



OPEN

Hypergraph regularized nonnegative triple decomposition for multiway data analysis

Qingshui Liao^{1,2✉}, Qilong Liu² & Fatimah Abdul Razak^{1✉}

Tucker decomposition is widely used for image representation, data reconstruction, and machine learning tasks, but the calculation cost for updating the Tucker core is high. Bilevel form of triple decomposition (TriD) overcomes this issue by decomposing the Tucker core into three low-dimensional third-order factor tensors and plays an important role in the dimension reduction of data representation. TriD, on the other hand, is incapable of precisely encoding similarity relationships for tensor data with a complex manifold structure. To address this shortcoming, we take advantage of hypergraph learning and propose a novel hypergraph regularized nonnegative triple decomposition for multiway data analysis that employs the hypergraph to model the complex relationships among the raw data. Furthermore, we develop a multiplicative update algorithm to solve our optimization problem and theoretically prove its convergence. Finally, we perform extensive numerical tests on six real-world datasets, and the results show that our proposed algorithm outperforms some state-of-the-art methods.

Keywords Nonnegative tensor decomposition, Triple decomposition, Hypergraph regularization, Data analysis

A massive amount of high-dimensional data has been accumulated in social networks, neural networks, data mining, computer vision, and other domains as data extraction technology has advanced. A number of issues arise when analyzing and processing high-dimensional data, such as the need for long computation times and large memory spaces. As a result, dimensionality reduction is commonly conducted prior to further processing and analysis of these data. High-dimensional data is often vectorized to form a larger matrix. Matrix-based methods, such as principal component analysis (PCA)¹, singular value decomposition (SVD)², multiway extensions of the SVD³, and linear discriminant analysis (LDA)⁴, are then used for dimensionality reduction. However, the matrix-based dimensionality reduction methods ignore the internal structure of the data. Therefore, tensor decomposition techniques are used to gain a better understanding of data features. There are some widely used tensor decomposition methods, such as Eckart-Young decomposition⁵, CANDECOMP/PARAFAC (CP) decomposition⁶, Tucker decomposition (TD)⁷, and the family of principal component decomposition models related to TD^{8–12}. TD is the decomposition of a tensor into the product of the core tensor and some factor matrices in different directions. When the core tensor in TD is taken to be the unit tensor, it degenerates to CP decomposition. Different from CP, multiway versions of principal component decompositions related to TD focus on underlining different numbers of main influence components for various multiway data via feature extraction along different modes of models.

TD has been successfully applied in the fields of pattern recognition, cluster analysis, image denoising, and image complementation. Due to the powerful data representation capabilities of TD, many TD variants have been developed in recent years based on reasonable assumptions such as sparsity¹³, smoothness¹⁴, and convolution¹⁵. However, TD faces some challenges when dealing with high-dimensional data: (i) The size of the core tensor in TD grows rapidly as the order of the data increases, which may result in a high cost of calculation and estimation complexity; (ii) TD does not consider the variability in each direction. This variability is widespread in some real data, such as traffic and internet data, where the three modes of the third-order tensor have strong temporal, spatial, and periodic significance¹⁶. To remedy these shortcomings, Qi et al.¹⁷ proposed a bilevel form of triple decomposition (TriD). The triple decomposition for third-order tensors transforms a third-order tensor into a product of three third-order factor tensors. Each factor tensor represents a different meaning and is

¹Department of Mathematical Sciences, Faculty of Science & Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia. ²School of Mathematical Sciences, Guizhou Normal University, Guiyang 550025, People's Republic of China. ✉email: Liaoqingshui2021@163.com; fatima84@ukm.edu.my

of lower dimension in two directions. TriD performs TD on a tensor and triple decomposes the Tucker core at the same time. The number of parameters in TriD is less than that of TD in substantial cases. Therefore, TriD is less costly than TD.

Although TriD has achieved better results in tensor data recovery experiments, it does not take into account the geometrical manifold structure of the data. In the past decade, manifold learning has been widely adopted to preserve the geometric information of original data. Cai et al.¹⁸ explored the geometrical information by constructing a k -nearest neighbor graph and proposed the graph regularized nonnegative matrix factorization (GNMF), which demonstrated promising performance in clustering analysis. To improve the robustness of GNMF, some variants of GNMF have been proposed as described in the literatures^{19–24}. Li et al.²⁵ introduced a manifold regularization term on the core tensor and proposed a manifold regularization nonnegative Tucker decomposition (MR-NTD) method. Qiu et al.²⁶ proposed a graph regularized nonnegative Tucker decomposition (GNTD) method by applying Laplacian regularization to the last nonnegative factor matrix. Liu et al.²⁷ presented a technique known as graph regularized L_p smooth NTD (GSNTD) via embedding graph regularization and L_p smooth constraint into the original model of NTD. Subsequently, Wu et al.²⁸ proposed a manifold regularization nonnegative triple decomposition (MRNTriD) of tensor sets that takes advantage of tensor geometry information. These graph-based manifold learning methods perform well in clustering. They, however, only consider the pairwise relationship between samples and ignore the high-order relationship among samples. Hypergraph learning is a good candidate for solving this problem.

Using a hypergraph to model the high-order relationship between samples will improve classification performance. There are numerous significant methods combined with hypergraphs that work well in clustering tasks: Zeng et al.²⁹ presented a hypergraph regularized nonnegative matrix factorization (HNMF) method. Wang et al.³⁰ introduced a hypergraph regularization to $L_{1/2}$ -NMF (HSNMF) for exploiting spectral-spatial joint structure of hyperspectral images. Huang et al.³¹ constructed a sparse hypergraph for better clustering and proposed a sparse hypergraph regularized NMF (SHNMF) method. Yin et al.³² proposed a hypergraph regularized nonnegative tensor factorization (HyperNTF) method by incorporating hypergraph into nonnegative tensor decomposition. Zhao et al.³³ introduced a hypergraph regularized term into the framework of the nonnegative tensor ring decomposition and proposed a hypergraph regularized nonnegative tensor ring decomposition (HGNTR). To reduce computational complexity and suppress noise, they applied the low-rank approximation trick to accelerate HGNTR (LraHGNTR)³³. Huang et al.³⁴ designed a method to dynamically update the hypergraph and proposed a dynamic hypergraph regularized nonnegative Tucker decomposition (DHNTD) method.

To the best of our knowledge, there is no method to consider higher-order relationships among data sample points in TriD. Inspired by the advantages of hypergraph learning and TriD, in this paper, we present a hypergraph regularized nonnegative triple decomposition (HNTriD) model. HNTriD can explore low-dimensional parts-based representations while preserving detailed complex geometrical information from high-dimensional tensor data. Then, we develop an iterative multiplicative updating algorithm to solve the HNTriD model. The following are the main contributions of this paper:

- HNTriD is a novel dimensionality reduction method by incorporating hypergraph learning into TriD. It is good at dealing with the clustering tasks for tensor data, and the computation cost and containment resources could be greatly reduced.
- HNTriD embraces the merit of the complex connections of observed samples while retaining raw data structural information in dimensionality reduction. We attribute this excellent performance to the hypergraph regularized term's ability which can successfully approximate the inner relationships of original data.
- HNTriD makes sense for some practical applications, such as clustering tasks, because it performs well at multiway data learning and can successfully preserve the important characteristics in dimensionality reduction. Experimental results in some popular datasets, including COIL20, GEORGIA, MNIST, ORL, PIE, and USPS, show that HNTriD outperforms existing rival approaches in cluster analysis.

The remainder of this paper is organized as follows: Section 2 goes over some fundamental concepts, such as NTD, TriD, and hypergraph learning, that will be used in the subsequent sections. The objective function of the HNTriD model is proposed in Section 3, and we discuss the HNTriD optimization algorithm in detail, including the updating rules for the parameters of the model, the convergence analysis of the proposed method, and the computation complexity analysis of HNTriD. In Section 4, we present some experimental results that can be used to validate the efficacy and accuracy of our proposed method. The last section is the conclusion.

Related work

In this section, we briefly overview some basic definitions, including NTD^{32,34,35}, TriD^{17,25}, hypergraph learning^{36–38}. The notations used in this paper are listed in Table 1.

Nonnegative tensor decomposition (NTD)

TD is a popular class of methods for dimensionality reduction of high-dimensional data⁷. The data collected in real life are usually nonnegative, so it makes more physical sense to add nonnegative constraints to all factors in TD. Therefore, we focus on the nonnegative tensor decomposition (NTD). In fact, NTD is a multiway extension of nonnegative matrix factorization (NMF)³⁹, which imposes nonnegative constraints to the TD model³⁵, and it preserves the multilinear structure of data. Given a nonnegative third-order tensor $\mathcal{X} \in \mathbb{R}_+^{n_1 \times n_2 \times n_3}$, NTD can be expressed as a core nonnegative tensor $\hat{\mathcal{X}} \in \mathbb{R}_+^{r_1 \times r_2 \times r_3}$ multiplied by three nonnegative factor matrices $\mathbf{U} \in \mathbb{R}_+^{n_1 \times r_1}$, $\mathbf{V} \in \mathbb{R}_+^{n_2 \times r_2}$, and $\mathbf{W} \in \mathbb{R}_+^{n_3 \times r_3}$, and it can be formulated as

Notations	Descriptions	Notations	Descriptions
\mathbf{U}	A matrix	\mathcal{X}	A tensor
$\ \cdot\ _F$	Frobenius norm	$\text{Tr}(\cdot)$	Trace
$\llbracket \cdot \rrbracket$	Triple product	$\mathbf{Y}_{(n)}$	The mode- n matricization of tensor \mathcal{Y}
\otimes	Kronecker product	\mathcal{X}_{ijl}	The (i, j, l) element of a third-order tensor
\circledast	Hadamard product	$\text{vec}(\cdot)$	The operator vectorizes a subject into a vector
$\mathcal{O}(\cdot)$	Computation cost	\times_n	The mode- n product

Table 1. List of the notations relevant to this paper.

$$\mathcal{X} = \hat{\mathcal{X}} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}. \tag{1}$$

If the smallest integers r_1, r_2, r_3 such that (1) holds, then we call the vector (r_1, r_2, r_3) the Tucker rank. In the process of solving the optimal solution, we usually use its transformation of the mode- n matricization, and (1) can be expressed in the following equivalent forms

$$\mathbf{X}_{(1)} = \mathbf{U} \hat{\mathbf{X}}_{(1)} (\mathbf{W} \otimes \mathbf{V})^\top, \mathbf{X}_{(2)} = \mathbf{V} \hat{\mathbf{X}}_{(2)} (\mathbf{W} \otimes \mathbf{U})^\top, \text{ and } \mathbf{X}_{(3)} = \mathbf{W} \hat{\mathbf{X}}_{(3)} (\mathbf{V} \otimes \mathbf{U})^\top, \tag{2}$$

where $\mathbf{X}_{(n)}$ denotes the mode- n matricization of the tensor \mathcal{X} , “ \otimes ” denotes the Kronecker product of two matrices.

Bilevel form of triple decomposition (TriD)

In the TD and NTD methods, the size of the core tensor grows rapidly as the order of data increases, which may result in a high cost of calculation. To overcome this shortcoming, Qi et al.¹⁷ recently proposed a new form of triple decomposition for third-order tensors, which reduces a third-order tensor to the product of three third-order factor tensors.

