \boldsymbol{I}_{r_n},$$

其中, $\boldsymbol{I}_{r_n}$ 为 r_n 阶单位矩阵.

计算共享因子集 $\{\boldsymbol{V}_d\}_{d=1}^{D}$: 对于固定的 $\{\boldsymbol{\mathcal{F}}_k\}$ 和非共享因子矩阵集 $\{\boldsymbol{U}_i^{(k)}\}_{i=D+1}^{N_k}$, 最大化问题 (3) 的目标函数可表示为关于共享因子矩阵集 $\{\boldsymbol{V}_d\}_{d=1}^{D}$ 的优化问题:

$$\max_{\{\boldsymbol{V}_d\}_{d=1}^{D}} \sum_{k=1}^{M} \left\| \mathcal{P}^{\mathrm{T}}(\boldsymbol{\mathcal{F}}_k, \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}) \right\|_F^2. \tag{7}$$

假设 $\{\boldsymbol{V}_d\}_{d=1,d\neq n}^{D}$ 也为固定的, 则最大化问题 (7) 可转化为 D 个关于非共享因子矩阵 $\boldsymbol{V}_n$ 的优化问题:

$$\max_{\boldsymbol{V}_n} \sum_{k=1}^{M} \left\| \boldsymbol{V}_n^{\mathrm{T}} \left[\mathcal{P}_n^{\mathrm{T}}(\boldsymbol{\mathcal{F}}_k, \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}) \right]_{(n)} \right\|_F^2. \tag{8}$$

令

$$\boldsymbol{S}_n^{(k)} = \left[\mathcal{P}_n^{\mathrm{T}}(\boldsymbol{\mathcal{F}}_k, \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}) \right]_{(n)},$$

则优化问题 (8) 中的目标函数可表示为

$$f_{\boldsymbol{V}_n} = \sum_{k=1}^{M} \left\| \boldsymbol{V}_n^{\mathrm{T}} \boldsymbol{S}_n^{(k)} \right\|_F^2 = \sum_{k=1}^{M} \mathrm{tr}\left(\boldsymbol{S}_n^{(k)\mathrm{T}} \boldsymbol{V}_n \boldsymbol{V}_n^{\mathrm{T}} \boldsymbol{S}_n^{(k)} \right),$$

进而得出 $f_{\boldsymbol{V}_n}$ 关于共享因子矩阵 $\boldsymbol{V}_n$ 的梯度:

$$\frac{\partial f_{\boldsymbol{V}_n}}{\partial \boldsymbol{V}_n} = 2\sum_{k=1}^{M} \boldsymbol{S}_n^{(k)} \boldsymbol{S}_n^{(k)\mathrm{T}} \boldsymbol{V}_n.$$

最终可以借助 Poblano 工具箱 [18], 采用非线性共轭梯度 (NCG) 法求解最优共享因子矩阵 $\boldsymbol{V}_n$.

计算核心张量 $\{\boldsymbol{\mathcal{G}}_k\}$: 将式 (4) 代入问题 (2) 的目标函数, 可得关于 $\{\boldsymbol{\mathcal{G}}_k\}$ 的目标函数 $f_{\boldsymbol{G}_k}$:

$$f_{\boldsymbol{G}_k} = \sum_{k=1}^{M} \left\| \boldsymbol{\mathcal{W}}_{\Omega}^{(k)} \cdot \left[\boldsymbol{\mathcal{X}}_k - \mathcal{P}(\boldsymbol{\mathcal{G}}_k, \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}) \right] \right\|_F^2.$$

因此, 若固定共享因子矩阵集 $\{\boldsymbol{V}_d\}_{d=1}^{D}$ 和非共享因子矩阵集 $\{\boldsymbol{U}_i^{(k)}\}_{i=D+1}^{N_k}$, 最小化问题 (2) 等价于如下关于核心张量 $\{\boldsymbol{\mathcal{G}}_k\}$ 的无约束优化问题:

$$\min_{\{\boldsymbol{\mathcal{G}}_k\}} f_{\boldsymbol{G}_k}.$$

与共享因子集 $\{\boldsymbol{V}_d\}_{d=1}^{D}$ 的计算类似, 可以利用非线性共轭梯度 (NCG) 法求解核心张量 $\{\boldsymbol{\mathcal{G}}_k\}$, $f_{\boldsymbol{\mathcal{G}}_k}$ 关于核心张量 $\{\boldsymbol{\mathcal{G}}_k\}$ 的梯度为

$$\frac{\partial f_{\boldsymbol{\mathcal{G}}_k}}{\partial \boldsymbol{\mathcal{G}}_k} = -2\mathcal{P}^{\mathrm{T}} \left[\boldsymbol{\mathcal{W}}_{\Omega}^{(k)} \cdot \left(\mathcal{P}(\boldsymbol{\mathcal{G}}_k, \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}) - \boldsymbol{\mathcal{X}}_k \right), \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k} \right].$$

令式 (2) 中的目标函数在第 t 次迭代时的值为 Obj_t, 将算法的终止条件设定为

$$\mathrm{Obj}_{t+1} - \mathrm{Obj}_t < \frac{\epsilon}{M} \sum_{i=1}^{M} \|\boldsymbol{\mathcal{F}}_i\|_F^2.$$

