

# 基于超图正则化非负张量链分解的聚类分析

戴浩磊, 黄永慧, 周郭许

(广东工业大学 自动化学院, 广州 510006)

**摘要:** 非负张量链分解作为一种重要的张量分解模型, 可保留数据内部结构信息, 广泛应用于高维数据的特征提取和表示。从流形学习角度出发, 高维数据信息通常潜在于低维空间的非线性流形结构中, 然而现有图学习理论只能建模对象间的成对关系, 很难准确刻画具有复杂流形结构的高维数据的相似关系。引入超图学习, 提出一种超图正则化非负张量链(HGNTT)分解方法, 在高维数据中提取低维表示的同时通过构建超图描述样本数据间的高阶关系, 从而保留非线性流形结构, 同时采用乘法更新方法对HGNTT模型进行优化求解并证明其收敛性。在ORL和Faces95这两个公开数据集上的聚类实验结果表明, 相比于NMF、GNMF等方法, HGNTT方法的聚类准确率和归一化互信息分别提升了1.2%~7.6%和0.2%~3.0%, 验证了HGNTT方法的有效性。

**关键词:** 非负张量链分解; 特征提取; 超图学习; 乘法更新方法; 聚类分析

开放科学(资源服务)标志码(OSID):



中文引用格式: 戴浩磊, 黄永慧, 周郭许. 基于超图正则化非负张量链分解的聚类分析[J]. 计算机工程, 2023, 49(6): 81-89.

英文引用格式: DAI H L, HUANG Y H, ZHOU G X. Clustering analysis based on hyper-graph regularized non-negative tensor train decomposition[J]. Computer Engineering, 2023, 49(6): 81-89.

## Clustering Analysis Based on Hyper-graph Regularized Non-Negative Tensor Train Decomposition

DAI Haolei, HUANG Yonghui, ZHOU Guoxu

(School of Automation, Guangdong University of Technology, Guangzhou 510006, China)

**[Abstract]** Non-negative Tensor Train (NTT) decomposition, as an important tensor decomposition model, can preserve the internal structure information of data and is widely used in feature extraction and representation tasks of high-dimensional data. From the perspective of manifold learning, high-dimensional data information is usually latent in the nonlinear manifold structure in low-dimensional space. However, existing graph learning theories can only model pairwise relationships between objects, and accurately portraying the similar relationships of high-dimensional data with a complex manifold structure is difficult. By introducing hyper-graph learning, this study proposes a Hyper-Graph regularized Non-negative Tensor Train (HGNTT) decomposition method for extracting low-dimensional representations from high-dimensional data while describing the higher-order relationships between sample data points by constructing hyper-graphs, thereby preserving the nonlinear manifold structure. Moreover, a Multiplicative Update (MU) method is used to optimally solve the HGNTT model and prove its convergence. Clustering experiments on two publicly available datasets, ORL and Faces95, show that the clustering accuracy and Normalized Mutual Information (NMI) of the HGNTT method improves by 1.2%-7.6% and 0.2%-3.0%, respectively, compared with those of NMF and GNMF, thereby validating the effectiveness of the HGNTT method.

**[Key words]** Non-negative Tensor Train (NTT) decomposition; feature extraction; hyper-graph learning; Multiplicative Update (MU) method; clustering analysis

DOI: 10.19678/j.issn.1000-3428.0064740

### 0 概述

随着科学技术的迅速发展, 高维数据如今已存

在于人们生活的方方面面, 如多媒体数据、网络文档数据、脑电图数据等。这些数据具有高维特性, 结构复杂, 内部包含丰富的物理信息。传统矩阵分解方

基金项目: 国家自然科学基金(62073087); 广东省重点领域研发计划(2019B010154002)。

作者简介: 戴浩磊(1997—), 男, 硕士研究生, 主研方向为高阶张量数据处理、高性能计算; 黄永慧(通信作者), 讲师、博士; 周郭许, 教授、博士。

收稿日期: 2022-05-18 修回日期: 2022-07-28 E-mail: msh\_huizi@gdut.edu.cn

法<sup>[1]</sup>在处理高维数据时,不能较好地捕捉高维数据的多线性结构,从而降低了其获取的低维特征的有效性。张量<sup>[2]</sup>作为向量与矩阵的高阶推广,在保持高维数据的多线性结构方面具有一定优势,应用前景广阔。与此同时,受益于多重线性代数的发展,使用张量分解的数据分析技术在选择与数据属性相匹配的约束条件和提取数据中更普遍的潜在成分方面相较于矩阵分解方法更具优势。

经典的张量分解模型包括 CANDEC-OMP/PARAFAC(CP)分解<sup>[3-4]</sup>和 Tucker 分解<sup>[5]</sup>模型。CP 分解将一个张量分解为多个秩-1 张量之和的形式。Tucker 分解将一个张量分解为一个核张量和多个因子矩阵的多线性乘积的形式<sup>[6]</sup>。基于这两种模型,研究人员进行了许多研究并提出了一系列改进模型,如完全贝叶斯 CP(Fully Bayesian CP, FBCP)分解<sup>[7]</sup>、半监督非负 Tucker 分解(Semi-supervised Non-negative Tucker Decomposition, SNTD)<sup>[8]</sup>和正交非负 Tucker 分解(Orthogonal Non-negative Tucker Decomposition, ONTD)<sup>[9]</sup>等。然而,这两种模型本身也存在局限性:CP 分解虽然能提供紧凑的张量表示,但很难找到最优的低秩结构,尤其在高阶情况下;Tucker 分解虽然在高阶数据拟合上具有优越性能,但参数量会随着阶数的增加成指数增长,在实际应用中容易引起维度灾难。为了解决这些问题,张量链(Tensor Train, TT)<sup>[10-12]</sup>分解被提出并得到广泛应用,将高阶张量分解为一系列二阶或三阶核心张量的多线性乘积,结构简单且容易实现。此外,由于该模型的参数量与张量阶数成线性关系,具有较强的数据压缩能力,因此不受维度诅咒的影响,是处理高阶张量的理想模型。受非负矩阵分解(Negative Matrix Factorization, NMF)的启发,为了从张量数据中更好地发现基于局部特征表示,文献[13-15]提出非负张量链(Non-negative Tensor Train, NTT)分解模型并得到广泛应用。

近年来,为了进一步增强非负矩阵或张量分解方法在特征提取上的表示能力,流形学习<sup>[16-18]</sup>受到了研究人员的关注,他们认为部分高维数据其实是由低维的流形结构嵌入高维空间得到,流形学习的目的是将高维空间中的数据映射回低维空间,使得到的低维数据能够体现高维空间中数据的部分本质结构特征。经典的流形学习算法有局部线性嵌入(Locally Linear Embedding, LLE)<sup>[19]</sup>、等距特征映射(ISOMAP)<sup>[20]</sup>和拉普拉斯特征映射(Laplacian Eigenmap, LE)<sup>[21]</sup>等。这些算法均使用了局部一致性假设,即空间上邻近的点可能具有相似的嵌入。基于这一假设,CAI 等<sup>[22]</sup>通过构造最近邻图来反映数据空间的几何信息,提出图正则化非负矩阵分解(Graph regularized Non-negative Matrix Factorization, GNMF)方法用于数据表示。考虑到数据中的低秩

成分,LI 等<sup>[23]</sup>提出图正则化非负低秩矩阵分解(Graph regularized Non-negative Low-rank Matrix Factorization, GNLMF)方法。张量作为矩阵的高阶扩展,也受到了研究人员的关注。QIU 等<sup>[24]</sup>通过对最后一个非负因子矩阵施加图约束,提出一种图正则化非负 Tucker 分解(Graph regularized Non-negative Tucker Decomposition, GNTD)方法。SOFUOGLU 等<sup>[25]</sup>引入斯蒂弗尔流形,提出一种图正则化张量链(Graph regularized Tensor Train, GTT)分解方法。吴泽鑫<sup>[26]</sup>提出图正则化非负张量链(Graph regularized Non-negative Tensor Train, GNNT)分解方法。这些方法通过在原始数据空间中构造初始图来提高聚类性能。