Definition 1¹⁷ Let $\hat{\mathcal{X}} = (\hat{x}_{ijl}) \in \mathbb{R}^{r_1 \times r_2 \times r_3}$ be a nonzero tensor. We say that $\hat{\mathcal{X}}$ is the triple product of three third-order square tensors $\mathcal{A} \in \mathbb{R}^{r_1 \times r_1 \times r}$, $\mathcal{B} \in \mathbb{R}^{r_2 \times r_2 \times r}$, and $\mathcal{C} \in \mathbb{R}^{r_3 \times r_3 \times r}$, triple product of the tensors is denoted by

$$\hat{\mathcal{X}} = \llbracket \mathcal{A} \mathcal{B} \mathcal{C} \rrbracket, \tag{3}$$

where \mathcal{A} , \mathcal{B} , and \mathcal{C} are named horizontally square tensor, laterally square tensor, and frontally square tensor, respectively. For $i = 1, 2, \dots, r_1, j = 1, 2, \dots, r_2, l = 1, 2, \dots, r_3$, the elementwise definition of the triple product can be illustrated as

$$\hat{x}_{ijl} = \sum_{p=1}^r \sum_{q=1}^r \sum_{s=1}^r \mathcal{A}_{iqs} \mathcal{B}_{pjs} \mathcal{C}_{pql}. \tag{4}$$

If

$$r \leq \text{mid}\{r_1, r_2, r_3\},$$

where “ $\text{mid}\{\cdot\}$ ” denotes the median, we call (3) is a low rank triple decomposition of $\hat{\mathcal{X}}$. \mathcal{A} , \mathcal{B} , and \mathcal{C} are the factor tensors of $\hat{\mathcal{X}}$. The smallest value of r such that (4) holds is known as the triple rank of $\hat{\mathcal{X}}$, which is denoted as $\text{TriRank}(\hat{\mathcal{X}})=r$. The triple rank of a zero tensor is defined as zero.

If a third-order tensor is decomposed by TD, and its Tucker core is triple decomposed into three tensors simultaneously. Then we get a bilevel form of the triple decomposition, that is shown below.

Definition 2¹⁷ Based on the definition of NTD shown in (1), if the core tensor $\hat{\mathcal{X}}$ has a triple decomposition $\hat{\mathcal{X}} = \llbracket \mathcal{A} \mathcal{B} \mathcal{C} \rrbracket$, where $\text{TriRank}(\hat{\mathcal{X}})=r, \mathcal{A} \in \mathbb{R}_+^{r_1 \times r_1 \times r}, \mathcal{B} \in \mathbb{R}_+^{r_2 \times r_2 \times r}, \text{ and } \mathcal{C} \in \mathbb{R}_+^{r_3 \times r_3 \times r}$. Then \mathcal{X} can be represented as

$$\mathcal{X} = \llbracket \mathcal{A} \mathcal{B} \mathcal{C} \rrbracket \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}. \tag{5}$$

We call (5) a bilevel form of the triple decomposition of \mathcal{X} , which is always referred as TriD. \mathcal{A} , \mathcal{B} , and \mathcal{C} are the inner factor tensors.

From (1), the minimum number in parameters of NTD of the third-order tensor \mathcal{X} is $n_1 r_1 + n_2 r_2 + n_3 r_3 + r_1 r_2 r_3$, where (r_1, r_2, r_3) is the Tucker rank of \mathcal{X} . On the other hand, the number of parameters of TriD is $n_1 r_1 + n_2 r_2 + n_3 r_3 + (r_1 + r_2 + r_3) r^2$, where r is the triple rank of $\hat{\mathcal{X}}$. Generally, the triple rank of $\hat{\mathcal{X}}$ is far less than each of the Tucker rank’s components of the original tensor \mathcal{X} . Then there are substantial cases where the number of parameters of TriD is strictly less than that of the TD.

NTD and TriD are linear dimensionality reduction techniques that may miss the essential nonlinear data structure. Manifold learning, on the other hand, is an effective technique for discovering geometric structure in multiway data, and hypergraph learning is a promising manifold learning method.

Hypergraph learning

To improve clustering performance, it is necessary to maintain the internal hidden geometry structure information, which can be detailed by the hypergraph learning. Given n_3 grayscale image $\{X_1, X_2, \dots, X_{n_3}\}$, each grayscale image can be viewed as a matrix of size $n_1 \times n_2$. These n_3 matrices are stacked to form a tensor \mathcal{X} of size $n_1 \times n_2 \times n_3$. The i -th frontal slice $\mathcal{X}(:, :, i)$ of the tensor \mathcal{X} is exactly the matrix X_i . In addition, we can build a hypergraph $(\mathbb{V}, \mathbb{E}; S)$ to encode the geometrical structure of raw data⁴⁰. Each node $v_i \in \mathbb{V}$ represents a related data X_i and every hyperedge $e_i \in \mathbb{E}$ consists of several nodes that are clustered by some constraints. For each vertex v_i , we form a hyperedge e_i of v_i and the k -neighbours of v_i . For each hyperedge e_i with a weight $s(e_i)$ which is used to measure the similarity of the contained image nodes. The weight $s(e_i)$ can be calculated as follows:

$$s(e_i) = \sum_{X_j \in e_i} \exp\left(\frac{-\|X_i - X_j\|_F^2}{\sigma^2}\right), \tag{6}$$

where $\sigma = \frac{1}{kn_3} \sum_{i=1}^{n_3} \sum_{j \in e_i} \|X_i - X_j\|_F$ denotes the mean distance among all vertices in hyperedge e_i . In particular, we can construct an incidence matrix \mathbf{H} as follows:

$$\mathbf{H}(v_i, e_q) = \begin{cases} 1, & \text{if } v_i \in e_q, \\ 0, & \text{if } v_i \notin e_q. \end{cases}$$

The degrees of a node v_i and a hyperedge e_q can be expressed as

$$d(v_i) = \sum_{e_q \in \mathbb{E}, v_i \in \mathbb{V}} s(e_q) \mathbf{H}(v_i, e_q)$$

and

$$d(e_q) = |e_q| = \sum_{v_i \in \mathbb{V}} \mathbf{H}(v_i, e_q),$$

respectively. We use \mathbf{D}_v , \mathbf{D}_e , and \mathbf{S}_e to denote diagonal matrices whose elements are $d(v_i)$, $d(e_q)$, and $s(e_q)$, respectively.

To make the hypergraph more visual, we show the spatial structure in Figure 1. Herein every $v_j (j = 1, 2, \dots, 9)$ represents a node, and each $e_i (i = 1, 2, \dots, 5)$ denotes a hyperedge.

If two matrix data X_i and X_j are similar in the original raw observation, it is reasonable to assume that their low-dimensional representations w_i and w_j are adjacent to each other. Combined with practical application and theoretical analysis of hypergraph³¹⁻³³, we can assume that w_i and w_j are the corresponding vectors that are related to the nodes v_i and v_j . Then, the following expression can be used to calculate the clustering similarity of the original data X_i and X_j in the low-dimensional approximation.

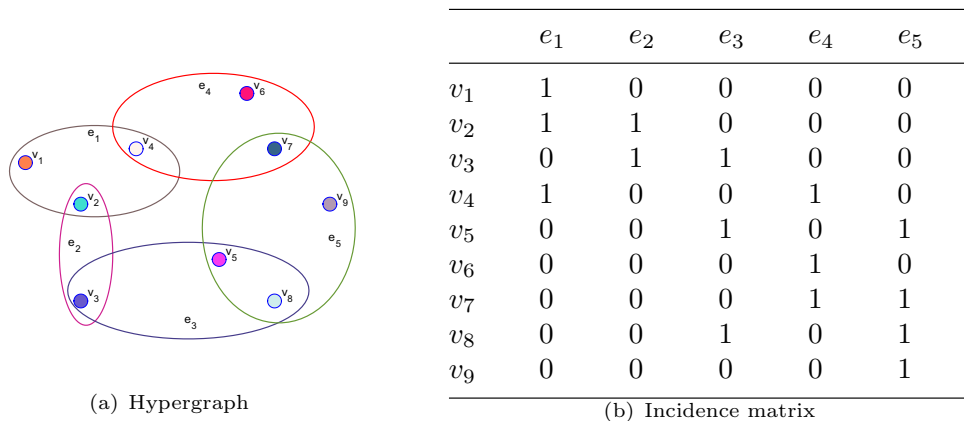


Figure 1. An example of hypergraph and its incident relationship.

$$\begin{aligned}
 & \frac{1}{2} \sum_{e_q \in \mathbb{E}} \sum_{v_i, v_j \in \mathbb{V}} \frac{s(e_q) \mathbf{H}(v_i, e_q) \mathbf{H}(v_j, e_q)}{\delta(e_q)} \|\mathbf{w}_i - \mathbf{w}_j\|_2^2 \\
 &= \sum_{e_q \in \mathbb{E}} \sum_{v_i \in \mathbb{V}} s(e_q) \mathbf{H}(v_i, e_q) \|\mathbf{w}_i\|_2^2 \sum_{v_j \in \mathbb{V}} \frac{\mathbf{H}(v_j, e_q)}{\delta(e_q)} \\
 &\quad - \sum_{e_q \in \mathbb{E}} \sum_{v_i, v_j \in \mathbb{V}} \frac{s(e_q) \mathbf{H}(v_i, e_q) \mathbf{H}(v_j, e_q)}{\delta(e_q)} \mathbf{w}_i^\top \mathbf{w}_j \\
 &= \text{Tr}[\mathbf{W}^\top (\mathbf{D}_v - \mathbf{H} \mathbf{S}_e \mathbf{D}_e^{-1} \mathbf{H}^\top) \mathbf{W}] \\
 &= \text{Tr}(\mathbf{W}^\top \mathbf{L} \mathbf{W}),
 \end{aligned}$$

where $\mathbf{L} = \mathbf{D}_v - \bar{\mathbf{S}}$ is a hypergraph Laplacian matrix that characterizes the data manifold, and $\bar{\mathbf{S}} = \mathbf{H} \mathbf{S}_e \mathbf{D}_e^{-1} \mathbf{H}^\top$.

Hypergraph regularized nonnegative triple decomposition (HNTriD)

TriD is a significant tensor data dimensional reduction algorithm, but it ignores higher-order relationships between the inner parts of raw data and does not consider nonnegative constraints, which may result in a big gap in data clustering performance. Modeling the high-order relationship among samples will help to improve performance. Hypergraph learning is an effective tool for illustrating the inner complex connections of multiway data. By incorporating the hypergraph Laplacian regularized term into the bilevel form of triple decomposition, we get a new method named HNTriD, as shown in the following subsection.

Objective function of HNTriD

Suppose $\mathcal{X} \in \mathbb{R}_+^{n_1 \times n_2 \times n_3}$ be a third-order nonnegative tensor which we stack the samples that are represented by n_3 second-order data $\mathbf{X}_i \in \mathbb{R}_+^{n_1 \times n_2}$ ($i = 1, 2, \dots, n_3$) as the elements of the third mode, each \mathbf{X}_i represents an original data sample of the raw data. Note that $\mathbf{X}_{(3)} = [\text{vec}(\mathbf{X}_1), \text{vec}(\mathbf{X}_2), \dots, \text{vec}(\mathbf{X}_3)]^\top$, unfolding \mathcal{X} along the third mode we can simplify (1) into its matricization form that equals to the third equation of (2), which can be written as

$$\mathbf{X}_{(3)}^\top = \left[(\mathbf{V} \otimes \mathbf{U}) \hat{\mathbf{X}}_{(3)}^\top \right] \mathbf{W}^\top.$$

Let $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_{n_3}]^\top$, each $\mathbf{w}_k \in \mathbb{R}_+^{n_3}$ can be regarded as a low-dimensional representation for the data \mathbf{X}_k under the basis of $(\mathbf{V} \otimes \mathbf{U}) \hat{\mathbf{X}}_{(3)}^\top$.