算法 1 SF-MTC 算法

Require: M 来源待填充张量 $\{\boldsymbol{\mathcal{X}}_k\}_{k=1}^{M}$;
Ensure: 完整张量 $\{\hat{\boldsymbol{\mathcal{X}}}_k\}_{k=1}^{M}$;
  利用 HOSVD 算法初始化 $\{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}$ 和 $\{\boldsymbol{\mathcal{G}}_k\}$;
  while not converged **do**
    for $n = 1 : D$ **do**
      计算共享因子矩阵 $\boldsymbol{V}_n$;
    end for
    for $k = 1 : M$ **do**
      计算 $\boldsymbol{\mathcal{F}}_k = \boldsymbol{\mathcal{W}}_{\Omega^C}^{(k)} \cdot \mathcal{P}(\boldsymbol{\mathcal{G}}_k, \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}) + \boldsymbol{\mathcal{W}}_{\Omega}^{(k)} \cdot \boldsymbol{\mathcal{X}}_k$;
      for $n = D+1 : N_k$ **do**
        计算非共享因子矩阵 $\boldsymbol{U}_n^{(k)}$;
      end for
      计算核心张量 $\boldsymbol{\mathcal{G}}_k$;
    end for
  end while
  for $k = 1 : M$ **do**
    $\hat{\boldsymbol{\mathcal{X}}}_k \leftarrow \boldsymbol{\mathcal{F}}_k$;
  end for
  return $\{\hat{\boldsymbol{\mathcal{X}}}_k\}_{k=1}^{M}$

表 1 张量运算的计算复杂度

Table 1 The computational complexity of tensor operation

Tensor operation	Computational complexity
$\mathcal{P}^{\mathrm{T}}(\boldsymbol{\mathcal{F}}_k, \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k})$	$O(\sum_{n=1}^{N_k} r_n \prod_{i=1}^{N_k} I_i^{(k)})$
$\mathcal{P}(\boldsymbol{\mathcal{G}}_k, \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k})$	$O(\sum_{n=1}^{N_k} I_n^{(k)} \prod_{i=1}^{N_k} r_i)$
Compute $\{\boldsymbol{U}_i^{(k)}\}_{i=D+1}^{N_k}$ via SVD	$O(\sum_{k=1}^{M} \sum_{n=D+1}^{N_k} I_n^{(k)} \prod_{i=1}^{N_k} r_i)$
Compute $\{\boldsymbol{V}_d\}_{d=1}^{D}$ via CG	$O(\sum_{n=1}^{D} I_n^2 \prod_{i=1}^{N_k} r_i)$
Compute $\{\boldsymbol{\mathcal{G}}_k\}$ via CG	$O(\sum_{k=1}^{M} \sum_{n=1}^{N_k} r_n \prod_{i=1}^{N_k} I_i^{(k)})$

综上所述, 基于共享因子的多源张量填充 (SF-MTC) 算法的具体过程见算法 1.

若非线性共轭梯度法在计算核心张量 $\{\boldsymbol{\mathcal{G}}_k\}$ 时收敛, 即 $\frac{\partial f_{\boldsymbol{\mathcal{G}}_k}}{\partial \boldsymbol{\mathcal{G}}_k} = 0$, 则式 (2) 关于 $\{\boldsymbol{\mathcal{G}}_k\}$ 的 KKT 条件:

$$\mathcal{P}^{\mathrm{T}}\left[\boldsymbol{\mathcal{W}}_{\Omega}^{(k)} \cdot \left(\mathcal{P}(\boldsymbol{\mathcal{G}}_k, \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}) - \boldsymbol{\mathcal{X}}_k\right), \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k}\right] = 0$$

成立. 又因 $\{\boldsymbol{\mathcal{F}}_k\}$ 的更新过程由式 (2) 的 KKT 条件导出, 故算法 1 能够收敛到式 (2) 的 KKT 点.

5.2 计算复杂度分析

在算法 1 的每步迭代中, 均需要进行张量运算 $\mathcal{P}^{\mathrm{T}}(\boldsymbol{\mathcal{F}}_k, \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k})$ 和 $\mathcal{P}(\boldsymbol{\mathcal{G}}_k, \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k})$, 其计算复杂度见表 1. 对于每一个来源的张量, 其非共享因子集 $\{\boldsymbol{U}_i^{(k)}\}_{i=D+1}^{N_k}$, 需要利用 SVD 求解 $N_k - D$ 次, 由表 1 可知其计算复杂度与张量运算 $\mathcal{P}(\boldsymbol{\mathcal{G}}_k, \{\widetilde{\boldsymbol{U}}_n^{(k)}\}_D^{N_k})$ 类似. 由于 M 个张量在前 D 个模式上具有共享因子, 因此共享因子矩阵集 $\{\boldsymbol{V}_d\}_{d=1}^{D}$ 的计算与 $\{\boldsymbol{V}_d\}_{d=1}^{D}$ 和 $\{\boldsymbol{\mathcal{G}}_k\}$ 的计算不同, 其无需重复计算 M 次 (见表 1).