然而,传统图结构只能描述数据间的成对关系,在处理具有高阶关系的数据时可能会造成部分有用信息的丢失<sup>[27-28]</sup>。为了解决该问题,SCHÖLKOPF 等<sup>[29]</sup>引入超图模型,它可以连接任意数量的顶点,描述数据间的复杂多元关系。由于这一优势,因此基于超图的学习方法受到了广泛关注。ZENG 等<sup>[30]</sup>将超图正则化项引入标准 NMF 框架,提出一种超图正则化非负矩阵分解(Hyper-graph regularized Non-negative Matrix Factorization, HNMF)方法。陈璐瑶<sup>[31]</sup>将超图约束纳入非负 Tucker 分解,提出一种超图正则化非负 Tucker 分解(Hyper-Graph regularized Non-negative Tucker Decomposition, HGNTD)方法。

受超图学习的启发,结合 NTT 模型在数据表示方面的优势,本文提出超图正则化非负张量链(Hyper-Graph regularized Non-negative Tensor Train, HGNTT)分解方法,在继承非负张量链优秀数据表示能力的同时通过超图刻画数据间的高阶关系。采用  $K$ -最近邻构建超图对原始数据空间的几何结构信息进行编码,然后将其引入非负张量链分解模型,同时利用乘法更新(Multiplicative Update, MU)方法迭代求解得到数据的低维表示。

## 1 相关工作

### 1.1 符号和基本操作

简要介绍张量分解的一些重要概念,同时在表 1 中列出所用到的一些基本符号,更多细节设置可参考文献[2]。

表 1 基本符号表示

Table 1 Basic symbol representation			
符号	含义	符号	含义
$\langle \cdot, \cdot \rangle$	内积	$\otimes$	哈达玛积
$\circ$	外积	$\otimes$	克罗内克积
$\text{tr}\{\cdot\}$	迹运算	*	缩并操作

**定义 1(模式- $n$ 展开)** 张量按模式- $n$ 展开后是 1 个矩阵,对 1 个给定的  $N$  阶张量  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ ,经过

模式- $n$ 展开后得到展开矩阵  $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 \times I_2 \times \dots \times I_{n-1} \times I_{n+1} \times \dots \times I_N}$ 。原始张量  $\mathcal{X}$  中的元素  $x_{i_1, i_2, \dots, i_N}$  映射为模式- $n$  矩阵  $\mathbf{X}_{(n)}$  中的元素  $x_{i_1, i_2, \dots, i_N}^{(I_n \times I_1 \times \dots \times I_{n-1} \times I_{n+1} \times \dots \times I_N)}$ 。

**定义 2 (模式- $n$  乘积)** 1 个  $N$  阶张量  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  和矩阵  $\mathbf{A} \in \mathbb{R}^{J_n \times I_n}$  的模式- $n$  乘积为  $\mathcal{Z} = \mathcal{X} \times_n \mathbf{A} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_{n-1} \times J_n \times I_{n+1} \times \dots \times I_N}$ , 其元素形式表示为 
$$z_{i_1, i_2, \dots, i_{n-1}, i_{n+1}, \dots, i_N} = \sum_{j_n=1}^{I_n} x_{i_1, i_2, \dots, i_n, i_{n+1}, \dots, i_N} a_{j_n, i_n}。$$

**定义 3 (张量的缩并)** 对于 1 个  $K$  阶张量  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N \times \dots \times I_K}$  和 1 个  $D$  阶的张量  $\mathcal{Y} \in \mathbb{R}^{J_1 \times J_2 \times \dots \times J_M \times \dots \times J_D}$ , 其中  $I_N = J_M$ , 这 2 个张量经过缩并积操作后可以整理为一个新的张量  $\mathcal{Z} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_{N-1} \times I_{N+1} \times \dots \times I_K \times J_1 \times J_2 \times \dots \times J_{M-1} \times J_{M+1} \times \dots \times J_D}$ , 可记作  $\mathcal{Z} = \mathcal{X} * \mathcal{Y}$ 。

### 1.2 非负张量链分解

LEE 等<sup>[13]</sup> 提出 NTT 分解方法, 主要思想是将复杂的高维数据优化问题转化为一系列易于处理的低维数据优化问题。因此, NTT 分解旨在用一系列二阶或三阶核心张量的多重线性乘积来表示高维张量。对于一个给定  $N$  阶张量  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ , NTT 分

解旨在搜索原始张量  $\mathcal{X}$  的  $N$  个潜在的二阶或三阶核心张量  $\mathcal{G} = \mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \dots, \mathcal{G}^{(N)}$ , 可以转化为如下最优化问题:

$$\begin{aligned} \min_{\mathcal{G}^{(n)}} & \|\mathcal{X} - \mathcal{G}^{(1)} * \mathcal{G}^{(2)} * \dots * \mathcal{G}^{(N)}\|_F^2 \\ \text{s.t. } & \mathcal{G}^{(n)} \geq 0, n = 1, 2, \dots, N \end{aligned} \quad (1)$$

其中:  $\mathcal{G}^{(n)} \in \mathbb{R}^{R_{n-1} \times I_n \times R_n}$ 。

在这种情况下, 核心张量  $\mathcal{G}$  与  $\mathcal{X}$  的每个元素的关系可以表示如下<sup>[10]</sup>:

$$\mathcal{X}(i_1, i_2, \dots, i_N) = \sum_{r_0, r_1, \dots, r_N} \mathcal{G}^{(1)}(r_0, i_1, r_1) \mathcal{G}^{(2)}(r_1, i_2, r_2) \dots \mathcal{G}^{(N)}(r_{N-1}, i_N, r_N) \quad (2)$$

其中:  $n = 1, 2, \dots, N$ ;  $\mathbf{R} = [R_1, R_2, \dots, R_N]^T \in \mathbb{R}^{N \times 1}$  是 NTT 的秩, 但值得注意的是, NTT 模型秩中限制  $R_0 = R_N = 1$ 。将其扩展到矩阵层面, 可得到:

$$\mathcal{X}(i_1, i_2, \dots, i_N) = \mathbf{G}_{i_1}^{(1)} \times \mathbf{G}_{i_2}^{(2)} \times \dots \times \mathbf{G}_{i_N}^{(N)} \quad (3)$$

其中:  $\mathbf{G}_{i_n}^{(N)} \in \mathbb{R}^{R_{n-1} \times R_n}$  是第  $N$  个核心张量的第  $N$  个侧面切片矩阵。

NTT 分解示意图如图 1 所示。

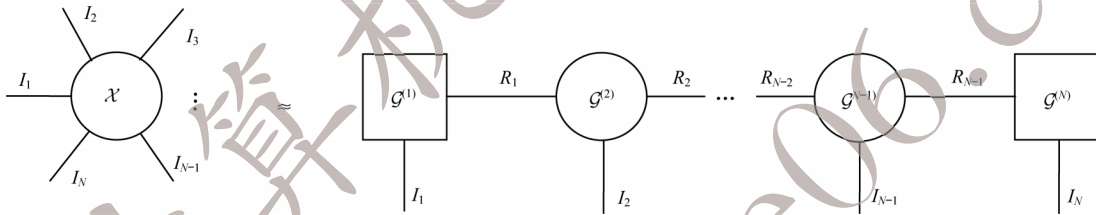


图 1 NTT 分解示意图

Fig.1 Schematic diagram of NTT decomposition

如果对 1 个张量  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$  进行 NTT 分解, 可以用 2 条由对应的核心张量缩并操作得到的子链  $\mathcal{G}^{<n}$  和  $\mathcal{G}^{>n}$  来简化表示:

$$\begin{aligned} \mathcal{G}^{<n} &= \mathcal{G}^{(1)} * \mathcal{G}^{(2)} * \dots * \mathcal{G}^{(n-1)} \in \mathbb{R}^{I_1 \times \dots \times I_{n-1} \times R_{n-1}} \\ \mathcal{G}^{>n} &= \mathcal{G}^{(n+1)} * \mathcal{G}^{(n+2)} * \dots * \mathcal{G}^{(N)} \in \mathbb{R}^{R_n \times I_{n+1} \times \dots \times I_N} \end{aligned} \quad (4)$$