To improve the multiway data representation ability and brush up operational efficiency, we propose the following HNTriD model, which incorporates the hypergraph constraint into the TriD model. For a given nonnegative tensor, $\mathcal{X} \in \mathbb{R}_+^{n_1 \times n_2 \times n_3}$, HNTriD aims to find three nonnegative tensors $\mathcal{A} \in \mathbb{R}_+^{r_1 \times r \times r}$, $\mathcal{B} \in \mathbb{R}_+^{r \times r_2 \times r}$, and $\mathcal{C} \in \mathbb{R}_+^{r \times r \times r_3}$ and three nonnegative factor matrices $\mathbf{U} \in \mathbb{R}_+^{n_1 \times r_1}$, $\mathbf{V} \in \mathbb{R}_+^{n_2 \times r_2}$, and $\mathbf{W} \in \mathbb{R}_+^{n_3 \times r_3}$ such that

$$\begin{aligned}
 \min_{\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathbf{U}, \mathbf{V}, \mathbf{W}} f_{\text{HNTriD}} &= \frac{1}{2} \|\mathcal{X} - \llbracket \mathcal{ABC} \rrbracket \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}\|_F^2 + \frac{\alpha}{2} \text{Tr}(\mathbf{W}^\top \mathbf{L} \mathbf{W}), \\
 \text{s.t. } \mathcal{A} \geq 0, \mathcal{B} \geq 0, \mathcal{C} \geq 0, \mathbf{U} \geq 0, \mathbf{V} \geq 0, \mathbf{W} \geq 0.
 \end{aligned} \tag{7}$$

The first and second parts of (7) are the reconstruction error term and the hypergraph regularized term, respectively. The reconstruction error term in (7) can be seen as a deep nonnegative tensor decomposition with two layers. The first layer is a TD in the following form

$$\mathcal{X} \approx \mathcal{Y} \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W},$$

where $\mathcal{Y} \times_1 \mathbf{U} \times_2 \mathbf{V}$ denotes the set of multilinear bases of the original data \mathcal{X} and \mathbf{W} denotes the encoding matrix of \mathcal{X} under this set of multilinear bases. The second layer is the triple decomposition, which takes the following form

$$\mathcal{Y} \approx \llbracket \mathcal{ABC} \rrbracket,$$

where each factor tensor represents a different meaning in different application problems. For example, in social networks and transportation data, different characteristics such as temporal stability, spatial correlation, and traffic periodicity may be reflected in each of these three factors. This two-layer decomposition not only reduces the computation required to update the core tensor, but also takes into account the respective advantages of the TD and the triple decomposition. The variable α is an adjustment parameter that is used to measure the importance of the hypergraph regularization term. The hypergraph regularization term preserves the multilateral relationships among the data, so we establish model (7).

HNTriD is used to represent high-dimensional data in a low-dimensional form. To better show the implications of HNTriD, we draw a flowchart to provide a concise overview of the implementation procedure in Figure 2.

Optimization algorithm

When the parameters $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathbf{U}, \mathbf{V}$, and \mathbf{W} are considered simultaneously, the objective function f_{HNTriD} of HNTriD in (7) is not convex. Therefore, obtaining the global optimal solution is difficult. To deal with it, we

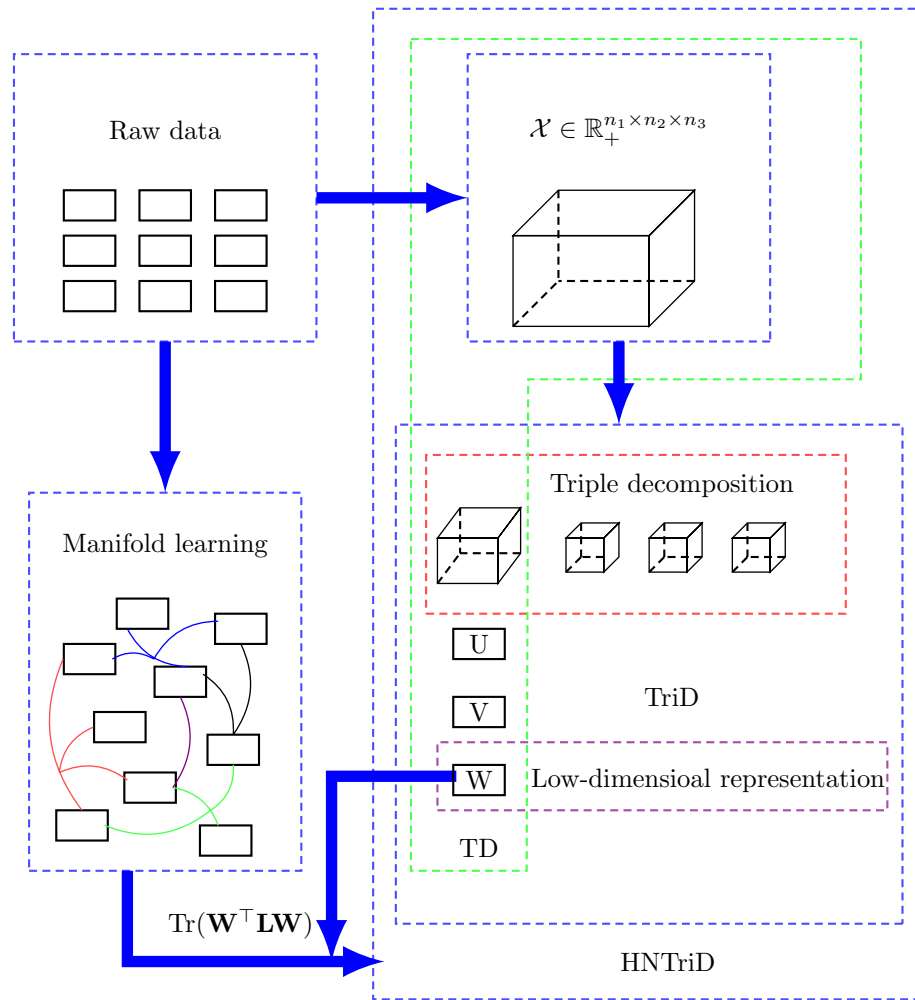


Figure 2. A flowchart used to show the implementation process of HNTriD in data analysis.

introduce an iterative algorithm that achieves a local minimum. To simplify the process of solving the optimal algorithm, we show two important lemmas that will be frequently used.

Lemma 1 ¹⁷ Let $\hat{\mathcal{X}} = [\mathcal{A}\mathcal{B}\mathcal{C}]$, we define three third-order tensors $\mathcal{F} \in \mathbb{R}_+^{r^2 \times r_2 \times r_3}$, $\mathcal{G} \in \mathbb{R}_+^{r_1 \times r^2 \times r_3}$, and $\mathcal{H} \in \mathbb{R}_+^{r_1 \times r_2 \times r^2}$ with entries

$$\mathcal{F}_{kjt} = \sum_{p=1}^r \mathcal{B}_{pjs} \mathcal{C}_{pqt}, \quad \mathcal{G}_{ilt} = \sum_{q=1}^r \mathcal{A}_{iqs} \mathcal{C}_{pqt}, \quad \text{and} \quad \mathcal{H}_{ijm} = \sum_{s=1}^r \mathcal{A}_{iqs} \mathcal{B}_{pjs}, \quad (8)$$

where $k = q + (s - 1)r$, $l = p + (s - 1)r$, and $m = p + (q - 1)r$, respectively. Then

$$\hat{\mathcal{X}} = \mathcal{F} \times_1 \mathbf{A}_{(1)} = \mathcal{G} \times_2 \mathbf{B}_{(2)} = \mathcal{H} \times_3 \mathbf{C}_{(3)}.$$

Lemma 2 Let $\mathbf{M} \in \mathbb{R}^{m \times n}$, $\mathbf{N} \in \mathbb{R}^{n \times p}$, $\mathbf{P} \in \mathbb{R}^{p \times q}$, and $\mathbf{Q} \in \mathbb{R}^{m \times q}$. Then

$$\frac{\partial \|\mathbf{Q} - \mathbf{MNP}\|_F^2}{\partial \mathbf{N}} = 2\mathbf{M}^\top (\mathbf{MNP} - \mathbf{Q})\mathbf{P}^\top. \quad (9)$$

Proof According to

$$\begin{aligned} \|\mathbf{Q} - \mathbf{MNP}\|_F^2 &= \text{Tr} \left[(\mathbf{Q} - \mathbf{MNP})^\top (\mathbf{Q} - \mathbf{MNP}) \right] \\ &= \text{Tr} \left[(\mathbf{MNP})^\top (\mathbf{MNP}) \right] - 2\text{Tr}(\mathbf{PQ}^\top \mathbf{MN}) + \text{Tr}(\mathbf{Q}^\top \mathbf{Q}), \end{aligned}$$

one has

$$\frac{\partial \|Q - MNP\|_F^2}{\partial \text{vec}(\mathbf{N})} = \frac{\partial \text{Tr}[(MNP)^\top (MNP)]}{\partial \text{vec}(\mathbf{N})} - 2 \frac{\partial \text{Tr}(PQ^\top MN)}{\partial \text{vec}(\mathbf{N})}.$$

Combining it with

$$\begin{aligned} \frac{\partial \text{Tr}[(MNP)^\top (MNP)]}{\partial \text{vec}(\mathbf{N})} &= \frac{\partial \text{Tr}[(MNP)^\top (MNP)]}{\partial \text{vec}(MNP)} \cdot \frac{\partial \text{vec}(MNP)}{\partial \text{vec}(MN)} \cdot \frac{\partial \text{vec}(MN)}{\partial \text{vec}(\mathbf{N})} \\ &= 2[\text{vec}(MNP)]^\top (P^\top \otimes \mathbf{I}_m)(\mathbf{I}_q \otimes \mathbf{M}) \\ &= 2[\text{vec}(MNP)]^\top (P^\top \otimes \mathbf{M}) \\ &= 2[\text{vec}(M^\top MNPP^\top)]^\top \end{aligned}$$

and

$$\frac{\partial \text{Tr}(PQ^\top MN)}{\partial \text{vec}(\mathbf{N})} = [\text{vec}(M^\top QP^\top)]^\top$$

yields

$$\frac{\partial \|Q - MNP\|_F^2}{\partial \text{vec}(\mathbf{N})} = 2 \text{vec} [M^\top (MNP - Q)P^\top]^\top.$$

Therefore, $\frac{\partial \|MNP - Q\|_F^2}{\partial \mathbf{N}} = 2M^\top (MNP - Q)P^\top$ is obtained. This completes the proof. □

Solutions of inner factor tensors

When the variables $\mathcal{B}, \mathcal{C}, \mathbf{U}, \mathbf{V}$, and \mathbf{W} are fixed, then the objective function of HNTriD is equivalent to

$$\min_{\mathcal{A} \geq 0} \frac{1}{2} \|\mathcal{X} - \llbracket \mathcal{ABC} \rrbracket \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}\|_F^2. \tag{10}$$

The Lagrange function of the above optimization problem (10) is

$$L_{\mathcal{A}} = \frac{1}{2} \|\mathcal{X} - \llbracket \mathcal{ABC} \rrbracket \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}\|_F^2 - \text{Tr}(\Phi_1 \mathbf{A}_{(1)}^\top). \tag{11}$$

The matricization form of (11) that along the mode-1 is

$$L_{\mathcal{A}} = \frac{1}{2} \|\mathbf{X}_{(1)} - \mathbf{U} \mathbf{A}_{(1)} \mathbf{F}_{(1)} (\mathbf{W} \otimes \mathbf{V})^\top\|_F^2 - \text{Tr}(\Phi_1 \mathbf{A}_{(1)}^\top),$$

where $\mathbf{F}_{(1)}$ is the unfolding form of \mathcal{F} that defined as (8). By Lemma 2, the gradient of $L_{\mathcal{A}}$ with respect to $\mathbf{A}_{(1)}$ is given by

$$\frac{\partial L_{\mathcal{A}}}{\partial \mathbf{A}_{(1)}} = \mathbf{U}^\top \left(\mathbf{U} \mathbf{A}_{(1)} \mathbf{F}_{(1)} (\mathbf{W} \otimes \mathbf{V})^\top - \mathbf{X}_{(1)} \right) (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top - \Phi_1.$$