6 实验结果与分析

本文中多源张量填充的目的是同时利用多源的不完整张量 $\{\boldsymbol{\mathcal{X}}_k\} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{N_k}}$ 来恢复真实张量 $\boldsymbol{\mathcal{D}}_k$. 假设 $\overline{\boldsymbol{\mathcal{X}}}_k$ ($\boldsymbol{\mathcal{X}}_k = \boldsymbol{\mathcal{W}}_\Omega^{(k)} \cdot \overline{\boldsymbol{\mathcal{X}}}_k$) 可分解为一个秩 -$(r_1, r_2, \ldots, r_{N_k})$ 的真实张量 $\boldsymbol{\mathcal{D}}_k$ 和一个噪声张量 $\boldsymbol{\mathcal{E}}_k$:

$$\overline{\boldsymbol{\mathcal{X}}}_k = \boldsymbol{\mathcal{D}}_k + \boldsymbol{\mathcal{E}}_k,$$

则张量填充的性能可以利用填充后张量 $\hat{\boldsymbol{\mathcal{X}}}_k$ 的相对平方误差 (RSE) 来度量:

$$\mathrm{RSE}(\hat{\boldsymbol{\mathcal{X}}}_k) = \frac{\|\boldsymbol{\mathcal{D}}_k - \hat{\boldsymbol{\mathcal{X}}}_k\|_F}{\|\boldsymbol{\mathcal{D}}_k\|_F},$$

简记为 RSEk = RSE($\hat{\boldsymbol{\mathcal{X}}}_k$). 最后, 将依据不同的采样比例 (SR) 进行随机采样, 得到待填充张量.

本节通过在人工数据集和实际数据集上的实验验证所提出的 SF-MTC 的精确性和有效性. 实验中的算法利用 Matlab 的张量工具箱 (Tensor Toolbox Version 2.6) 实现.

6.1 人工数据集

首先, 构造 M 个低 n- 秩张量 $\{\boldsymbol{\mathcal{D}}_k\}_{k=1}^M$ 作为真实的多源张量:

$$\{\widetilde{\boldsymbol{B}}_n^{(k)}\}_J^{N_k} = \{\boldsymbol{A}_1, \ldots, \boldsymbol{A}_J, \boldsymbol{B}_{J+1}^{(k)}, \boldsymbol{B}_{J+2}^{(k)}, \ldots, \boldsymbol{B}_{N_k}^{(k)}\}, \quad \boldsymbol{\mathcal{D}}_k = \mathcal{P}(\boldsymbol{\mathcal{T}}_k, \{\widetilde{\boldsymbol{B}}_n^{(k)}\}_J^{N_k}) \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_{N_k}}. \tag{9}$$

如文献 [5] 中人工张量数据集的设定, 将张量在各个模式上的维数设置为相同的, 即 $I_1 = I_2 = \cdots = I_{N_k}$, 并且张量真实的 n- 秩为 $\hat{r}_1 = \hat{r}_2 = \cdots = \hat{r}_{N_k}$. 然后, 在真实张量 $\{\boldsymbol{\mathcal{D}}_k\}_{k=1}^M$ 中加入噪声, 得到一组张量 $\{\overline{\boldsymbol{\mathcal{X}}}_k\}_{k=1}^M$:

$$\overline{\boldsymbol{\mathcal{X}}}_k = \boldsymbol{\mathcal{D}}_k + \mathrm{PGN}_k \frac{\|\boldsymbol{\mathcal{D}}_k\|_F}{\|\boldsymbol{\mathcal{N}}_k\|_F} \boldsymbol{\mathcal{N}}_k,$$

其中, PGN$_k$ 为加入 Gauss 噪声的比例, $\boldsymbol{\mathcal{N}}_k$ 为元素服从标准正态分布的噪声张量. 最后, 依据不同的采样比例 (SR) 对 $\{\overline{\boldsymbol{\mathcal{X}}}_k\}_{k=1}^M$ 进行随机采样, 得到待填充张量 $\{\boldsymbol{\mathcal{X}}_k\}_{k=1}^M$.

本节将提出的 SF-MTC 与当前主流的张量填充方法进行比较, 包括 CP-WOPT [11], Tuck-WOPT [12], FaLRTC 和 HaLRTC [5]. 依据文献 [12] 中的实验设定, 真实 n- 秩 $\hat{r}_n$ 的估计 r_n 设定为 $r_n = \hat{r}_n + 5$. 此外, 取算法终止条件中的 $\epsilon = 10^{-3}$, 并将每个来源张量的 Gauss 噪声比例均设定为 PGN$_k = 20\%$.

6.1.1 精度与效率

令 $M = 3, J = 1$, 构造 3 个具有 1 个共享因子矩阵的 3 阶张量 $\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2, \boldsymbol{\mathcal{D}}_3 \in \mathbb{R}^{60 \times 60 \times 60}$ 作为多源的真实张量, 其真实的 n- 秩为 $\hat{r}_1 = \hat{r}_2 = \hat{r}_3 = 10$, 在该人工张量数据集上填充的误差及运行时间见表 2. 类似地, 构造 3 个具有 1 个共享因子矩阵的 4 阶张量 $\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2, \boldsymbol{\mathcal{D}}_3 \in \mathbb{R}^{50 \times 50 \times 50 \times 50}$, 真实的 n- 秩为 $\hat{r}_1 = \hat{r}_2 = \hat{r}_3 = \hat{r}_4 = 10$, 采用不同方法的填充结果见表 3.

在计算精度方面, 由表 2 和 3 中的误差结果可以看出, SF-MTC 在 $D = 0$ (不考虑共享因子) 时, 其填充精度高于 CP-WOPT, 且与 Tuck-WOPT 的填充精度相似, 这是由于在 $D = 0$ 时, SF-MTC 可以看成一种利用 Tucker 分解填充多个张量的方法. 而 SF-MTC 在 $D = 1$ (设置 1 个共享因子) 时, 其填充精度高于其他算法, 这说明了借助共享因子矩阵可以挖掘出多源张量的潜在共享信息, 进而提升多源张量填充的精度.