推导得到原始张量  $\mathcal{X}$  的模式- $n$  矩阵  $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 \times \dots \times I_{n-1} \times I_{n+1} \times \dots \times I_N}$ , 可表示如下:

$$\mathbf{X}_{(n)} = \mathbf{G}_{(2)}^{(n)} (\mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n}) \quad (5)$$

其中:  $\mathbf{G}_{(2)}^{(n)}$  是核心张量  $\mathcal{G}^{(n)}$  按模式-2 展开得到的展开矩阵;  $\mathbf{G}_{(1)}^{<n}$  和  $\mathbf{G}_{(n)}^{>n}$  分别是对子链  $\mathcal{G}^{<n}$  和  $\mathcal{G}^{>n}$  按模式- $n$  和模式-1 展开得到的展开矩阵。

结合式 (5), NTT 的子问题形式  $\mathcal{O}_{\text{NTT}}^{(n)}$  可以表示如下:

$$\begin{aligned} \mathcal{O}_{\text{NTT}}^{(n)} &= \frac{1}{2} \|\mathbf{X}_{(n)} - \mathbf{G}_{(2)}^{(n)} (\mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n})\|_F^2 \\ \text{s.t. } & \mathbf{G}_{(2)}^{(n)} \geq 0, \mathbf{G}_{(1)}^{>n} \geq 0, \mathbf{G}_{(n)}^{<n} \geq 0, n = 1, 2, \dots, N \end{aligned} \quad (6)$$

在求解每个子问题时, 将得到的变量  $\mathbf{G}_{(2)}^{(n)}$  视为当前子问题的最优核张量的展开矩阵。

### 1.3 超图学习

超图是简单图的一种扩展形式, 与简单图之间的区别在于每条边所能连接的顶点数, 在普通图中每条边只能连接 2 个顶点, 而超图中的超边可以连

接 2 个以上的顶点, 这使得超图更适合用来描述具有复杂多元关系的数据<sup>[27]</sup>, 如图 2 所示。超图学习目前已广泛应用于分类、聚类、检索、嵌入等任务, 并取得了较好的效果。

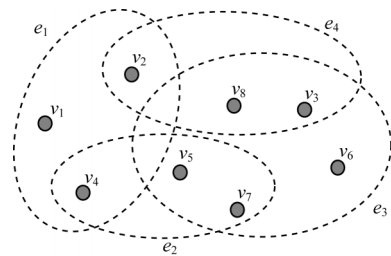


图 2 超图 G 示意图

Fig.2 Schematic diagram of hyper-graph G

假定超图  $G = (V, E, W)$ , 其中:  $V = \{v_1, v_2, \dots, v_m\}$  表示顶点集, 集合  $V$  中的每一个元素称为超图的顶点;  $E = \{e_1, e_2, \dots, e_n\}$  表示超边集, 每条超边  $e$  都是顶点集  $V$  的一个子集;  $W \in \mathbb{R}^{n \times n}$  是超图的权重矩阵, 是由每条超边对应的权重  $w(e)$  构成的一个对角矩阵。超图可以由一个  $|V| \times |E|$  的关联矩阵  $\mathbf{H} \in \mathbb{R}^{m \times n}$ , 如表 2 所示, 可表示如下:

$$\mathbf{H}(v, e) = \begin{cases} 1, v \in e \\ 0, v \notin e \end{cases} \quad (7)$$

其中: 顶点  $v \in V$ ; 超边  $e \in E$ 。

表2 关联矩阵  $\mathbf{H}$

Table 2 Correlation matrix  $\mathbf{H}$

顶点	$e_1$	$e_2$	$e_3$	$e_4$
$v_1$	1	0	0	0
$v_2$	1	0	0	1
$v_3$	0	0	1	1
$v_4$	1	1	0	0
$v_5$	0	1	1	0
$v_6$	0	0	1	0
$v_7$	0	1	1	0
$v_8$	0	0	1	1

超边  $e$  的度表示其所包含的顶点个数, 记作  $\delta(e)$ , 可表示如下:

$$\delta(e) = \sum_{v \in V} \mathbf{H}(v, e) \quad (8)$$

顶点  $v$  的度表示  $v$  所属的超边的权值累加和, 记作  $d(v)$ , 可表示如下:

$$d(v) = \sum_{e \in E} w(e) \mathbf{H}(v, e) \quad (9)$$

超边的权值  $w(e)$  可表示如下:

$$w(e) = \frac{1}{\delta(e)(\delta(e)-1)} \sum_{u, v \in e} \exp\left(-\frac{\|u-v\|^2}{\sigma^2}\right) \quad (10)$$

设  $\mathbf{D}_e \in \mathbb{R}^{|E| \times |E|}$  和  $\mathbf{D}_v \in \mathbb{R}^{|V| \times |V|}$  分别是超边和顶点的度所构建的对角矩阵。对于一个简单图, 拉普拉斯矩阵定义为  $\mathbf{L} = \mathbf{D} - \mathbf{A}$ , 其中,  $\mathbf{D}$  是顶点度的对角矩阵,  $\mathbf{A}$  是邻接矩阵。对于超图, 拉普拉斯矩阵更为复杂, 可表示如下:

$$\mathbf{L}_{\text{hyper}} = \mathbf{D}_v - \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}^T \quad (11)$$

由此可知, 超边可以拥有任意数量的顶点, 但考虑到实际应用, 假定每条超边具有同样数量的顶点。

## 2 超图正则化非负张量链分解

介绍 HGNTT 模型, 同时给出超图的构造方法, 设计基于乘法更新方法的 HGNTT 算法 (简称为 HGNTT-MU 算法) 对该模型进行优化求解。

### 2.1 超图构造

为了利用超图表示数据的相关性, 首先选择合适的方法从原始数据中构造出一个较为理想的超图。与普通图类似, 超图的构造主要包括超边的选取和超边权重的设计两部分。采用基于样本特征的超边构造方法, 针对所采用的图像数据集  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ , 将每幅图像  $\mathbf{x}_i$  看作一个质心顶点  $v_i$ , 然后在特征空间中寻找与当前质心顶点距离最近的

$K$  个顶点, 并将选定的这  $K$  个最邻近顶点构成一条超边  $e_i \in E$ 。此外, 超边权重的不同<sup>[28]</sup> 也会令超图模型表示的信息有许多差异, 进而对算法的结果产生巨大的影响。因此, 选择一个合适的超边权重设计方法是十分有必要的。

目前, 常见的超边权重设计方法主要包括 0-1 加权、热核加权和点积加权 3 种方法。给定 1 个数据点, 在 0-1 加权方法中  $K$  个最近邻点具有相同的重要性, 可表示如下:

$$\mathbf{W}_{ij} = \begin{cases} 1, (i, j) \in E \\ 0, \text{其他} \end{cases} \quad (12)$$