According to⁴¹, we can take advantage of the Karush-Kuhn-Tucker (KKT) conditions $\frac{\partial L_{\mathcal{A}}}{\partial \mathbf{A}_{(1)}} = \mathbf{0}$ and $\Phi_1 \otimes \mathbf{A}_{(1)} = \mathbf{0}$, then the following equation is satisfied,

$$\left(\mathbf{U}^\top \mathbf{U} \mathbf{A}_{(1)} \mathbf{F}_{(1)} (\mathbf{W} \otimes \mathbf{V})^\top (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top \right) \otimes \mathbf{A}_{(1)} - \left(\mathbf{U}^\top \mathbf{X}_{(1)} (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top \right) \otimes \mathbf{A}_{(1)} = \mathbf{0}.$$

Based on the above equation, we obtain the following updating rule for \mathcal{A} , and

$$[\mathbf{A}_{(1)}]_{ij} \leftarrow [\mathbf{A}_{(1)}]_{ij} \frac{[\mathbf{U}^\top \mathbf{X}_{(1)} (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top]_{ij}}{[\mathbf{U}^\top \mathbf{U} \mathbf{A}_{(1)} \mathbf{F}_{(1)} (\mathbf{W} \otimes \mathbf{V})^\top (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top]_{ij}}. \tag{12}$$

Using the same technique, updating rules for inner factor tensors \mathcal{B} and \mathcal{C} are obtained, which can be expressed as

$$[\mathbf{B}_{(2)}]_{ij} \leftarrow [\mathbf{B}_{(2)}]_{ij} \frac{[\mathbf{V}^\top \mathbf{X}_{(2)} (\mathbf{W} \otimes \mathbf{U}) \mathbf{G}_{(2)}^\top]_{ij}}{[\mathbf{V}^\top \mathbf{V} \mathbf{B}_{(2)} \mathbf{G}_{(2)} (\mathbf{W} \otimes \mathbf{U})^\top (\mathbf{W} \otimes \mathbf{U}) \mathbf{G}_{(2)}^\top]_{ij}} \tag{13}$$

and

$$[\mathbf{C}_{(3)}]_{ij} \leftarrow [\mathbf{C}_{(3)}]_{ij} \frac{[\mathbf{W}^\top \mathbf{X}_{(3)} (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top]_{ij}}{[\mathbf{W}^\top \mathbf{W} \mathbf{C}_{(3)} \mathbf{H}_{(3)} (\mathbf{V} \otimes \mathbf{U})^\top (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top]_{ij}}, \tag{14}$$

respectively.

Solutions of factor matrices

When the variables $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathbf{U}$, and \mathbf{V} are fixed, then the objective function of HNTriD is equivalent to

$$\min_{\mathbf{W} \geq 0} \frac{1}{2} \|\mathcal{X} - \llbracket \mathcal{ABC} \rrbracket \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}\|_F^2 + \frac{\alpha}{2} \text{Tr}(\mathbf{W}^\top \mathbf{LW}). \tag{15}$$

The Lagrange function of the optimization problem (15) is

$$L_{\mathbf{W}} = \frac{1}{2} \|\mathcal{X} - \llbracket \mathcal{ABC} \rrbracket \times_1 \mathbf{U} \times_2 \mathbf{V} \times_3 \mathbf{W}\|_F^2 + \frac{\alpha}{2} \text{Tr}(\mathbf{W}^\top \mathbf{LW}) - \text{Tr}(\Psi_3 \mathbf{W}^\top). \tag{16}$$

By using a transformation of the mode-3 matricization of the tensor \mathcal{X} and $\hat{\mathcal{X}}$, (16) is obtained as follows

$$L_{\mathbf{W}} = \frac{1}{2} \|\mathbf{X}_{(3)} - \mathbf{W} \mathbf{C}_{(3)} \mathbf{H}_{(3)} (\mathbf{V} \otimes \mathbf{U})^\top\|_F^2 + \frac{\alpha}{2} \text{Tr}(\mathbf{W}^\top \mathbf{LW}) - \text{Tr}(\Psi_3 \mathbf{W}^\top).$$

By Lemma 2, the gradient of $L_{\mathbf{W}}$ with respect to \mathbf{W} is given by

$$\frac{\partial L_{\mathbf{W}}}{\partial \mathbf{W}} = (\mathbf{W} \mathbf{C}_{(3)} \mathbf{H}_{(3)} (\mathbf{V} \otimes \mathbf{U})^\top - \mathbf{X}_{(3)}) (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top + \alpha \mathbf{LW} - \Psi_3.$$

Using the Karush-Kuhn-Tucker (KKT) conditions $\frac{\partial L_{\mathbf{W}}}{\partial \mathbf{W}} = \mathbf{0}$ and $\Psi_3 \otimes \mathbf{W} = \mathbf{0}$, the following equation is satisfied,

$$\begin{aligned} & (\mathbf{W} \mathbf{C}_{(3)} \mathbf{H}_{(3)} (\mathbf{V} \otimes \mathbf{U})^\top (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top + \alpha \mathbf{D}_v \mathbf{W}) \otimes \mathbf{W} \\ & - (\mathbf{X}_{(3)} (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top + \alpha \bar{\mathbf{S}} \mathbf{W}) \otimes \mathbf{W} = \mathbf{0}. \end{aligned}$$

Based on the above equation, we obtain the following updating rule for \mathbf{W} , and

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij} \frac{[\mathbf{X}_{(3)} (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top + \alpha \bar{\mathbf{S}} \mathbf{W}]_{ij}}{[\mathbf{W} \mathbf{C}_{(3)} \mathbf{H}_{(3)} (\mathbf{V} \otimes \mathbf{U})^\top (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top + \alpha \mathbf{D}_v \mathbf{W}]_{ij}}. \tag{17}$$

Using the same technique, updating rules for the inner factor matrices \mathbf{U} and \mathbf{V} are obtained, which can be presented as

$$\mathbf{U}_{ij} \leftarrow \mathbf{U}_{ij} \frac{[\mathbf{X}_{(1)} (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top \mathbf{A}_{(1)}^\top]_{ij}}{[\mathbf{U} \mathbf{A}_{(1)} \mathbf{F}_{(1)} (\mathbf{W} \otimes \mathbf{V})^\top (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top \mathbf{A}_{(1)}^\top]_{ij}} \tag{18}$$

and

$$\mathbf{V}_{ij} \leftarrow \mathbf{V}_{ij} \frac{[\mathbf{X}_{(2)} (\mathbf{W} \otimes \mathbf{U}) \mathbf{G}_{(2)}^\top \mathbf{B}_{(2)}^\top]_{ij}}{[\mathbf{V} \mathbf{B}_{(2)} \mathbf{G}_{(2)} (\mathbf{W} \otimes \mathbf{U})^\top (\mathbf{W} \otimes \mathbf{U}) \mathbf{G}_{(2)}^\top \mathbf{B}_{(2)}^\top]_{ij}}, \tag{19}$$

respectively.

Convergence analysis theoretically

In this subsection, the convergence of the iterative updating algorithm is investigated. Our proof will make use of an auxiliary function that is defined as below.

Definition 3 ⁴² $\mathbb{G}(x, \tilde{x})$ is an auxiliary function for $\mathbb{F}(x)$ if the conditions

$$\mathbb{G}(x, \tilde{x}) \geq \mathbb{F}(x) \quad \text{and} \quad \mathbb{G}(x, x) = \mathbb{F}(x)$$

are satisfied.

The auxiliary function is of great help due to the key property that is shown as follows:

Lemma 3 ⁴² *If $\mathbb{G}(x, \tilde{x})$ is an auxiliary function of $\mathbb{F}(x)$, then $\mathbb{F}(x)$ is non-increasing under the update*

$$x^{t+1} = \arg \min_x \mathbb{G}(x, x^t). \tag{20}$$

Now, we are going to show that the update rule for $\mathbf{A}_{(1)}$ shown in (12) is exactly the same as that shown in (20) with a proper auxiliary function. Considering the i th row and j th column entry $[\mathbf{A}_{(1)}]_{ij}$ in $\mathbf{A}_{(1)}$, we use \mathbb{F}_{ij} to denote the part of the objective function (7) that is relevant only to $[\mathbf{A}_{(1)}]_{ij}$. The first and second derivatives of \mathbb{F}_{ij} are

$$\mathbb{F}'_{ij} = \left[\mathbf{U}^\top \left(\mathbf{U} \mathbf{A}_{(1)} \mathbf{F}_{(1)} (\mathbf{W} \otimes \mathbf{V})^\top - \mathbf{X}_{(1)} \right) (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top \right]_{ij}$$

and

$$\mathbb{F}''_{ij} = (\mathbf{U}^\top \mathbf{U})_{ii} \left(\mathbf{F}_{(1)} (\mathbf{W} \otimes \mathbf{V})^\top (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top \right)_{jj},$$

respectively.

Lemma 4 *The function*

$$\begin{aligned} \mathbb{G}(x, [\mathbf{A}_{(1)}]_{ij}^t) = & \mathbb{F}_{ij}([\mathbf{A}_{(1)}]_{ij}^t) + \mathbb{F}'_{ij}([\mathbf{A}_{(1)}]_{ij}^t)(x - [\mathbf{A}_{(1)}]_{ij}^t) \\ & + \frac{\left[\mathbf{U}^\top \mathbf{U} \mathbf{A}_{(1)} \mathbf{F}_{(1)} (\mathbf{W} \otimes \mathbf{V})^\top (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top \right]_{ij}}{2[\mathbf{A}_{(1)}]_{ij}^t} (x - [\mathbf{A}_{(1)}]_{ij}^t)^2 \end{aligned} \tag{21}$$

is an auxiliary function for \mathbb{F}_{ij} , which is only relevant to $[\mathbf{A}_{(1)}]_{ij}$.