表 2 人工数据集 $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3 \in \mathbb{R}^{60\times60\times60}$ 上张量填充方法比较

Table 2 Comparison for tensor completion on synthetic tensors $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3 \in \mathbb{R}^{60\times60\times60}$

SR	CP-WOPT				Tuck-WOPT				SF-MTC ($D=0$)			
	RSE1	RSE2	RSE3	Time (s)	RSE1	RSE2	RSE3	Time (s)	RSE1	RSE2	RSE3	Time (s)
20%	0.1520	0.1603	0.1560	45.32	0.1326	0.1481	0.1332	36.23	0.1294	0.1353	0.1287	32.23
30%	0.1344	0.1323	0.1331	42.41	0.1139	0.1203	0.1145	40.22	0.1052	0.1060	0.1009	29.70
40%	0.1262	0.1328	0.1159	34.43	0.0932	0.0945	0.0873	36.81	0.0953	0.0929	0.0841	26.29
SR	FaLRTC				HaLRTC				SF-MTC ($D=1$)			
	RSE1	RSE2	RSE3	Time (s)	RSE1	RSE2	RSE3	Time (s)	RSE1	RSE2	RSE3	Time (s)
20%	0.1345	0.1374	0.1281	24.55	0.1201	0.1323	0.1235	34.52	0.0870	0.0890	0.0845	25.86
30%	0.1119	0.1134	0.1073	24.23	0.1022	0.1020	0.1062	31.45	0.0523	0.0521	0.0515	23.58
40%	0.0730	0.0821	0.0787	18.45	0.0788	0.0712	0.0736	25.39	0.0424	0.0435	0.0413	19.82

表 3 人工数据集 $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3 \in \mathbb{R}^{50\times50\times50\times50}$ 上张量填充方法比较

Table 3 Comparison for tensor completion on synthetic tensors $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3 \in \mathbb{R}^{50\times50\times50\times50}$

SR	CP-WOPT				Tuck-WOPT				SF-MTC ($D=0$)			
	RSE1	RSE2	RSE3	Time (s)	RSE1	RSE2	RSE3	Time (s)	RSE1	RSE2	RSE3	Time (s)
20%	0.2359	0.2387	0.2243	1834.34	0.1724	0.1634	0.1765	1734.53	0.1637	0.1657	0.1684	1502.24
30%	0.2062	0.2032	0.1977	1523.51	0.1642	0.1582	0.1643	1523.31	0.1556	0.1570	0.1578	1264.51
40%	0.1881	0.1834	0.1798	1375.13	0.1506	0.1515	0.1475	1262.85	0.1493	0.1532	0.1562	1009.18
SR	FaLRTC				HaLRTC				SF-MTC ($D=1$)			
	RSE1	RSE2	RSE3	Time (s)	RSE1	RSE2	RSE3	Time (s)	RSE1	RSE2	RSE3	Time (s)
20%	0.1631	0.1643	0.1689	1324.24	0.1604	0.1583	0.1524	1372.34	0.1421	0.1432	0.1440	1200.53
30%	0.1567	0.1503	0.1512	962.36	0.1468	0.1483	0.1435	1134.55	0.1109	0.1134	0.1170	974.49
40%	0.1434	0.1356	0.1457	834.12	0.1379	0.1234	0.1287	1003.45	0.0912	0.1021	0.1013	898.38

在计算效率方面, 由于 SF-MTC 算法在求解因子矩阵时引入了 SVD 方法, 其计算复杂度要低于 Tuck-WOPT 中用到的 NCG 方法 (见表 1). 表 2 和 3 中, SF-MTC 在 $D=0$ 时的耗时少于 Tuck-WOPT 方法耗时的 3 倍也说明了这一点. 而 SF-MTC 在 $D=1$ 时较 $D=0$ 时的耗时更少, 这是由于通过引入共享因子矩阵, 多个张量在共享模式上只需求解一次因子矩阵. 由表 3 可知, 在 $D=1$ 时, SF-MTC 填充 3 个高阶张量时的耗时远小于其他方法的 3 倍, 这说明了具有共享因子的 SF-MTC 在填充高阶张量时可以获得更高的计算效率.

当 $\boldsymbol{\mathcal{X}}_1$ 和 $\boldsymbol{\mathcal{X}}_2$ 的采样比例 (SR) 不同时, 记 $\boldsymbol{\mathcal{X}}_1$ 和 $\boldsymbol{\mathcal{X}}_2$ 的采样比例分别为 SR1 和 SR2. 取 $M=2, J=1$, 构造人工数据集 $\mathcal{D}_1, \mathcal{D}_2 \in \mathbb{R}^{50\times50\times50}$, 比较不同采样比例下张量填充方法的精度和效率. 从表 4 的结果可以看出, SF-MTC 较 CP-WOPT 与 Tuck-WOPT 具有较高的填充精度和计算效率. 当其中一个来源的张量具有较大的采样比例时, SF-MTC 对多源张量填充的精度提升较为明显.

6.1.2 共享因子的作用

在人工张量数据集上, 针对多源张量具有单个共享因子的情形, $D=1$ 时的 SF-MTC 获得了较好的填充精度和计算效率. 这一节研究不同共享因子情形下, SF-MTC 的填充性能. 下面在不引入 Gauss 噪声的情形下, 构造人工张量数据, 研究共享因子对多源张量填充方法 SF-MTC 的影响.