其中:  $\mathbf{W} \in \mathbb{R}^{m \times m}$  为权重矩阵, 代入式 (10) 中计算超边  $e_i$  对应的权重  $w(e_i)$ 。热核和点积加权方法对其  $K$  个最近邻点的处理有所不同, 可分别表示如下:

$$\mathbf{W}_{ij} = \begin{cases} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right), (i, j) \in E \\ 0, \text{其他} \end{cases} \quad (13)$$

$$\mathbf{W}_{ij} = \begin{cases} \mathbf{x}_i^T \mathbf{x}_j, (i, j) \in E \\ 0, \text{其他} \end{cases} \quad (14)$$

其中:  $\sigma$  是一个平滑参数, 表示全部顶点间的平均距离。

基于流形学习思想, 如果 2 个特征向量在原始空间中的几何分布相似, 那么这 2 个特征向量相对于新的基空间的编码向量也应该相似。因此, 采用超图正则化项的代价函数来度量低维编码数据的光滑性<sup>[27]</sup>, 可表示如下:

$$\begin{aligned} \Omega &= \frac{1}{2} \sum_{e \in E} \sum_{u, v \in V} \frac{w(e) \mathbf{H}(u, e) \mathbf{H}(v, e)}{\delta(e)} \|\mathbf{f}(u) - \mathbf{f}(v)\|_F^2 = \\ & \sum_{e \in E} \sum_{u \in V} w(e) \|\mathbf{f}(u)\|_F^2 \mathbf{H}(u, e) \sum_{v \in V} \frac{\mathbf{H}(v, e)}{\delta(e)} - \\ & \sum_{e \in E} \sum_{u, v \in V} \frac{w(e) \mathbf{H}(u, e) \mathbf{H}(v, e)}{\delta(e)} \mathbf{f}^T(u) \mathbf{f}(v) = \\ & \sum_{e \in E} w(e) \mathbf{H}(u, e) \|\mathbf{f}(u)\|_F^2 - \\ & \sum_{u, v \in V} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}^T \mathbf{f}^T(u) \mathbf{f}(v) = \\ & \text{tr}(\mathbf{F} \mathbf{D}_v \mathbf{F}^T) - \text{tr}(\mathbf{F} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}^T \mathbf{F}^T) = \\ & \text{tr}(\mathbf{F}^T \mathbf{L}_{\text{hyper}} \mathbf{F}) \end{aligned} \quad (15)$$

其中:  $\text{tr}(\cdot)$  是迹运算;  $\mathbf{F} = [\mathbf{f}(u_1), \mathbf{f}(u_2), \dots, \mathbf{f}(u_m)]$  是低维编码数据矩阵;  $\mathbf{f}(u)$  是待学习样本标签的低维表示。

### 2.2 目标函数

将超图正则化项添加到原始的 NTT 模型中, 提出 HGNTT 模型, 目标是用超图所刻画的样本间的高阶关系对核心张量进行约束, 提升模型的整体性能。

与此同时, 考虑到张量数据的最后一个维度为样本维, 默认表示样本数量, 如果将张量  $\mathcal{G}^{(1)} * \mathcal{G}^{(2)} * \dots * \mathcal{G}^{(N)}$  的模式- $N$  展开看作聚类中心点, 那么最后一个核心张量  $\mathcal{G}^{(N)}$  则可以被认为是特征核心张量, 作为基于局部特征和超图拉普拉斯学习的聚类指示子张量。基于上述分析, 将超图约束添加到最后一个核心张量  $\mathcal{G}^{(N)}$  上, 利用样本间的高阶关系, 进一步增强提取到可靠的低维特征, 因此目标函数可表示如下:

$$\begin{aligned} \mathcal{O}_{\text{HGNTT}}^{(n)} = & \frac{1}{2} \left\| \mathbf{X}_{(n)} - \mathbf{G}_{(2)}^{(n)} \left( \mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n} \right) \right\|_F^2 + \\ & \frac{\lambda}{2} \text{tr} \left( \left( \mathbf{G}_{(2)}^{(n)} \right)^T \mathbf{L}_{\text{hyper}} \mathbf{G}_{(2)}^{(n)} \right) \\ \text{s.t. } & \mathbf{G}_{(2)}^{(n)} \geq 0, \mathbf{G}_{(1)}^{>n} \geq 0, \mathbf{G}_{(n)}^{<n} \geq 0, n = 1, 2, \dots, N \end{aligned} \quad (16)$$

其中: 系数  $\lambda$  为惩罚因子, 用来控制目标函数中超图正则化项的强度。使用超图拉普拉斯矩阵  $\mathbf{L}_{\text{hyper}} = \mathbf{D}_v - \mathbf{HWD}_e^{-1} \mathbf{H}^T$  将数据中的相似结构嵌入第  $N$  个核心张量的模式-2 展开矩阵  $\mathbf{G}_{(2)}^{(n)}$  中。

### 2.3 优化方法

为了优化求解式 (16) 中的目标函数, 采用乘法更新方法<sup>[22]</sup>, 其在保证优化变量为非负值的同时通过梯度下降优化目标函数。因为超图约束是添加到最后一个核心张量  $\mathcal{G}^{(N)}$  上, 只有在  $n=N$  时模型才会学习数据的几何结构信息, 所以分  $n \neq N$  和  $n=N$  这 2 种情况对目标函数进行优化。

应用拉格朗日乘子法, 将  $\Phi_{(n)}$  定义为约束  $\mathbf{G}_{(2)}^{(n)} \geq 0$  的拉格朗日乘子矩阵, 然后基于式 (16) 的拉格朗日函数可表示如下:

$$\begin{aligned} \mathcal{L}_{\text{HGNTT}}^{(n)} = & \frac{1}{2} \left\| \mathbf{X}_{(n)} - \mathbf{G}_{(2)}^{(n)} \left( \mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n} \right) \right\|_F^2 + \\ & \frac{\lambda}{2} \text{tr} \left( \left( \mathbf{G}_{(2)}^{(n)} \right)^T \mathbf{L}_{\text{hyper}} \mathbf{G}_{(2)}^{(n)} \right) + \text{tr} \left( \Phi_{(n)} \left( \mathbf{G}_{(2)}^{(n)} \right)^T \right) = \\ & \frac{1}{2} \text{tr} \left( \mathbf{X}_{(n)} \mathbf{X}_{(n)}^T \right) - \text{tr} \left( \mathbf{X}_{(n)} \left( \mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n} \right)^T \left( \mathbf{G}_{(2)}^{(n)} \right)^T \right) + \\ & \frac{1}{2} \text{tr} \left( \mathbf{G}_{(2)}^{(n)} \left( \mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n} \right) \left( \mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n} \right)^T \left( \mathbf{G}_{(2)}^{(n)} \right)^T \right) + \\ & \frac{\lambda}{2} \text{tr} \left( \left( \mathbf{G}_{(2)}^{(n)} \right)^T \mathbf{L}_{\text{hyper}} \mathbf{G}_{(2)}^{(n)} \right) + \text{tr} \left( \Phi_{(n)} \left( \mathbf{G}_{(2)}^{(n)} \right)^T \right) \end{aligned} \quad (17)$$