Proof Since $\mathbb{G}(x, x) = \mathbb{F}_{ij}(x)$ is obvious, we only need to show that the condition $\mathbb{G}(x, [\mathbf{A}_{(1)}]_{ij}^t) \geq \mathbb{F}_{ij}(x)$ holds. To achieve this, we take into consideration the Taylor series expansion of $\mathbb{F}_{ij}(x)$ which can be formalized as

$$\mathbb{F}_{ij}(x) = \mathbb{F}_{ij}([\mathbf{A}_{(1)}]_{ij}^t) + \mathbb{F}'_{ij}([\mathbf{A}_{(1)}]_{ij}^t)(x - [\mathbf{A}_{(1)}]_{ij}^t) + \frac{1}{2} \mathbb{F}''_{ij}([\mathbf{A}_{(1)}]_{ij}^t)(x - [\mathbf{A}_{(1)}]_{ij}^t)^2. \tag{22}$$

Comparing (21) with (22), we can get that $\mathbb{G}(x, [\mathbf{A}_{(1)}]_{ij}^t) \geq \mathbb{F}_{ij}(x)$ is satisfied as long as

$$\frac{\left[\mathbf{U}^\top \mathbf{U} \mathbf{A}_{(1)} \mathbf{F}_{(1)} (\mathbf{W} \otimes \mathbf{V})^\top (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top \right]_{ij}}{2[\mathbf{A}_{(1)}]_{ij}^t} \geq \frac{1}{2} \mathbb{F}''_{ij}([\mathbf{A}_{(1)}]_{ij}^t)$$

holds, which can be expressed as

$$\begin{aligned} & \left[\mathbf{U}^\top \mathbf{U} \mathbf{A}_{(1)} \mathbf{F}_{(1)} (\mathbf{W} \otimes \mathbf{V})^\top (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top \right]_{ij} \\ & \geq [\mathbf{A}_{(1)}]_{ij}^t (\mathbf{U}^\top \mathbf{U})_{ii} \left(\mathbf{F}_{(1)} (\mathbf{W} \otimes \mathbf{V})^\top (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top \right)_{jj}. \end{aligned} \tag{23}$$

Since

$$\begin{aligned} & \left[\mathbf{U}^\top \mathbf{U} \mathbf{A}_{(1)} \mathbf{F}_{(1)} (\mathbf{W} \otimes \mathbf{V})^\top (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top \right]_{ij} \\ & = \sum_{i_2=1}^{r_1} \sum_{i_3=1}^{r_2} (\mathbf{U}^\top \mathbf{U})_{i_2 i_2} [\mathbf{A}_{(1)}]_{i_2 i_3} [\mathbf{F}_{(1)} (\mathbf{W} \otimes \mathbf{V})^\top (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top]_{i_3 j} \\ & = \sum_{i_2=1, i_2 \neq i}^{r_1} \sum_{i_3=1, i_3 \neq j}^{r_2} (\mathbf{U}^\top \mathbf{U})_{i_2 i_2} [\mathbf{A}_{(1)}]_{i_2 i_3} [\mathbf{F}_{(1)} (\mathbf{W} \otimes \mathbf{V})^\top (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top]_{i_3 j} \\ & \quad + [\mathbf{A}_{(1)}]_{ij}^t (\mathbf{U}^\top \mathbf{U})_{ii} \left(\mathbf{F}_{(1)} (\mathbf{W} \otimes \mathbf{V})^\top (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top \right)_{jj} \\ & \geq [\mathbf{A}_{(1)}]_{ij}^t (\mathbf{U}^\top \mathbf{U})_{ii} \left(\mathbf{F}_{(1)} (\mathbf{W} \otimes \mathbf{V})^\top (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top \right)_{jj}, \end{aligned}$$

which implies (23) holds, then $\mathbb{G}(x, [\mathbf{A}_{(1)}]_{ij}^t) \geq \mathbb{F}_{ij}(x)$ is satisfied. This completes the proof. □

Theorem 1 *The objective function of the HNTriD model (7) is non-increasing under the updating rule $\mathbf{A}_{(1)}$ represented as (12).*

Proof Replacing the auxiliary function $\mathbb{G}(x, x^t)$ of (20) with (21) yields

$$[\mathbf{A}_{(1)}]_{ij}^{t+1} = \arg \min_x \mathbb{G}(x, [\mathbf{A}_{(1)}]_{ij}^t).$$

According to

$$\begin{aligned} & \frac{\partial \mathbb{G}(x, [\mathbf{A}_{(1)}]_{ij}^t)}{\partial x} \\ &= \mathbb{F}'_{ij}([\mathbf{A}_{(1)}]_{ij}^t) + \frac{\left[\mathbf{U}^\top \mathbf{U} \mathbf{A}_{(1)} \mathbf{F}_{(1)} (\mathbf{W} \otimes \mathbf{V})^\top (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top \right]_{ij}}{[\mathbf{A}_{(1)}]_{ij}^t} (x - [\mathbf{A}_{(1)}]_{ij}^t) = 0, \end{aligned}$$

we have

$$[\mathbf{A}_{(1)}]_{ij}^{t+1} = [\mathbf{A}_{(1)}]_{ij}^t \frac{\left[\mathbf{U}^\top \mathbf{X}_{(1)} (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top \right]_{ij}}{\left[\mathbf{U}^\top \mathbf{U} \mathbf{A}_{(1)} \mathbf{F}_{(1)} (\mathbf{W} \otimes \mathbf{V})^\top (\mathbf{W} \otimes \mathbf{V}) \mathbf{F}_{(1)}^\top \right]_{ij}}. \tag{24}$$

Then we can see that (24) agrees with (12), and the Lemma 4 guarantees that (21) is an auxiliary function of \mathbb{F}_{ij} . Based on this, in conjunction with Lemma 3, we can get that f_{HNTriD} is non-increasing under the update rule of (12). The proof is then finished. \square

We are going to state that the update for \mathbf{W} expressed as (17) is equal to the update (20) with an appropriate auxiliary function. Considering the i th row and j th column entry \mathbf{W}_{ij} in \mathbf{W} , we use $\hat{\mathbb{F}}_{ij}$ to denote the part of the objective function (7) that is only relevant to \mathbf{W}_{ij} . The first and second derivatives of $\hat{\mathbb{F}}_{ij}$ are shown below

$$\hat{\mathbb{F}}'_{ij} = \left[\left(\mathbf{W} \mathbf{C}_{(3)} \mathbf{H}_{(3)} (\mathbf{V} \otimes \mathbf{U})^\top - \mathbf{X}_{(3)} \right) (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top + \alpha \mathbf{L} \mathbf{W} \right]_{ij}$$

and

$$\hat{\mathbb{F}}''_{ij} = \left[\mathbf{C}_{(3)} \mathbf{H}_{(3)} (\mathbf{V} \otimes \mathbf{U})^\top (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top \right]_{ij} + \alpha \mathbf{L}_{ii},$$

respectively.

Lemma 5 *The function*

$$\begin{aligned} \hat{\mathbb{G}}(x, \mathbf{W}_{ij}^t) &= \hat{\mathbb{F}}_{ij}(\mathbf{W}_{ij}^t) + \hat{\mathbb{F}}'_{ij}(\mathbf{W}_{ij}^t)(x - \mathbf{W}_{ij}^t) \\ &+ \frac{\left[\mathbf{W} \mathbf{C}_{(3)} \mathbf{H}_{(3)} (\mathbf{V} \otimes \mathbf{U})^\top (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top + \alpha \mathbf{D}_v \mathbf{W} \right]_{ij}}{2 \mathbf{W}_{ij}^t} (x - \mathbf{W}_{ij}^t)^2 \end{aligned} \tag{25}$$

is an auxiliary function for $\hat{\mathbb{F}}_{ij}$, which is only relevant to \mathbf{W}_{ij} .

Proof Since $\hat{\mathbb{G}}(x, x) = \hat{\mathbb{F}}_{ij}(x)$ is obvious, we only need to illustrate that the condition $\hat{\mathbb{G}}(x, x) \geq \hat{\mathbb{F}}_{ij}(x)$ holds. To achieve this, we take into consideration the Taylor series expansion of $\hat{\mathbb{F}}_{ij}(x)$ which can be expressed as follows

$$\hat{\mathbb{F}}_{ij}(x) = \hat{\mathbb{F}}_{ij}(\mathbf{W}_{ij}^t) + \hat{\mathbb{F}}'_{ij}(\mathbf{W}_{ij}^t)(x - \mathbf{W}_{ij}^t) + \frac{1}{2} \hat{\mathbb{F}}''_{ij}(\mathbf{W}_{ij}^t)(x - \mathbf{W}_{ij}^t)^2. \tag{26}$$

Combing (25) with (26) we can find that $\hat{\mathbb{G}}(x, \mathbf{W}_{ij}^t) \geq \hat{\mathbb{F}}_{ij}(x)$ is equivalent to

$$\frac{\left[\mathbf{W} \mathbf{C}_{(3)} \mathbf{H}_{(3)} (\mathbf{V} \otimes \mathbf{U})^\top (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top + \alpha \mathbf{D}_v \mathbf{W} \right]_{ij}}{2 \mathbf{W}_{ij}^t} \geq \frac{1}{2} \hat{\mathbb{F}}''_{ij}(\mathbf{W}_{ij}^t).$$

And the above equation can be rewritten as

$$\begin{aligned} & \left[\mathbf{W} \mathbf{C}_{(3)} \mathbf{H}_{(3)} (\mathbf{V} \otimes \mathbf{U})^\top (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top + \alpha \mathbf{D}_v \mathbf{W} \right]_{ij} \\ & \geq \mathbf{W}_{ij}^t \left[\mathbf{C}_{(3)} \mathbf{H}_{(3)} (\mathbf{V} \otimes \mathbf{U})^\top (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top \right]_{ij} + \alpha \mathbf{W}_{ij}^t \mathbf{L}_{ii}. \end{aligned} \tag{27}$$

Since

$$\begin{aligned}
 & \left[\mathbf{W} \mathbf{C}_{(3)} \mathbf{H}_{(3)} (\mathbf{V} \otimes \mathbf{U})^\top (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top + \alpha \mathbf{D}_v \mathbf{W} \right]_{ij} \\
 &= \sum_{k=1}^{r_3} \mathbf{W}_{ik} \left[\mathbf{C}_{(3)} \mathbf{H}_{(3)} (\mathbf{V} \otimes \mathbf{U})^\top (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top \right]_{kj} + \alpha (\mathbf{D}_v)_{ii} \mathbf{W}_{ij}^t \\
 &= \sum_{k=1, k \neq j}^{r_3} \mathbf{W}_{ik} \left[\mathbf{C}_{(3)} \mathbf{H}_{(3)} (\mathbf{V} \otimes \mathbf{U})^\top (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top \right]_{kj} \\
 &\quad + \mathbf{W}_{ij}^t \left[\mathbf{C}_{(3)} \mathbf{H}_{(3)} (\mathbf{V} \otimes \mathbf{U})^\top (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top \right]_{jj} + \alpha \mathbf{L}_{ii} \mathbf{W}_{ij}^t + \alpha \bar{\mathbf{S}}_{ii} \mathbf{W}_{ij}^t \\
 &\geq \mathbf{W}_{ij}^t \left[\mathbf{C}_{(3)} \mathbf{H}_{(3)} (\mathbf{V} \otimes \mathbf{U})^\top (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top \right]_{jj} + \alpha \mathbf{W}_{ij}^t \mathbf{L}_{ii},
 \end{aligned}$$

which implies (27) holds, and $\hat{\mathbb{G}}(x, \mathbf{W}_{ij}^t) \geq \hat{\mathbb{F}}_{ij}(x)$ is satisfied. This completes the proof. \square

Theorem 2 *The objective function of the HNTriD model (7) is non-increasing under the updating rule \mathbf{W} represented as (17).*

Proof Using (25) to replace the $\mathbb{G}(x, x^t)$ that lies in (20), we obtain

$$\mathbf{W}_{ij}^{t+1} = \arg \min_x \hat{\mathbb{G}}(x, \mathbf{W}_{ij}^t).$$

According to

$$\begin{aligned}
 & \frac{\partial \hat{\mathbb{G}}(x, \mathbf{W}_{ij}^t)}{\partial x} \\
 &= \hat{\mathbb{F}}'_{ij}(\mathbf{W}_{ij}^t) + \frac{\left[\mathbf{W} \mathbf{C}_{(3)} \mathbf{H}_{(3)} (\mathbf{V} \otimes \mathbf{U})^\top (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top + \alpha \mathbf{D}_v \mathbf{W} \right]_{ij}}{\mathbf{W}_{ij}^t} (x - \mathbf{W}_{ij}^t) = 0,
 \end{aligned}$$

we have

$$\mathbf{U}_{ij}^{t+1} = \mathbf{U}_{ij}^t \frac{\left[\mathbf{X}_{(3)} (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top + \alpha \bar{\mathbf{S}} \mathbf{W} \right]_{ij}}{\left[\mathbf{W} \mathbf{C}_{(3)} \mathbf{H}_{(3)} (\mathbf{V} \otimes \mathbf{U})^\top (\mathbf{V} \otimes \mathbf{U}) \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top + \alpha \mathbf{D}_v \mathbf{W} \right]_{ij}}. \tag{28}$$

It is worth noting that (28) is consistent with (17). Lemma 5 ensures that (25) is an auxiliary function of $\hat{\mathbb{F}}_{ij}$, which combined with Lemma 3 results in f_{HNTriD} being non-increasing under the update rule (17). This brings the proof to a close. \square

Applying the same techniques to parameters $\mathcal{B}, \mathcal{C}, \mathbf{U}$, and \mathbf{V} to check the convergence of HNTriD. To summarize, we can obtain that f_{HNTriD} is non-increasing under each of the update rules for inner factor tensors and matrices $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathbf{U}, \mathbf{V}$, and \mathbf{W} while fixing the others. Before imposing our algorithm on real-world datasets for clustering tasks, it is necessary to simplify the calculation formulas of the parameters $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathbf{U}, \mathbf{V}$, and \mathbf{W} , as in the following Remark.