表 4 人工数据集 $\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2 \in \mathbb{R}^{50\times50\times50}$ 上张量填充方法比较

Table 4 Comparison for tensor completion on synthetic tensors $\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2 \in \mathbb{R}^{50\times50\times50}$

SR1	SR2	CP-WOPT			Tuck-WOPT			SF-MTC ($D=1$)		
		RSE1	RSE2	Time (s)	RSE1	RSE2	Time (s)	RSE1	RSE2	Time (s)
20%	20%	0.1345	0.1283	30.97	0.1134	0.0945	28.53	0.0720	0.0678	10.95
30%	20%	0.1034	0.1235	28.56	0.0998	0.0955	23.93	0.0620	0.0636	9.82
40%	20%	0.0882	0.1302	25.29	0.0623	0.0887	22.57	0.0518	0.0605	9.13
50%	20%	0.0813	0.1280	24.38	0.0542	0.0968	19.43	0.0357	0.0537	8.06
60%	20%	0.0582	0.1284	23.53	0.0434	0.0937	20.92	0.0152	0.0486	7.24
70%	20%	0.0503	0.1257	18.46	0.0227	0.0960	17.63	0.0120	0.0441	6.45
80%	20%	0.0357	0.1258	16.35	0.0104	0.0936	15.56	0.0073	0.0422	6.59

首先, 在式 (9) 中取 $J=0$, 构造 4 个无共享因子矩阵的 5 阶张量 $\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2, \boldsymbol{\mathcal{D}}_3, \boldsymbol{\mathcal{D}}_4 \in \mathbb{R}^{10\times10\times10\times10\times10}$ 作为多源张量, 其真实的 n- 秩为 $\hat{r}_1 = \hat{r}_2 = \cdots = \hat{r}_5 = 3$. 分别取共享因子个数 $D=0,1,2$, 利用 SF-MTC 进行填充. 记 MRSE 为多个张量填充误差 RSE 的均值. 由图 1 中的填充结果可见, 如果多源张量数据没有共享的信息, 那么利用基于共享因子的张量填充方法在不同采样比例下并不能获得好的填充结果, 而在不考虑共享因子的情形下进行多张量填充可以获得较高的精度. 此时, 由于待填充的多源张量数据并不具有相关性, 共享因子的数量设置的越大, 填充的精度越低.

然后, 取 $J=1$, 按照上述设定构造 4 个具有 1 个共享因子的 5 阶张量, 分别取共享因子个数 $D=0,1,2$, 利用 SF-MTC 进行填充. 由图 2 可以看出, 对于具有共享因子的多源张量数据集, 通过选取适当的共享因子个数 D, 利用 SF-MTC 可以获得较高的填充精度. 此外, 在高采样比例时, 与不设置共享因子的情形相比较, 即使设置的共享因子数量多于真实值, 由于利用共享因子引入的相关性信息的作用, 利用 SF-MTC 进行多源张量填充, 也可以获得较高的填充精度.

最后, 构造 4 个具有 2 个共享因子的 5 阶张量, 利用 SF-MTC 进行填充. 由图 3 可见, 当 D 选取为张量数据集的真实共享因子个数时, 利用 SF-MTC 可以获得较高的填充精度. 而当 $D=1$ 时, 即使共享因子的数量低于真实值, 利用 SF-MTC 进行多源张量填充的精度仍然高于不设置共享因子的情形, 这验证了 SF-MTC 在具有相关性的多源张量数据集上的有效性.

在实际问题中, 若已知多源张量数据在某些模式上具有相关性, 则可直接将这些模式设置为共享模式; 若未知待填充张量数据的相关性, 则可将共享因子的数目设置为 1, 一方面, 若待填充张量数据在该模式上具有相关性, 则利用该方法可以提升填充精度, 另一方面, 若数据不具有相关性, 则不会对填充结果产生较大影响.

6.2 实际数据集

本节在医疗数据集、化学数据集上验证 SF-MTC 的精确性和有效性.

6.2.1 医疗数据集

在 OsiriX 医疗数据集上对 SF-MTC 的性能进行测试. OsiriX 医疗数据集中包含大量的 MRI 图像, 实验选取 BRAINIX 数据集, 该数据集中包含 22 张同一患者的脑部 MRI 图像, 从 BRAINIX 数据集中选择 3 张作为原始图像 $\{\boldsymbol{\mathcal{D}}_k\}_{k=1}^3$, 每一张的维数均为 288×288. 设定 SR 为 30%、PGN_k 为 5%, 构造待填充多源张量 $\{\boldsymbol{\mathcal{X}}_k\}_{k=1}^3$.