#### 2.3.1 核心张量 $\mathbf{G}_{(2)}^{(n)} (n \neq N)$ 优化

$\mathcal{L}_{\text{HGNTT}}^{(n)}$  关于优化变量  $\mathbf{G}_{(2)}^{(n)}$  的求导结果如式 (18) 所示。考虑到 Karush-Kuhn-Tucker (KKT) 条件, 令  $\frac{\partial \mathcal{L}_{\text{HGNTT}}^{(n)}}{\partial \mathbf{G}_{(2)}^{(n)}} = 0$  且  $(\Phi_{(n)})_{ij} (\mathbf{G}_{(2)}^{(n)})_{ij} = 0$ , 得到式 (19)。根据梯度下降方法, 得到式 (20)。若选取步长  $\alpha_n$  [如式 (21) 所示], 则  $\mathbf{G}_{(2)}^{(n)}$  可按照如式 (22) 所示的更新规则进行优化。

$$\frac{\partial \mathcal{L}_{\text{HGNTT}}^{(n)}}{\partial \mathbf{G}_{(2)}^{(n)}} = \mathbf{G}_{(2)}^{(n)} \left( \mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n} \right) \left( \mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n} \right)^T - \mathbf{X}_{(n)} \left( \mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n} \right)^T + \Phi_{(n)} \quad (18)$$

$$\left[ \mathbf{G}_{(2)}^{(n)} \left( \mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n} \right) \left( \mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n} \right)^T - \mathbf{X}_{(n)} \left( \mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n} \right)^T \right]_{ij} \left( \mathbf{G}_{(2)}^{(n)} \right)_{ij} = 0 \quad (19)$$

$$\left( \mathbf{G}_{(2)}^{(n)} \right)_{ij} \leftarrow \left( \mathbf{G}_{(2)}^{(n)} \right)_{ij} - \alpha_n \left[ \mathbf{G}_{(2)}^{(n)} \left( \mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n} \right) \left( \mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n} \right)^T - \mathbf{X}_{(n)} \left( \mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n} \right)^T \right]_{ij} \quad (20)$$

$$\alpha_n = \frac{\left[ \mathbf{G}_{(2)}^{(n)} \right]_{ij}}{\left[ \mathbf{G}_{(2)}^{(n)} \left( \mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n} \right) \left( \mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n} \right)^T \right]_{ij}} \quad (21)$$

$$\left( \mathbf{G}_{(2)}^{(n)} \right)_{ij} \leftarrow \left( \mathbf{G}_{(2)}^{(n)} \right)_{ij} \frac{\left[ \mathbf{X}_{(n)} \left( \mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n} \right)^T \right]_{ij}}{\left[ \mathbf{G}_{(2)}^{(n)} \left( \mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n} \right) \left( \mathbf{G}_{(1)}^{>n} \otimes \mathbf{G}_{(n)}^{<n} \right)^T \right]_{ij}} \quad (22)$$

#### 2.3.2 核心张量 $\mathbf{G}_{(2)}^{(N)} (n=N)$ 优化

当  $n=N$  时, 在最后一个核心张量  $\mathcal{G}^{(N)}$  上施加了超图约束, 因此  $\mathcal{L}_{\text{HGNTT}}^{(N)}$  关于优化变量  $\mathbf{G}_{(2)}^{(N)}$  的求导结果如式 (23) 所示。同样地, 考虑到 KKT 条件, 令  $\frac{\partial \mathcal{L}_{\text{HGNTT}}^{(N)}}{\partial \mathbf{G}_{(2)}^{(N)}} = 0$  且  $(\Phi_{(N)})_{ij} (\mathbf{G}_{(2)}^{(N)})_{ij} = 0$ , 得到式 (24)。在式 (23) 中的  $\mathbf{L}_{\text{hyper}}$  用  $\mathbf{D}_v - \mathbf{HWD}_e^{-1} \mathbf{H}^T$  进行替换得到  $\mathbf{S} = \mathbf{HWD}_e^{-1} \mathbf{H}^T$ , 根据梯度下降方法得到式 (25)。若取步长  $\alpha_N$  [如式 (26) 所示],  $\mathbf{G}_{(2)}^{(N)}$  可按照如式 (27) 所示的更新规则进行优化。

$$\frac{\partial \mathcal{L}_{\text{HGNTT}}^{(N)}}{\partial \mathbf{G}_{(2)}^{(N)}} = \mathbf{G}_{(2)}^{(N)} \left( \mathbf{G}_{(1)}^{>N} \otimes \mathbf{G}_{(N)}^{<N} \right) \left( \mathbf{G}_{(1)}^{>N} \otimes \mathbf{G}_{(N)}^{<N} \right)^T - \mathbf{X}_{(N)} \left( \mathbf{G}_{(1)}^{>N} \otimes \mathbf{G}_{(N)}^{<N} \right)^T + \lambda \mathbf{L}_{\text{hyper}} \mathbf{G}_{(2)}^{(N)} + \Phi_{(N)} \quad (23)$$

$$\left[ \mathbf{G}_{(2)}^{(N)} \left( \mathbf{G}_{(1)}^{>N} \otimes \mathbf{G}_{(N)}^{<N} \right) \left( \mathbf{G}_{(1)}^{>N} \otimes \mathbf{G}_{(N)}^{<N} \right)^T - \mathbf{X}_{(N)} \left( \mathbf{G}_{(1)}^{>N} \otimes \mathbf{G}_{(N)}^{<N} \right)^T + \lambda \mathbf{D}_v \mathbf{G}_{(2)}^{(N)} - \lambda \mathbf{S} \mathbf{G}_{(2)}^{(N)} \right]_{ij} \left( \mathbf{G}_{(2)}^{(N)} \right)_{ij} = 0 \quad (24)$$

$$\left( \mathbf{G}_{(2)}^{(N)} \right)_{ij} \leftarrow \left( \mathbf{G}_{(2)}^{(N)} \right)_{ij} - \alpha_N \left[ \mathbf{G}_{(2)}^{(N)} \left( \mathbf{G}_{(1)}^{>N} \otimes \mathbf{G}_{(N)}^{<N} \right) \left( \mathbf{G}_{(1)}^{>N} \otimes \mathbf{G}_{(N)}^{<N} \right)^T + \lambda \mathbf{D}_v \mathbf{G}_{(2)}^{(N)} - \mathbf{X}_{(N)} \left( \mathbf{G}_{(1)}^{>N} \otimes \mathbf{G}_{(N)}^{<N} \right)^T - \lambda \mathbf{S} \mathbf{G}_{(2)}^{(N)} \right]_{ij} \quad (25)$$

$$\alpha_N = \frac{\left[ \mathbf{G}_{(2)}^{(N)} \right]_{ij}}{\left[ \mathbf{G}_{(2)}^{(N)} \left( \mathbf{G}_{(1)}^{>N} \otimes \mathbf{G}_{(N)}^{<N} \right) \left( \mathbf{G}_{(1)}^{>N} \otimes \mathbf{G}_{(N)}^{<N} \right)^T + \lambda \mathbf{D}_v \mathbf{G}_{(2)}^{(N)} \right]_{ij}} \quad (26)$$

$$\left( \mathbf{G}_{(2)}^{(N)} \right)_{ij} \leftarrow \left( \mathbf{G}_{(2)}^{(N)} \right)_{ij} \cdot \frac{\left[ \mathbf{X}_{(N)} \left( \mathbf{G}_{(0)}^{>N} \otimes \mathbf{G}_{(N)}^{<N} \right)^T + \lambda \mathbf{S} \mathbf{G}_{(2)}^{(N)} \right]_{ij}}{\left[ \mathbf{G}_{(2)}^{(N)} \left( \mathbf{G}_{(0)}^{>N} \otimes \mathbf{G}_{(N)}^{<N} \right) \left( \mathbf{G}_{(0)}^{>N} \otimes \mathbf{G}_{(N)}^{<N} \right)^T + \lambda \mathbf{D}_v \mathbf{G}_{(2)}^{(N)} \right]_{ij}} \quad (27)$$

综合以上分析结果, HGNTT-MU 算法的伪代码具体如下:

#### 算法 1 HGNTT-MU 算法

输入 非负张量  $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ 、最大迭代次数  $t_{\max}$ 、秩  $\mathbf{R} = [R_1, R_2, \dots, R_N]^T$ 、惩罚因子  $\lambda$