Remark 1 From the form of updating rules of $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathbf{U}, \mathbf{V}$, and \mathbf{W} , it is a fact that each update needs to calculate the Kronecker products which requires costly storage resources. To simplify the produce of updating for mentioned parameters, we take advantages of the tensor property of the mode- n unfolding. Then, we get

$$\mathbf{X}_{(1)} (\mathbf{W} \otimes \mathbf{V}) = (\mathcal{X} \times_2 \mathbf{V}^\top \times_3 \mathbf{W}^\top)_{(1)}$$

and

$$\mathbf{F}_{(1)} (\mathbf{W} \otimes \mathbf{V})^\top (\mathbf{W} \otimes \mathbf{V}) = (\mathcal{F} \times_2 \mathbf{V}^\top \mathbf{V} \times_3 \mathbf{W}^\top \mathbf{W})_{(1)},$$

which means that (12) and (18) can be transformed as

$$[\mathbf{A}_{(1)}]_{ij} \leftarrow [\mathbf{A}_{(1)}]_{ij} \frac{\left[\mathbf{U}^\top (\mathcal{X} \times_2 \mathbf{V}^\top \times_3 \mathbf{W}^\top)_{(1)} \mathbf{F}_{(1)}^\top \right]_{ij}}{\left[\mathbf{U}^\top \mathbf{U} \mathbf{A}_{(1)} (\mathcal{F} \times_2 \mathbf{V}^\top \mathbf{V} \times_3 \mathbf{W}^\top \mathbf{W})_{(1)} \mathbf{F}_{(1)}^\top \right]_{ij}} \tag{29}$$

and

$$\mathbf{U}_{ij} \leftarrow \mathbf{U}_{ij} \frac{\left[(\mathcal{X} \times_2 \mathbf{V}^\top \times_3 \mathbf{W}^\top)_{(1)} \mathbf{F}_{(1)}^\top \mathbf{A}_{(1)}^\top \right]_{ij}}{\left[\mathbf{U} \mathbf{A}_{(1)} (\mathcal{F} \times_2 \mathbf{V}^\top \mathbf{V} \times_3 \mathbf{W}^\top \mathbf{W})_{(1)} \mathbf{F}_{(1)}^\top \mathbf{A}_{(1)}^\top \right]_{ij}}, \quad (30)$$

respectively. According to

$$\mathbf{X}_{(2)}(\mathbf{W} \otimes \mathbf{U}) = (\mathcal{X} \times_1 \mathbf{U}^\top \times_3 \mathbf{W}^\top)_{(2)}$$

and

$$\mathbf{G}_{(2)}(\mathbf{W} \otimes \mathbf{U})^\top (\mathbf{W} \otimes \mathbf{U}) = (\mathcal{G} \times_1 \mathbf{U}^\top \mathbf{U} \times_3 \mathbf{W}^\top \mathbf{W})_{(2)},$$

(13) and (19) can be calculated as

$$[\mathbf{B}_{(2)}]_{ij} \leftarrow [\mathbf{B}_{(2)}]_{ij} \frac{\left[\mathbf{V}^\top (\mathcal{X} \times_1 \mathbf{U}^\top \times_3 \mathbf{W}^\top)_{(2)} \mathbf{G}_{(2)}^\top \right]_{ij}}{\left[\mathbf{V}^\top \mathbf{V} \mathbf{B}_{(2)} (\mathcal{G} \times_1 \mathbf{U}^\top \mathbf{U} \times_3 \mathbf{W}^\top \mathbf{W})_{(2)} \mathbf{G}_{(2)}^\top \right]_{ij}} \quad (31)$$

and

$$\mathbf{V}_{ij} \leftarrow \mathbf{V}_{ij} \frac{\left[(\mathcal{X} \times_1 \mathbf{U}^\top \times_3 \mathbf{W}^\top)_{(2)} \mathbf{G}_{(2)}^\top \mathbf{B}_{(2)}^\top \right]_{ij}}{\left[\mathbf{V} \mathbf{B}_{(2)} (\mathcal{G} \times_1 \mathbf{U}^\top \mathbf{U} \times_3 \mathbf{W}^\top \mathbf{W})_{(2)} \mathbf{G}_{(2)}^\top \mathbf{B}_{(2)}^\top \right]_{ij}}. \quad (32)$$

Similarly, (14) and (17) can be further rewritten as

$$[\mathbf{C}_{(3)}]_{ij} \leftarrow [\mathbf{C}_{(3)}]_{ij} \frac{\left[\mathbf{W}^\top (\mathcal{X} \times_1 \mathbf{U} \times_2 \mathbf{V})_{(3)} \mathbf{H}_{(3)}^\top \right]_{ij}}{\left[\mathbf{W}^\top \mathbf{W} \mathbf{C}_{(3)} (\mathcal{H} \times_1 \mathbf{U}^\top \mathbf{U} \times_2 \mathbf{V}^\top \mathbf{V})_{(3)} \mathbf{H}_{(3)}^\top \right]_{ij}} \quad (33)$$

and

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij} \frac{\left[(\mathcal{X} \times_1 \mathbf{U} \times_2 \mathbf{V})_{(3)} \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top + \alpha \bar{\mathbf{S}} \mathbf{W} \right]_{ij}}{\left[\mathbf{W} \mathbf{C}_{(3)} (\mathcal{H} \times_1 \mathbf{U}^\top \mathbf{U} \times_2 \mathbf{V}^\top \mathbf{V})_{(3)} \mathbf{H}_{(3)}^\top \mathbf{C}_{(3)}^\top + \alpha \mathbf{D}_v \mathbf{W} \right]_{ij}}, \quad (34)$$

respectively.

Hence, the learning rules for the objective function are obtained via the multiplicative update methods described as above. Specifically, we randomly initialize the tensors and factor matrices \mathcal{A} , \mathcal{B} , \mathcal{C} , \mathbf{U} , \mathbf{V} , and \mathbf{W} , then iterate them by (29), (31), (33), (30), (32), and (34). Each iteration ends when the stopping criterion is met. After completing all iterations, we record the operations of the model and examine the convergence at the end of each iteration. The pseudo-code for HNTriD is given in Algorithm 1.

Input: Data tensor $\mathcal{X} \in \mathbb{R}_+^{n_1 \times n_2 \times n_3}$. The number of nearest neighbors k . The algorithm parameters r_1, r_2, r_3, r , and regularization parameter α . The stopping criterion ϵ , and the maximum number of iterations **maxiter**. Let $[f_{\text{HNTrID}}]_0 = 0$.

Output: Factor inner factor tensors $\mathcal{A}, \mathcal{B}, \mathcal{C}$, and factor matrices $\mathbf{U}, \mathbf{V}, \mathbf{W}$;

- 1: Randomly initialize $\mathcal{A} \in \mathbb{R}_+^{r_1 \times r \times r}, \mathcal{B} \in \mathbb{R}_+^{r \times r_2 \times r}, \mathcal{C} \in \mathbb{R}_+^{r \times r \times r_3}, \mathbf{U} \in \mathbb{R}_+^{n_1 \times r_1}, \mathbf{V} \in \mathbb{R}_+^{n_2 \times r_2}, \mathbf{W} \in \mathbb{R}_+^{n_3 \times r_3}$;
- 2: Calculate the matrix $\mathbf{D}_v, \mathbf{D}_e, \mathbf{S}_e$, and \mathbf{H} . Calculate the matrix $\bar{\mathbf{S}}$ using $\bar{\mathbf{S}} = \mathbf{H}\mathbf{S}_e\mathbf{D}_e^{-1}\mathbf{H}^\top$, and calculate the matrix $\mathbf{L} = \mathbf{D}_v - \bar{\mathbf{S}}$;
- 3: **for** $t = 1, 2, \dots, \text{maxiter}$ **do**
- 4: Compute matrix \mathcal{F} as (8) by updated tensors \mathcal{B} and \mathcal{C}
- 5: Update the low-rank tensor \mathcal{A} as (29)
- 6: Update the factor matrix \mathbf{U} as (30)
- 7: Compute matrix \mathcal{G} as (8) by updated tensors \mathcal{A} and \mathcal{C}
- 8: Update the low-rank tensor \mathcal{B} as (31)
- 9: Update the factor matrix \mathbf{V} as (32)
- 10: Compute matrix \mathcal{H} as (8) by updated tensors \mathcal{A} and \mathcal{B}
- 11: Update the low-rank tensor \mathcal{C} as (33)
- 12: Update the factor matrix \mathbf{W} as (34)
- 13: Compute $[f_{\text{HNTrID}}]_t$ by updated tensors $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and factor matrices $\mathbf{U}, \mathbf{V}, \mathbf{W}$.
- 14: **if** $\frac{|[f_{\text{HNTrID}}]_t - [f_{\text{HNTrID}}]_{t-1}|}{[f_{\text{HNTrID}}]_{t-1}} \leq \epsilon$ **then**
- 15: Return $\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathbf{U}, \mathbf{V}, \mathbf{W}$.
- 16: **end if**
- 17: **end for**