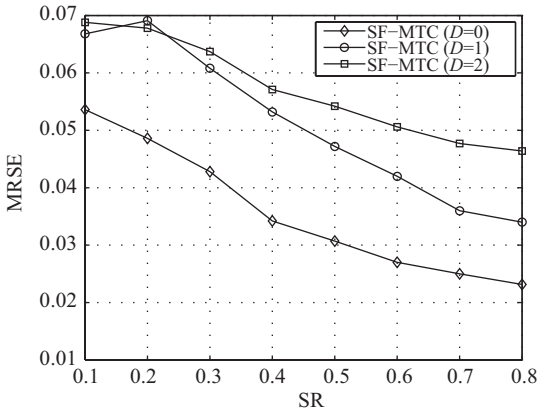

图 1 无共享因子

Figure 1 No shared factor

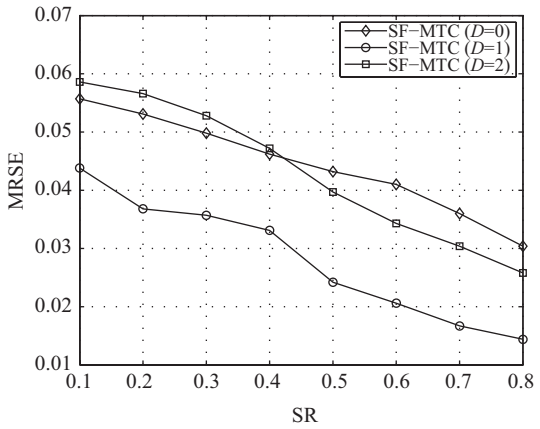

图 2 1 共享因子

Figure 2 1 shared factor

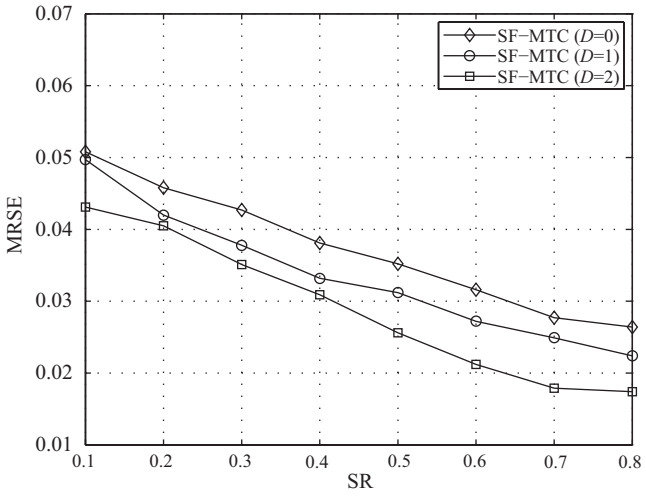

图 3 2 共享因子

Figure 3 2 shared factors

图 4(a), (e), (i) 分别为原始图像 $\{\boldsymbol{\mathcal{D}}_k\}_{k=1}^3$, 将对应的训练图像图 4(b), (f), (j) 分别作为待填充张量 $\{\boldsymbol{\mathcal{X}}_k\}_{k=1}^3$, 利用 HaLRTC, Tuck-WOPT 和 SF-MTC ($D=0,1$) 分别进行张量填充. 从图 4 的结果可以看出, 利用基于共享因子的 SF-MTC 得到的填充图像具有更小的误差, 并且由于 SF-MTC 对 3 个训练图像同时填充, 其计算效率要高于 Tuck-WOPT. 由表 5 的结果可知, SF-MTC ($D=1$) 在已知元素比例较低时, 较其他方法具有较高的填充精度, 验证了共享因子在多源多张量填充中的作用.

6.2.2 化学数据集

在化学数据集上将 SF-MTC 与已有方法进行对比. Amino Acids 数据集 ($5\times 201\times 61$) 由 5 个样本组成, 每个样本包含不同数量的酪氨酸、色氨酸、苯基丙氨酸. 实验中, 将 5 个样本作为多源张量进行填充. 将 Flow Injection 数据集 (12 (化学物质) × 100 (波长) × 89 (反应时间)) 中的数据划分为