输出 核心张量  $\mathcal{G}^{(n)}, n = 1, 2, \dots, N$

1. 初始化  $\mathcal{G}^{(n)}$ , 服从  $[0, 1]$  均匀分布;
2. 计算超图拉普拉斯矩阵  $\mathbf{L}_{\text{hyper}} = \mathbf{D}_v - \mathbf{S}$ ;
3. 重复:
4. for  $n = 1$  to  $N$  do
5. 计算子链的模式-2 展开矩阵  $\mathbf{G}_{(0)}^{(>n)} \otimes \mathbf{G}_{(n)}^{(<n)}$ ;
6. for  $t = 1$  to  $t_{\max}$  do
7. if  $n = N$  then
8. 通过式 (27), 更新  $\mathbf{G}_{(2)}^{(N)}$ ;
9. else
10. 通过式 (22), 更新  $\mathbf{G}_{(2)}^{(n)}$ ;
11. end if
12. end for
13. 通过  $\mathcal{G}^{(n)} \leftarrow \text{folding}(\mathbf{G}_{(2)}^{(n)})$ , 对  $\mathbf{G}_{(2)}^{(n)}$  张量化;
14. end for
15. 直到: 达到收敛条件

### 3 实验结果与分析

通过聚类实验来分析并验证 HGNTT 方法在不同数据集上的聚类性能, 并将其与通用算法进行对比, 验证 HGNTT 算法的先进性和优越性。

#### 3.1 评价指标

在进行聚类实验时, 将从不同模型中所得到的聚类结果与真实标签进行对比, 并使用合适的指标来评判实验结果, 进而验证各个模型在聚类性能上的优劣性。采用 2 个通用的指标来评估聚类性能: 聚类准确率 ( $A$ )<sup>[32]</sup> 和归一化互信息 (Normalized Mutual Information, NMI)。

聚类准确率可表示如下:

$$A(r, \hat{r}) = \frac{1}{n} \sum_{i=1}^n \delta(r_i, \text{map}(\hat{r}_i)) \quad (28)$$

其中:  $r$  为真实标签集合;  $\hat{r}$  为预测标签集合;  $n$  为样本总数;  $\text{map}(\cdot)$  为排列映射函数, 将真实标签当作参考标签, 然后依照同样的排列方式对预测标签进行重排, 聚类标签与真实标签之间是 1-1 映射关系。因此,  $\text{map}(\cdot)$  常被用于处理标签不统一问题, 结合

Kuhn-Munkres 算法通常可得到一个较理想的结果。

$\delta(x, y)$  函数可表示如下:

$$\delta(x, y) = \begin{cases} 1, & x = y \\ 0, & \text{其他} \end{cases} \quad (29)$$

NMI 是聚类分析中的一个重要度量指标, 用来表示 2 个聚类数据集之间的相似程度<sup>[31]</sup>。MI 表示互信息, 反映 2 个数据集  $r$  和  $\hat{r}$  之间信息量的相关性, 可表示如下:

$$\text{MI}(r, \hat{r}) = \sum_{r_i \in r, \hat{r}_i \in \hat{r}} p(r_i, \hat{r}_i) \cdot \text{lb} \frac{p(r_i, \hat{r}_i)}{p(r_i) \cdot p(\hat{r}_i)} \quad (30)$$

其中:  $p(r_i)$  和  $p(\hat{r}_i)$  分别表示在数据集中任意选取的样本分别属于集合  $r_i$  和  $\hat{r}_i$  的概率;  $p(r_i, \hat{r}_i)$  表示任意选取的样本同时属于集合  $r_i$  和  $\hat{r}_i$  的联合概率; MI 值越大, 2 个数据集之间的相似程度越高。

为了简化分析, 使用 NMI 来代替 MI, 可表示如下:

$$\text{NMI}(r, \hat{r}) = \frac{\text{MI}(r, \hat{r})}{\sqrt{H(r) \cdot H(\hat{r})}} \quad (31)$$

其中:  $H(r)$  和  $H(\hat{r})$  分别表示集合  $r$  和  $\hat{r}$  的熵值。归一化互信息  $\text{NMI}(r, \hat{r})$  的取值区间为 0~1。当 2 个集合  $r$  和  $\hat{r}$  完全相同时  $\text{NMI}(r, \hat{r})$  取值为 1, 当 2 个集合  $r$  和  $\hat{r}$  相互独立时  $\text{NMI}(r, \hat{r})$  取值为 0。

#### 3.2 实验数据与参数设置

为了验证 HGNTT 方法的有效性, 将其与 NMF<sup>[9]</sup>、NTT<sup>[10]</sup>、GNMF<sup>[18]</sup>、GNTT<sup>[26]</sup>、HNMF<sup>[30]</sup>、HGNTD<sup>[31]</sup>、 $k$ -means<sup>[33]</sup> 等 7 种主流方法进行对比。将  $k$ -means 作为基准方法, 直接对样本数据在原始空间上进行聚类。对于 NMF、NTT、GNMF、GNTT、HNMF、HGNTD、HGNTT 等方法, 样本数据可以从中学到紧凑的低维表征, 然后对这些经过处理后的低维表征使用  $k$ -means 进行聚类。在数据集选取上, 在 2 个常用公开数据集 ORL 和 Faces95 上进行聚类实验:

1) ORL 数据集。ORL 人脸数据集由 40 个不同采集者的 400 张灰度图像构成。每个被采集者分别有 10 张不同的图像, 每张图像的尺寸为  $119 \times 92$  像素。对于每个图像采集者, 分别在不同的时间和光照情况下, 拍摄不同的面部表情和面部细节。在实验中, 所有的图像尺寸都被调整为  $32 \times 27$  像素, 构成一个维度是  $\mathbb{R}^{32 \times 27 \times 400}$  (其中, 400 表示图像张数) 的 3 阶张量数据集。

2) Faces95 数据集。Faces95 人脸数据集由 72 个不同采集者的 1 440 张彩色图像构成。每个被采集

者分别有 20 张不同的图像, 每张图像的尺寸为  $180 \times 200$  像素。对于每个图像采集者, 拍摄在向前移动过程中的一系列面部表情。由于光线照射和身体摆动的影响, 同一个采集者在不同图像上的面部表情都会有一定程度的变化。在实验中, 所有的图像尺寸都被调整为  $50 \times 45$  像素, 构成一个维度是  $\mathbb{R}^{50 \times 45 \times 3 \times 1440}$  (其中, 3 表示三原色取值, 1440 表示图像张数) 的 4 阶张量数据集。

在实验中, 为了降低随机初始化对实验结果的影响, 重复进行多次实验, 然后计算平均值作为最终结果。为了确保实验结果的公平性, 对实验中所用到的公共参数进行统一设置。对于所有方法, 将迭代终止误差统一设置为  $10^{-4}$ 。对于 GNMF、HNMF、HGNTD、GNTT、HGNTT 方法, 考虑到参数  $\lambda$  对图和超图约束的影响, 采用网格搜索法从  $\{10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1\}$  中遍历参数进行对比实验, 选取性能最优的一组数据作为实验结果。

在秩的选取上, 对于 NMF、GNMF 和 HNMF, 将基矩阵的维数设置为样本类别数  $K$ 。对于 HGNTD, 将最后 1 维的秩固定为样本类别数  $K$ , 对于其他  $N-1$  个秩, 令其相等并在  $\left(\frac{I_n}{4}, \frac{I_n}{2}\right)$  中进行遍历搜索, 其中,  $I_n$  是除样本维度外其余维度数据的中值。对于 NTT、GNTT 和 HGNTT, 将第 1 维的秩固定为 1, 将最后 1 维的秩设置为选取的数据集样本类别数  $K$ 。同样地, 对于其他  $N-2$  个秩, 令其相等, 并在  $\left(\frac{I_n}{4}, \frac{I_n}{2}\right)$  中进行遍历搜索, 最后选出各方法性能最优的一组数据作为实验结果。