Algorithm 1. HNTrID algorithm
Computational complexity analysis

In this subsection, we analyze the computational complexity of the proposed HNTrID model. First, we consider the calculation cost for the tensor-tensor product in (8). In the process of computation tensors \mathcal{F}, \mathcal{G} , and \mathcal{H} takes $\mathcal{O}(r^3 r_2 r_3), \mathcal{O}(r^3 r_1 r_3)$, and $\mathcal{O}(r^3 r_1 r_2)$ operations, respectively. It requires $\mathcal{O}(n_1 r_1^2 + n_2 r_2^2 + n_3 r_3^2)$ operations to calculate symmetric matrices $\mathbf{U}^\top \mathbf{U}, \mathbf{V}^\top \mathbf{V}$, and $\mathbf{W}^\top \mathbf{W}$. It takes $\mathcal{O}(n_1 n_2 n_3 r_2 + n_1 n_3 r_2 r_3 + n_1 r^2 r_2 r_3)$ operations for calculating $(\mathcal{X} \times_2 \mathbf{V}^\top \times_3 \mathbf{W}^\top)_{(1)} \mathbf{F}_{(1)}^\top$. It takes $\mathcal{O}(r^2 r_2^2 r_3 + r^2 r_2 r_3^2 + r^4 r_2 r_3 + r^4 r_1 + n_1 r^2 r_1)$ operations to calculate $\mathbf{U} \mathbf{A}_{(1)} (\mathcal{F} \times_2 \mathbf{V}^\top \mathbf{V} \times_3 \mathbf{W}^\top \mathbf{W})_{(1)} \mathbf{F}_{(1)}^\top$. Since $(\mathcal{X} \times_2 \mathbf{V}^\top \times_3 \mathbf{W}^\top)_{(1)} \mathbf{F}_{(1)}^\top$ and $\mathbf{U} \mathbf{A}_{(1)} (\mathcal{F} \times_2 \mathbf{V}^\top \mathbf{V} \times_3 \mathbf{W}^\top \mathbf{W})_{(1)} \mathbf{F}_{(1)}^\top$ are available, the computational cost for each term, including $\mathbf{U}^\top (\mathcal{X} \times_2 \mathbf{V}^\top \times_3 \mathbf{W}^\top)_{(1)} \mathbf{F}_{(1)}^\top$, $\mathbf{U}^\top \mathbf{U} \mathbf{A}_{(1)} (\mathcal{F} \times_2 \mathbf{V}^\top \mathbf{V} \times_3 \mathbf{W}^\top \mathbf{W})_{(1)} \mathbf{F}_{(1)}^\top$, $(\mathcal{X} \times_2 \mathbf{V}^\top \times_3 \mathbf{W}^\top)_{(1)} \mathbf{F}_{(1)}^\top \mathbf{A}_{(1)}^\top$, and $\mathbf{U} \mathbf{A}_{(1)} (\mathcal{F} \times_2 \mathbf{V}^\top \mathbf{V} \times_3 \mathbf{W}^\top \mathbf{W})_{(1)} \mathbf{F}_{(1)}^\top \mathbf{A}_{(1)}^\top$, is equal to $\mathcal{O}(n_1 r^2 r_1)$. Then, the cost of computing the update rules of \mathcal{A} in (29) is about $\mathcal{O}(r^2 r_1)$. Assume that integers r_1, r_2, r_3 , and r are of the same order of magnitude and they are much smaller than n_1, n_2 , and n_3 . We claim that the total computational cost of computing the update rule of \mathcal{A} in (29) and \mathbf{U} in (30) is approximately

$$\mathcal{O}(n_1 n_2 n_3 r + n_1 n_3 r^2 + n_1 r^4 + r^6)$$

Similarly, the total computational cost of computing updating rules for \mathcal{B} in (31) and \mathbf{V} in (32) is about

$$\mathcal{O}(n_1 n_2 n_3 r + n_2 n_3 r^2 + n_2 r^4 + r^6).$$

The total computational cost of updating the rules for \mathcal{C} in (33) and \mathbf{W} in (34) is approximately

$$\mathcal{O}(n_1 n_2 n_3 r + n_1 n_2 r^2 + n_3 r^4 + r^6).$$

Therefore, we can get the total calculation cost of the HNTrID algorithm approximately as

$$\mathcal{O}(n_1 n_2 n_3 r + (n_1 n_3 + n_2 n_3 + n_1 n_2) r^2 + (n_1 + n_2 + n_3) r^4 + r^6).$$

Experiments

To check the validation of our proposed HNTrID algorithm for clustering data with dimensionality reduction, we run experiments on six popular datasets and compare the results of (7) with that of the related state-of-the-art methods, including NMF⁴², GNMF¹⁸, HNMF²⁹, HSNMF³¹, SHNMF⁴³, HGNTN³³, LraHGNTN³³, HyperNTF³², and TriD¹⁷. All the simulations will be performed on a desktop computer equipped with an Intel (R) Core (TM) i5-10400F CPU at 2.90 GHz and 16 GB of memory, running MATLAB 2015a in Windows 10.

Datasets

The clustering performance is evaluated on six widely used datasets, including COIL20, GEORGIA, MNIST, ORL, PIE, and USPS. The general statistical information of the datasets is summarized in Table 2, including the samples, sizes, and categories that were used in the numerical modeling tests of this paper. A brief overview of the mentioned datasets is presented below.

- COIL20 (<https://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php>): It is a grayscale image dataset comprised of photographs taken from 20 different individuals, and each person was photographed 72 pieces of images from different angles. After resizing each image to 32×32 , we can get a third-order tensor $\mathcal{Y} \in \mathbb{R}_+^{32 \times 32 \times 1,440}$.
- GEORGIA (http://www.anefian.com/research/face_reco.htm): It is a colored JPG image dataset, every image was drawn from 50 people and each person was photographed 15 pieces of images with cluttered backgrounds. The images used in this paper have been converted to grayscale and resized to 32×32 . We can obtain a tensor of third order, which defined as $\mathcal{Y} \in \mathbb{R}_+^{32 \times 32 \times 750}$.
- MNIST (<http://yann.lecun.com/exdb/mnist/>): It is a handwritten digit image dataset, and each image is 28×28 in size. More than 60,000 digit images were collected in the MNIST dataset range from “0” to “9”. In the numerical tests of this paper, we chose 100 images randomly for each single digit. Thus, the chosen images can be presented as a third-order tensor $\mathcal{Y} \in \mathbb{R}_+^{28 \times 28 \times 1,000}$.
- ORL (<https://github.com/saeid436/Face-Recognition-MLP/tree/main/ORL>): It is a dataset that includes 400 grayscale face images of 40 different people collected from different facial expressions, various facial details, and varying lighting, and each image is in size of 112×92 . A third-order tensor can be defined as $\mathcal{Y} \in \mathbb{R}_+^{112 \times 92 \times 400}$.
- PIE (http://www.ri.cmu.edu/projects/project_418.html): It is a dataset containing over 40,000 facial images collected from 68 different individuals. These images were taken in a variety of poses, lighting conditions, and expressions. We randomly selected 53 people with 22 different facial images for our numerical tests. We converted them to gray-level and resize them to 32×32 . Then the selected images can be expressed as a third-order tensor $\mathcal{Y} \in \mathbb{R}_+^{32 \times 32 \times 1,166}$.
- USPS (<https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#usps>): It is a dataset that includes 11,000 grayscale handwritten digits (from “0” to “9”) that are 16×16 in size. In the simulation tests of this paper, we chose 100 images at random for each digit. On this basis, we can build a third-order tensor $\mathcal{Y} \in \mathbb{R}_+^{16 \times 16 \times 1,000}$.

Evaluation metrics

Clustering analysis groups samples only according to the sample data itself and its aim is to group different objects into different groups according to the controlled conditions. The way to evaluate the efficiency of the clustering methods is that objects within groups are similar to each other, while objects differ from group to group. The greater the similarity within the group, the greater the difference between the groups, the better the clustering effect. As we know, the ACC, NMI, and PUR are widely used assessment criteria^{44,45} of clustering algorithm. The accuracy (ACC) can be defined as

$$\text{ACC}(\bar{x}_i, x_i) = \frac{1}{n} \sum_1^n \delta(\bar{x}_i, \text{map}(x_i)),$$

where n is the number of samples in datasets, \bar{x}_i and x_i denote the cluster sample and the original sample, respectively. The symbol $\text{map}(\cdot)$ indicates the matchup relationship mapping function, which is responsible for matching the cluster samples and original samples. The symbol $\delta(\cdot, \cdot)$ is the delta function shown as follows

$$\delta(\bar{x}_i, \text{map}(x_i)) = \begin{cases} 1, & \text{if } x_i \text{ is mapped into } \bar{x}_i, \\ 0, & \text{otherwise.} \end{cases}$$

In general, the agreement between two clusters can be measured with the mutual information (MI), which is widely used in clustering applications. Given two discrete random variables \bar{X} and X which stand for the cluster

Item	Dataset					
	COIL20	GEORGIA	MNIST	ORL	PIE	USPS
Sample	1,440	750	1,000	400	1,166	1,000
Size	32×32	32×32	28×28	112×92	32×32	16×16
Category	20	50	10	40	53	10

Table 2. Descriptions of the relevant six datasets used in this paper.

label sets and true label sets, \bar{x} and x are selected arbitrarily from \bar{X} and X , respectively. Then, the MI can be measured by

$$MI(\bar{X}, X) = \sum_{x \in X} \sum_{\bar{x} \in \bar{X}} p(\bar{x}, x) \log \left(\frac{p(\bar{x}, x)}{p(\bar{x})p(x)} \right),$$

where $p(\bar{x})$ and $p(x)$ are the edge probability distribution function which denote the probabilities of the samples. The $p(\bar{x}, x)$ denotes the joint probability distribution function of \bar{X} and X which means that the object belongs to category \bar{X} and category X at the same time. To force the score to have an upper bound, we take the NMI as one of the evaluation criterion, and the definition is

$$NMI(\bar{X}, X) = \frac{MI(\bar{X}, X)}{\max(T(\bar{X}), T(X))},$$

where $T(\bar{X})$ and $T(X)$ are the entropy of the cluster label set \bar{X} and the entropy of the true label set X . In this way, the score ranges of $NMI(\bar{X}, X)$ is from 0 to 1.

The purity (PUR) of a clustering algorithm is a simple assessment format which only have to calculate the proportion of the correct clustering to the total. In other words, the PUR is to scale the degree of correctness of measurement, the PUR score of a cluster is observed by a weighted sum of the PUR values of the respective clusters, which is denoted by

$$PUR(\bar{X}, X) = \frac{1}{n} \sum_{j=0}^k \max_i |\bar{x}_j \cap x_i|,$$

where $\bar{X} = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_k)$ is the cluster category set, the \bar{x}_i denotes the i th cluster set. $X = (x_1, x_2, \dots, x_k)$ is the original datasets that need to be clustered, x_i represents the i th original object. The total number of the objects is n that need to be clustered and the function $|\cdot|$ denotes the cardinality of a set.

Algorithms for comparison

To ensure the clustering performance, we compare the proposed HNTriD model with the following state-of-the-art clustering algorithms.

- NMF⁴²: It incorporates nonnegative constraint into two factor matrices decomposed from the original matrix.
- GNMF¹⁸: It imposes the graph constraint to the coefficient matrix of the NMF method.
- HNMF²⁹: It incorporates the hypergraph constraint into the coefficient matrix of the NMF method.
- HSNMF³⁰: It imposes the hypergraph constraint on the coefficient matrix based on the $L_{1/2}$ -NMF method.
- SHNMF³¹: It takes the sparse hypergraph as a regularization and adds it to the NMF framework.
- HGNTNTR³³: It includes the hypergraph constraint on the last TR core tensor and a nonnegative constraint on TR factor tensors.
- LraHGNTNTR³³: It is the low-rank approximation of HGNTNTR.
- HyperNTF³²: It imposes a hypergraph constraint on the last factor matrix of the CP model and limits all factor matrices to be nonnegative.
- TriD¹⁷: It is a bilevel form of the triple decomposition of a third-order tensor.

Parameters selection

To achieve the best performance, some critical parameters in the experimental simulations needed to be adjusted. In all tests, let $\epsilon = 10^{-5}$ and the maximum number of iterations be 1000 unless otherwise specified. We set the regularized term α at the grid of $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100, 1000\}$, and the k -nearest neighbors are chosen from $\{3, 4, 5, 6, 7\}$. The parameters r_1 and r_2 are integers empirically chosen from $\{3, 4, \dots, 32\}$, and the integer r is chosen from $\{2, 3, \dots, 20\}$. Furthermore, we choose the third mode, r_3 , as the number of categories in the related datasets, as shown in Table 2. In our experiments, we let one of the parameters r_1, r_2, r, k, α varies in the grid given above, and the rest of the parameters were fixed, and the parameters corresponding to the maximum

Dataset	Optimal parameter					
	r_1	r_2	r_3	r	k	α
COIL20	17	17	20	4	4	1000
GEORGIA	5	5	50	5	3	0.01
MNIST	15	32	10	15	5	10
ORL	11	11	40	11	5	0.01
PIE	8	6	53	19	6	0.01
USPS	3	16	10	2	4	100

Table 3. List of parameters' values corresponding to the maximum NMI of HNTriD on six datasets.

values of NMI in the experiments were recorded. The optimal parameters corresponding to each dataset are given in Table 3.