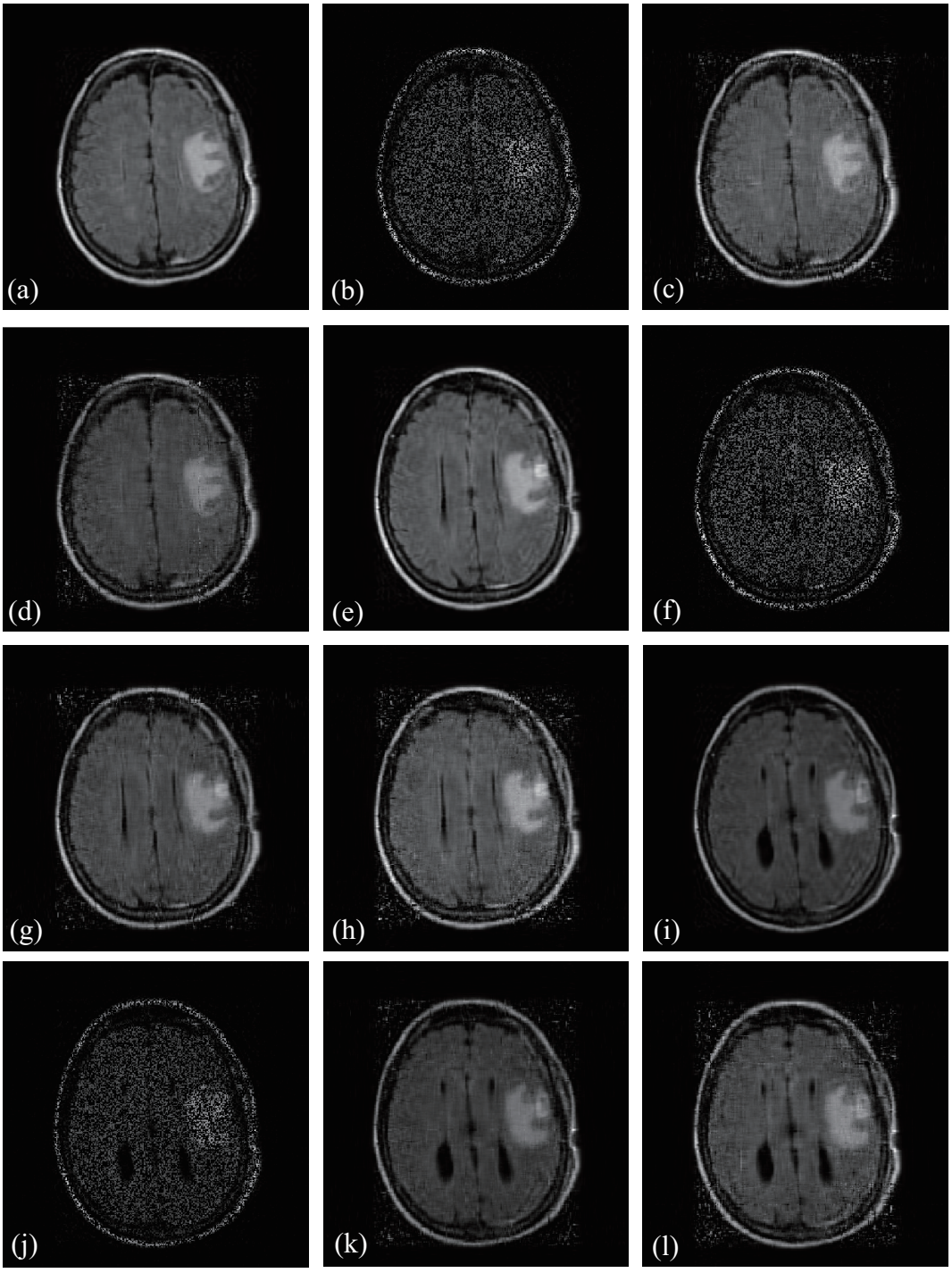

图 4 **BRAINIX** 数据集上的测试结果

Figure 4 Results on the BRAINIX dataset. (a), (e), (i) Original images; (b), (f), (j) training images; (c), (g), (k) results via SF-MTC ($D = 1$): RSE1 = 0.1738, RSE2 = 0.1923, RSE3 = 0.1913; (d), (h), (l) results via Tuck-WOPT: RSE1 = 0.2404, RSE2 = 0.2432, RSE3 = 0.2534

表 5 BRAINIX 数据集上张量填充方法比较

Table 5 Comparisons for tensor completion on the BRAINIX dataset

SR	HaLRTC			Tuck-WOPT			SF-MTC ($D=0$)			SF-MTC ($D=1$)		
	RSE1	RSE2	RSE3	RSE1	RSE2	RSE3	RSE1	RSE2	RSE3	RSE1	RSE2	RSE3
20%	0.2513	0.2733	0.2719	0.2631	0.2693	0.2746	0.2521	0.2620	0.2684	0.1903	0.2120	0.2138
30%	0.2432	0.2551	0.2522	0.2404	0.2432	0.2537	0.2323	0.2453	0.2468	0.1738	0.1923	0.1913
40%	0.1712	0.1804	0.1843	0.1734	0.1767	0.1864	0.1683	0.1784	0.1822	0.1345	0.1434	0.1457

表 6 化学数据集上张量填充方法比较

Table 6 Comparisons for tensor completion on the chemometrics dataset

	$D=0$	$D=1$	Tuck-Wopt	HaLRTC
	MRSE	MRSE	RSE ($\widetilde{\boldsymbol{\mathcal{X}}}$)	RSE ($\widetilde{\boldsymbol{\mathcal{X}}}$)
Amino Acids	0.0233	0.0194	0.0230	0.0244
Flow Injection	0.0403	0.0340	0.0423	0.0392
Sugar Process	0.0699	0.0629	0.0712	0.0691

3 个来源的张量, 每组的维数为 $4 \times 100 \times 89$. 从 Sugar Process 数据集中提取两个相关张量, 每个张量的维数为 $100 \times 571 \times 7$. 将采样比例设置为 SR = 30%, 利用 SF-MTC ($D = 0, 1$) 进行填充, 将多源多张量上的平均 RSE 记为 MRSE. 作为比较, 将上述多源张量合并为一个张量 $\widetilde{\boldsymbol{\mathcal{X}}}$, 利用 HaLRTC 和 Tuck-WOPT 进行填充. 表 6 中的结果说明, SF-MTC 在实际数据集上, 通过挖掘多源多张量的潜在相关性, 提高了张量填充的精度.

7 结语

本文基于多源数据的共享因子, 提出了一种对多个张量同时填充的方法. 该方法可以提取不同阶的张量数据在共享模式上的共同结构特征, 为张量的 Tucker 分解提供了低秩假设以外的结构信息, 提高了张量填充的精度. 由于共享因子集的建立减少了需要求解的因子矩阵的个数, 所提出的多个张量同时填充的方法也具有较高的计算效率. 实验进一步验证了所提出方法在处理多源相关张量数据时的精确性和高效性.