### 3.3 结果分析

为了使实验结果更具可信性, 对选取的数据集进行分类处理。根据包含的样本类别数  $K$  的不同, 将原本的数据集分割为多个不同的子集, 对这些子集依次进行聚类实验, 然后综合对比在简单样本和复杂样本条件下 HGNTT 方法的适用性。对于 ORL 数据集, 取前  $K$  ( $K$  为 10、15、20、25、30、35、40) 类样本构成 7 个新子集; 对于 Faces95 数据集, 取前  $K$  ( $K$  为 10、20、30、40、50、60) 类样本构成 6 个新子集。表 3 和表 4 给出了各方法在 ORL 数据集上进行聚类实验的准确率和归一化互信息, 表 5 和表 6 给出了各方法在 Faces95 数据集上进行聚类实验的准确率和归一化互信息, 其中最优指标值用加粗字体标示。

表 3 在 ORL 数据集的不同子集上的聚类准确率

Table 3 Clustering accuracy on different subsets

of the ORL dataset									%
$K$	$k$ -means	NMF	GNMF	HNMF	HGNTD	NTT	GNTT	HGNTT	
10	95.40	91.50	92.00	95.67	95.00	92.50	94.50	<b>96.00</b>	
15	84.00	83.67	86.67	89.56	89.33	85.33	89.67	<b>91.56</b>	
20	77.25	80.00	81.00	82.50	86.17	79.25	86.25	<b>87.15</b>	
25	73.40	78.20	80.80	80.40	83.60	78.60	82.40	<b>85.80</b>	
30	71.83	74.00	77.50	78.67	79.78	75.17	79.67	<b>80.17</b>	
35	68.43	70.78	73.86	73.52	75.91	72.86	76.34	<b>76.72</b>	
40	67.75	68.25	72.25	70.25	73.92	69.88	<b>74.25</b>	73.88	
Avg	76.86	78.06	80.58	81.51	83.39	79.08	83.30	<b>84.47</b>	

表 4 在 ORL 数据集的不同子集上的归一化互信息

Table 4 Normalized mutual information

on different subsets of the ORL dataset

of the ORL dataset									%
$K$	$k$ -means	NMF	GNMF	HNMF	HGNTD	NTT	GNTT	HGNTT	
10	94.70	92.23	91.04	94.66	93.68	91.37	93.06	<b>95.29</b>	
15	91.03	90.83	91.24	92.79	92.89	89.85	92.86	<b>92.96</b>	
20	87.86	88.50	88.85	89.98	91.30	88.28	91.13	<b>91.64</b>	
25	85.46	88.50	89.13	88.20	90.17	86.70	89.51	<b>90.38</b>	
30	84.44	85.57	88.55	89.36	89.24	84.71	88.45	<b>89.41</b>	
35	85.11	84.92	87.23	86.68	<b>88.16</b>	85.89	88.10	87.96	
40	84.71	83.77	86.13	84.84	<b>87.68</b>	85.04	87.31	86.84	
Avg	87.62	87.76	88.88	89.50	90.44	87.41	90.06	<b>90.64</b>	

表 5 在 Faces95 数据集的不同子集上的聚类准确率

Table 5 Clustering accuracy on different subsets

of the Faces95 dataset

of the Faces95 dataset									%
$K$	$k$ -means	NMF	GNMF	HNMF	HGNTD	NTT	GNTT	HGNTT	
10	61.25	57.75	61.25	61.85	66.50	64.75	68.75	<b>70.00</b>	
20	46.38	49.88	52.00	47.63	53.00	<b>56.25</b>	52.50	54.25	
30	50.67	48.58	53.03	51.50	53.83	52.83	53.42	<b>56.50</b>	
40	49.00	48.50	51.25	47.38	51.33	50.19	51.56	<b>53.44</b>	
50	46.35	46.55	49.30	47.60	50.20	47.85	50.00	<b>50.85</b>	
60	45.70	46.17	46.52	46.42	46.89	46.29	46.67	<b>47.54</b>	
Avg	49.89	48.41	52.23	50.40	53.63	53.03	53.82	<b>55.43</b>	

表 6 在 Faces95 数据集的不同子集上的归一化互信息

Table 6 Normalized mutual information

on different subsets of the Faces95 dataset

of the Faces95 dataset									%
$K$	$k$ -means	NMF	GNMF	HNMF	HGNTD	NTT	GNTT	HGNTT	
10	65.84	61.44	68.27	64.32	69.82	65.54	68.72	<b>69.91</b>	
20	60.75	61.89	64.66	61.13	63.98	<b>67.36</b>	63.98	64.36	
30	67.37	64.68	69.03	66.15	68.57	68.64	68.14	<b>69.22</b>	
40	69.77	67.08	69.97	67.03	69.22	67.50	69.63	<b>70.26</b>	
50	68.51	68.00	69.42	67.76	69.95	68.83	69.45	<b>70.06</b>	
60	69.13	68.71	69.05	68.90	69.14	69.15	68.93	<b>69.52</b>	
Avg	66.90	65.30	68.40	65.89	68.47	67.84	68.14	<b>68.89</b>	

从上述实验结果可以看出: 在保留了数据内部的多线性结构后, 张量分解方法相比于矩阵分解方法普遍具有更好的聚类性能; 在保留了数据的局部

几何结构信息后,施加了图约束和超图约束的GNMF、GNNT、HNMF、HGNTD等方法相比普通方法在聚类性能上有更大的性能提升;HGNTT方法在继承之前方法优良特性的基础上,进一步考虑数据间的复杂多元关系,相比于其他方法,聚类准确率提升了约1.2%~7.6%,归一化互信息提升了约0.2%~3.0%,验证了HGNTT方法在处理聚类任务时的有效性。

为了验证HGNTT-MU算法的收敛性,在ORL和Faces95数据集上进行收敛性实验,最大迭代次数统一设置为400。图3和图4分别给出了HGNTT-MU算法在ORL和Faces95数据集上的收敛曲线,可以看出2条曲线在总体上呈现单调下降趋势,最终趋于平稳,具体为:在迭代初期呈现快速下降趋势,在迭代次数接近50时达到相对稳定状态,之后平稳下降,最终趋于收敛。

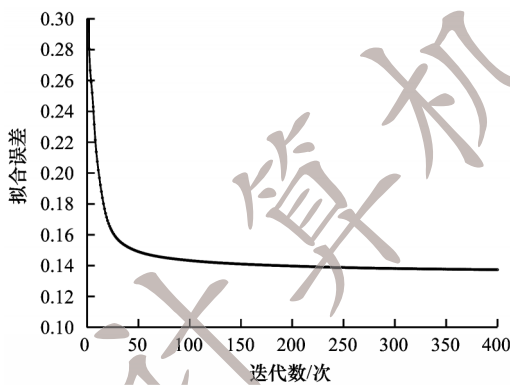


图3 HGNTT-MU算法在ORL数据集上的收敛曲线  
Fig.3 Convergence curve of HGNTT-MU algorithm on the ORL dataset

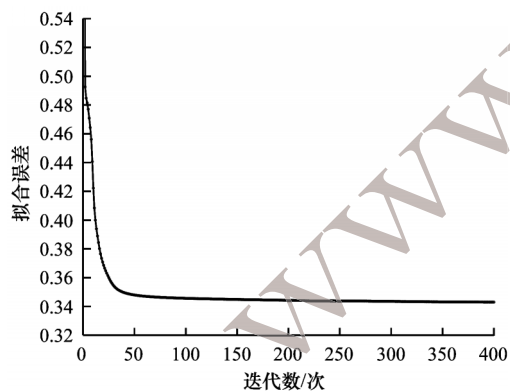


图4 HGNTT-MU算法在Faces95数据集上的收敛曲线  
Fig.4 Convergence curve of HGNTT-MU algorithm on the Faces95 dataset