参考文献

1 He X F, Cai D, Niyogi P. Tensor subspace analysis. In: Advances in Neural Information Processing Systems 18, Vancouver, 2005. 499–506

2 Hao Z F, He L F, Chen B Q, et al. A linear support higher-order tensor machine for classification. IEEE Trans Image Process, 2013, 22: 2911–2920

3 Kim T K, Cipolla R. Canonical correlation analysis of video volume tensors for action categorization and detection. IEEE Trans Pattern Anal Mach Intell, 2009, 31: 1415–1428

4 Kolda T G, Bader B W. Tensor decompositions and applications. SIAM Rev, 2009, 51: 455–500

5 Liu J, Musialski P, Wonka P, et al. Tensor completion for estimating missing values in visual data. Trans Pattern Anal Mach Intell, 2013, 35: 208–220

6 Candès E J, Recht B. Exact matrix completion via convex optimization. Found Comput Math, 2009, 9: 717–772

7 Recht B, Fazel M, Parrilo P A. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. SIAM Rev, 2010, 52: 471–501

8 Shamir O, Shalev-Shwartz S. Matrix completion with the trace norm: learning, bounding, and transducing. J Mach Learn Res, 2014, 15: 3401–3423
9 Tomioka R, Hayashi K, Kashima H. On the extension of trace norm to tensors. In: Proceedings of NIPS Workshop on Tensors, Kernels, and Machine Learning, Vancouver, 2010. 1–4
10 Liu Y Y, Shang F H, Fan W, et al. Generalized higher-order orthogonal iteration for tensor decomposition and completion. In: Advances in Neural Information Processing Systems 27, Montréal, 2014. 1763–1771
11 Acar E, Dunlavy D M, Kolda T G, et al. Scalable tensor factorizations for incomplete data. Chemometr Intell Lab Syst, 2011, 106: 41–56
12 Filipović M, Jukić A. Tucker factorization with missing data with application to low-n-rank tensor completion. Multidim Syst Signal Process, 2015, 26: 677–692
13 Narita A, Hayashi K, Tomioka R, et al. Tensor factorization using auxiliary information. Data Min Knowl Discov, 2012, 25: 298–324
14 Acar E, Rasmussen M A, Savorani F, et al. Understanding data fusion within the framework of coupled matrix and tensor factorizations. Chemometr Intell Lab Syst, 2013, 129: 53–63
15 Acar E, Dunlavy D M, Kolda T G, et al. Scalable tensor factorizations with missing data. In: Proceedings of the 10th SIAM International Conference on Data Mining, Columbus, 2010. 701–712
16 de Lathauwer L, de Moor B, Vandewalle J. On the best rank-1 and rank-$(r_1, r_2, \ldots, r_n)$ approximation of higher-order tensors. SIAM J Matrix Anal Appl, 2000, 21: 1324–1342
17 Cichocki A, Zdunek R, Phan A H, et al. Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation. Chichester: John Wiley & Sons, 2009
18 Dunlavy D M, Kolda T G, Acar E. Poblano v1.0: a Matlab toolbox for gradient-based optimization. Technical Report SAND2010-1422. 2010

Multi-tensor completion with shared factors from multiple sources

Xiao ZHANG, Qinghua HU & Shizhong LIAO*

School of Computer Science and Technology, Tianjin University, Tianjin 300350*, China*
*E-mail: szliao@tju.edu.cn

Abstract Tensor completion has been widely applied in data mining, machine learning, and biomedical signal processing. Most existing tensor completion methods recover a single tensor with low-rank assumption and exhibit low accuracy because of the complicated structures of tensor data. We address this issue by proposing a completion method based on a multiple sources tensors. Firstly, we formulate the least-squares problems for multiple sources tensor completion by using a Tucker decomposition. Then, we assume that tensors from different sources have common information at the shared modes, and create the shared factor matrices set of the Tucker decomposition from which the common latent structure from the shared modes can be extracted. Further, we develop a multiple sources tensor completion method with shared factors (SF-MTC). Finally, we utilize a nonlinear conjugate gradient method and singular value decomposition to compute the factor matrices and core tensors of the Tucker decomposition, and recover the tensors from multiple sources simultaneously. We also analyze the computational complexity of the SF-MTC. Experimental results obtained with both synthetic and real data demonstrate that the SF-MTC is efficient for recovering multiple tensors, and accurate on multiple related tensor datasets.

Keywords tensor completion, Tucker decomposition, shared modes, nonlinear conjugate gradient method, singular value decomposition

Xiao ZHANG was born in 1989. He received his M.S. degree in computational mathematics from Northwest Polytechnical University, Shaanxi, China, in 2015. Currently, he is a Ph.D. candidate with the School of Computer Science and Technology, Tianjin University, Tianjin, China. His main research interests include tensor learning, online learning, and kernel methods.

Shizhong LIAO was born in 1964. He received his Ph.D. degree in Computer Science from Tsinghua University, Beijing, China, in 1997. Currently, he is a professor and Head of the Computer Science Department with the School of Computer Science and Technology, Tianjin University, Tianjin, China. His research interests include machine learning, artificial intelligence, theoretical computer science.

Qinghua HU was born in 1976. He received his B.S., M.E., and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 1999, 2002, and 2008, respectively. He started working at the Harbin Institute of Technology in 2006, and was a postdoctoral fellow with the Hong Kong Polytechnic University from 2009 to 2011. In 2012, he joined Tianjin University. At present, he is a professor and the vice dean of the School of Computer Science and Technology, Tianjin University, and the director of the Lab of Pattern Analysis and and Computational Intelligence (PANDIT). His research interests include machine learning, pattern analysis, and uncertainty in artificial intelligence.