#### 4 结束语

本文提出一种基于超图正则化的非负张量链分解方法。通过构建超图,发现隐藏在数据间的复杂

多元的流形结构信息。结合NTT模型,保留高阶数据的多线性结构,挖掘数据内部的潜在信息,使其具备较强的数据压缩能力。采用乘法更新方法,对HGNTT模型进行优化求解并证明了其收敛性。在ORL和Faces95公开数据集上的聚类实验结果表明,HGNTT算法相比于同类算法聚类性能更优。下一步将构造自适应超图,得到通用HGNTT模型,以解决在一些具有异常噪声的数据中学习到的超图不理想的问题。

#### 参考文献

- [1] ZHOU G X, CICHOCKI A, XIE S L. Fast nonnegative matrix/tensor factorization based on low-rank approximation [J]. IEEE Transactions on Signal Processing, 2012, 60(6): 2928-2940.
- [2] 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.
- [3] QIU Y C, ZHOU G X, ZHANG Y, et al. Canonical Polyadic Decomposition (CPD) of big tensors with low multilinear rank [J]. Multimedia Tools and Applications, 2021, 80(15): 22987-23007.
- [4] QIU Y C, SUN W J, ZHANG Y, et al. Approximately orthogonal nonnegative Tucker decomposition for flexible multiway clustering [J]. Science China Technological Sciences, 2021, 64(9): 1872-1880.
- [5] KOLDA T G, BADER B W. Tensor decompositions and applications [J]. SIAM Review, 2009, 51(3): 455-500.
- [6] LIU Y, LONG Z, HUANG H, et al. Low CP rank and Tucker rank tensor completion for estimating missing components in image data [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 30: 944-954.
- [7] ZHAO Q B, ZHANG L Q, CICHOCKI A. Bayesian CP factorization of incomplete tensors with automatic rank determination [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1751-1763.
- [8] QIU Y N, ZHOU G X, CHEN X Q, et al. Semi-supervised non-negative Tucker decomposition for tensor data representation [J]. Science China (Technological Sciences), 2021, 64(9): 1881-1892.
- [9] PAN J J, NG M K, LIU Y, et al. Orthogonal nonnegative Tucker decomposition [J]. SIAM Journal on Scientific Computing, 2021, 43(1): 55-81.
- [10] OSELEDETS I V. Tensor-train decomposition [J]. SIAM Journal on Scientific Computing, 2011, 33(5): 2295-2317.
- [11] BENGUA J A, PHIEN H N, TUAN H D, et al. Efficient tensor completion for color image and video recovery: low-rank tensor train [J]. IEEE Transactions on Image Processing, 2017, 26(5): 2466-2479.
- [12] YANG Y, KROMPASS D, TRESP V. Tensor-train recurrent neural networks for video classification [EB/OL].

- [2022-04-11]. <https://arxiv.org/abs/1707.01786>.
- [13] LEE N, PHAN A H, CONG F, et al. Nonnegative tensor train decompositions for multi-domain feature extraction and clustering[C]//Proceedings of International Conference on Neural Information Processing. Berlin, Germany: Springer, 2016: 1-9.
- [14] SHCHERBAKOVA E, TYRTYSHNIKOV E. Nonnegative tensor train factorizations and some applications[M]. Berlin, Germany: Springer, 2020.
- [15] YUAN L H, ZHAO Q B, GUI L H, et al. High-order tensor completion via gradient-based optimization under Tensor Train format[J]. Image Communication, 2019, 73(C): 53-61.
- [16] LI S Z, HOU X W, ZHANG H J, et al. Learning spatially localized, parts-based representation[C]//Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2003: 1-10.
- [17] CAI D, HE X F, WU X Y, et al. Non-negative matrix factorization on manifold[C]//Proceedings of the 8th IEEE International Conference on Data Mining. Washington D. C., USA: IEEE Press, 2009: 63-72.
- [18] LENG C C, ZHANG H, CAI G R, et al. Graph regularized LP smooth non-negative matrix factorization for data representation[J]. IEEE/CAA Journal of Automatica Sinica, 2019, 6(2): 584-595.
- [19] ROWEIS S T, SAUL L K. Nonlinear dimensionality reduction by locally linear embedding[J]. Science, 2000, 290(5500): 2323-2326.
- [20] TENENBAUM J B, DE SILVA V, LANGFORD J C. A global geometric framework for nonlinear dimensionality reduction[J]. Science, 2000, 290(5500): 2319-2323.
- [21] BELKIN M, NIYOGI P. Laplacian eigenmaps for dimensionality reduction and data representation[J]. Neural Computation, 2003, 15(6): 1373-1396.
- [22] CAI D, HE X F, HAN J W, et al. Graph regularized nonnegative matrix factorization for data representation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(8): 1548-1560.
- [23] LI X L, CUI G S, DONG Y S. Graph regularized non-negative low-rank matrix factorization for image clustering[J]. IEEE Transactions on Cybernetics, 2017, 47(11): 3840-3853.
- [24] QIU Y N, ZHOU G X, ZHANG Y, et al. Graph regularized nonnegative tucker decomposition for tensor data representation[C]//Proceedings of 2019 IEEE International Conference on Acoustics, Speech and Signal Processing. Washington D. C., USA: IEEE Press, 2019: 8613-8617.
- [25] SOFUOGLU S E, AVIYENTE S. Graph regularized tensor train decomposition[C]//Proceedings of 2020 IEEE International Conference on Acoustics, Speech and Signal Processing. Washington D. C., USA: IEEE Press, 2020: 3912-3916.
- [26] 吴泽鑫. 图约束非负张量列分解算法及其在特征提取中的应用[D]. 广州: 广东工业大学, 2021.
- WU Z X. Graph constrained nonnegative tensor column decomposition algorithm and its application in feature extraction[D]. Guangzhou: Guangdong University of Technology, 2021. (in Chinese)
- [27] GAO Y, ZHANG Z Z, LIN H J, et al. Hypergraph learning: methods and practices[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(5): 2548-2566.
- [28] SHENG H, AHMED E, DAN Y. On the effect of hyperedge weights on hypergraph learning[J]. Image and Vision Computing, 2017, 57: 89-101.
- [29] SCHÖLKOPF B, PLATT J, HOFMANN T. Learning with hypergraphs: clustering, classification, and embedding[M]. Cambridge, USA: MIT Press, 2007.
- [30] ZENG K, YU J, LI C H, et al. Image clustering by hypergraph regularized non-negative matrix factorization[J]. Neurocomputing, 2014, 138: 209-217.
- [31] 陈璐瑶, 刘奇龙, 许云霞, 等. 基于超图正则化非负 Tucker 分解的图像聚类算法[J]. 计算机工程, 2022, 48(4): 197-205.
- CHEN L Y, LIU Q L, XU Y X, et al. Image clustering algorithm based on hypergraph regularized nonnegative Tucker decomposition[J]. Computer Engineering, 2022, 48(4): 197-205. (in Chinese)
- [32] XU W, LIU X, GONG Y H. Document clustering based on non-negative matrix factorization[C]//Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, USA: ACM Press, 2003: 267-273.
- [33] MACQUEEN J B. Some methods for classification and analysis of multivariate observations[EB/OL]. [2022-04-11]. <https://www.semanticscholar.org/paper/Some-methods-for-classification-and-analysis-of-MacQueen/ac8ab51a86f1a9ae74dd0e4576d1a019f5e654ed?p2df>.