In Figure 3, we show the effect of the parameters α and k on the three indicators ACC, NMI, and PUR on different datasets. In subplots (a), (c), and (e) of Figure 3, the remaining parameters except α are taken as in Table 3. In subplots (b), (d), and (f) of Figure 3, the remaining parameters except k are taken as in Table 3.

From Figure 3, we can conclude that when the parameter α is set to 10^3 , 10^{-2} , 10 , 10^{-2} , 10^{-2} , and 10^2 , the ACC, NMI, and PUR all perform better in clustering tasks on COIL20, GEORGIA, MNIST, ORL, PIE, and USPS

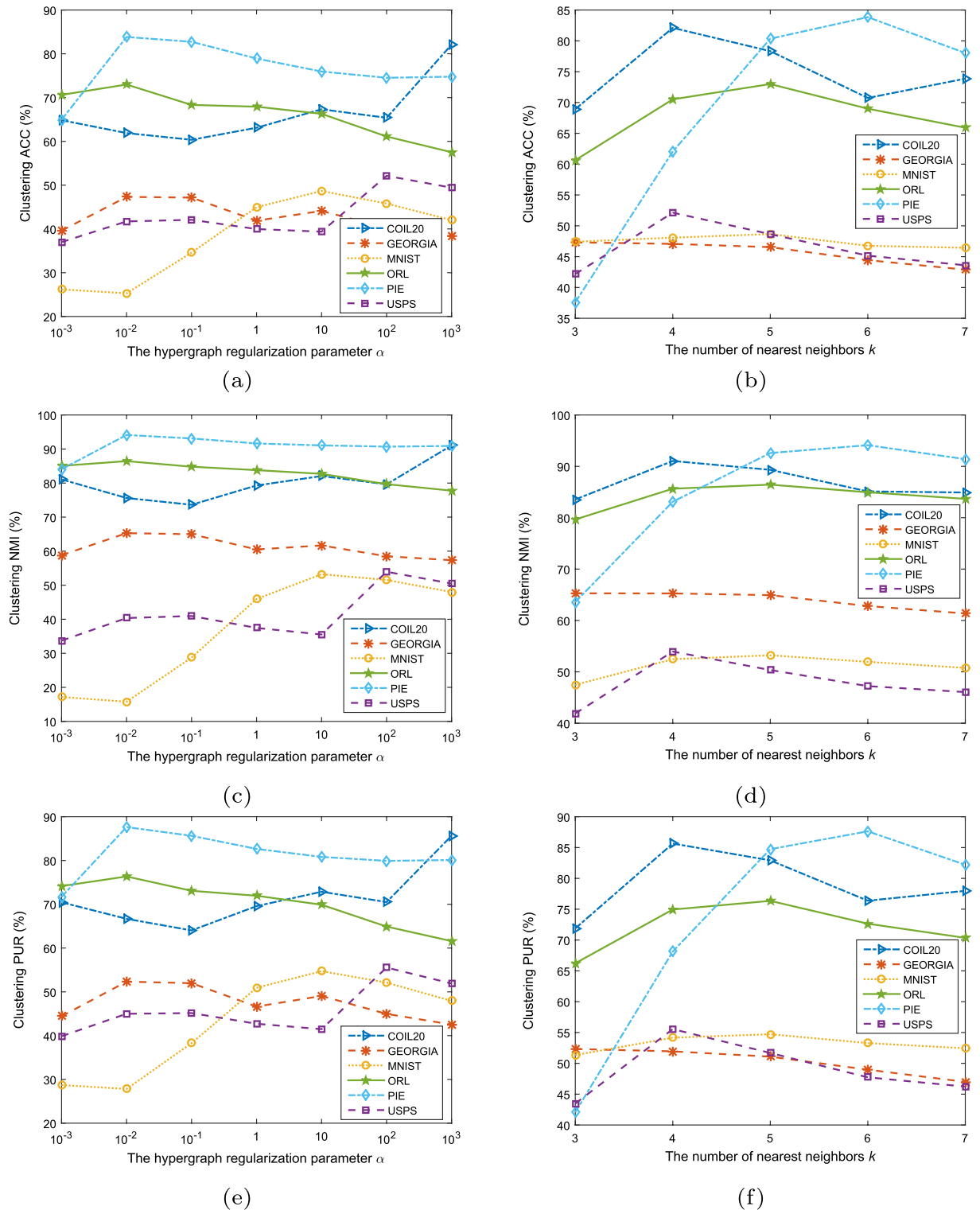


Figure 3. The clustering performance of the HNTrID model varies with different α and the number of nearest neighbors k .

datasets. The parameter k is set to 4, 3, 5, 5, 6, and 4, the ACC, NMI, and PUR achieve better results on COIL20, GEORGIA, MNIST, ORL, PIE, and USPS datasets, respectively.

Convergence study experimentally

In Section 3.3, we demonstrated that our HNTrID algorithm is non-increasing in theory. Here, we validate it using six HNTrID convergence curves tested from six related datasets, which are shown in Figure 4. There are two important key features that can be identified from Figure 4. First, as the number of iterations increases, the

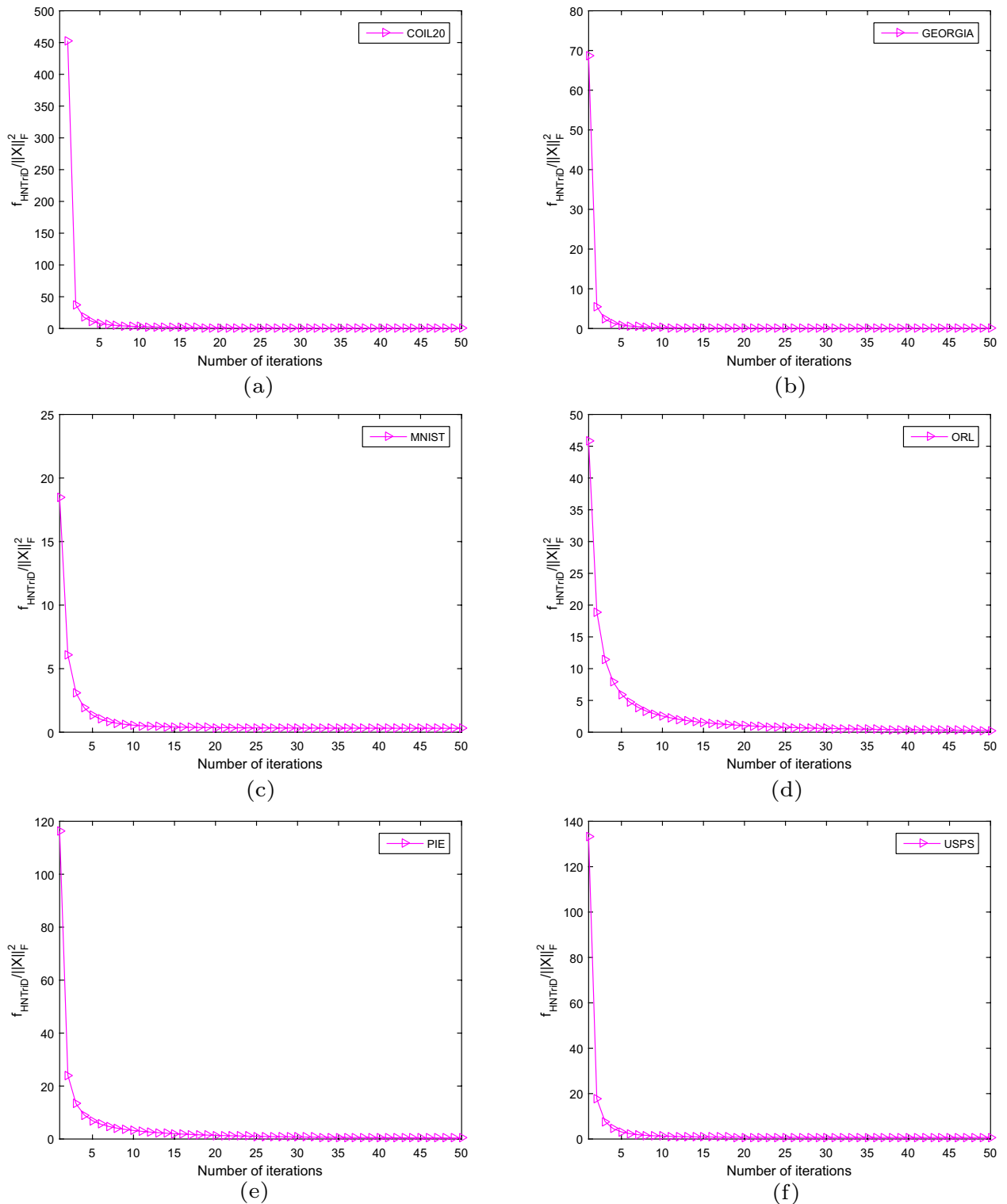


Figure 4. Convergence report of the proposed algorithm on six datasets.

objective function value of HNTriD decreases. Second, the convergence report states that the curve declines rapidly and reaches a relatively stable state within thirty iterations. To summarize, HNTriD experiments show that our method works well on the six datasets mentioned above.

Experimental comparison

To validate the effectiveness of the proposed HNTriD method, we compare it to some state-of-the-art methods under the assessment criteria ACC, NMI, and PUR. For HNTriD, the parameters are selected as in Table 3. For each method embodied with manifold learning, we set the regularized term α at the grid of $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100, 1000\}$, and the k -nearest neighbors are chosen from $\{3, 4, 5, 6, 7\}$. For methods based on TD, such as TriD and HyperNTE, we take the dimension of the third direction of the core tensor to be the class number of the original data. For methods based on tensor ring decomposition, such as HGNTD and LraHGNTD, we take the product of the first order and the third order to be the class number of the original data. The remaining parameters in the comparison algorithm are adjusted on the grid taken by HNTriD. First, we run a number of numerical tests to compare the clustering effect across different datasets. Second, statistical significance comparison is performed on COIL20 and MNIST using the t-test. Third, we present 2-D visualizations of different methods for clustering results on the COIL20 dataset and then complete the comparison tests by means of the t-SNE technique⁴⁶. Finally, we compare the amount of time they took to finish clustering tasks on six related real-world datasets.

Numerical comparison results

All experiments are run on the same sub-raw datasets, which are chosen at random from the corresponding database. Each experimental result is obtained only after the process has been repeated 100 times. The numerical tests, in particular, are performed in two steps. The first step is to choose a group of objects at random from the raw data and then decompose them into corresponding sub-raw data based on the parametric form of the model. To ensure that the experimental results are as accurate as possible to the real-world data clustering situation. We repeat the first step 10 times to obtain 10 groups of sub-raw data. In the second step, we use the K-means method to compute the evaluation index value for each group of sub-raw data. As before, we repeat the second step 10 times to obtain 10 evaluation values for each group of sub-raw data. Throughout the experiment, we can

Method	Dataset					
	COIL20	GEORGIA	MNIST	ORL	PIE	USPS
NMF	57.51±4.84	40.52±1.90	48.42±3.72	66.34±3.72	65.62±3.46	41.62±3.18
GNMF	69.32±3.45	41.54±2.10	49.35 ± 5.78	68.29±3.61	64.84±4.07	44.64±3.35
HNMF	75.13±3.11	41.88±1.69	46.33±5.96	68.61±3.05	64.33±3.45	44.84±3.69
HSNMF	74.59±3.01	38.19±1.83	45.29±5.74	41.28±1.65	<u>76.65 ± 1.96</u>	40.73±2.52
SHNMF	58.03±4.88	40.52±1.84	48.25±3.59	66.67±3.66	65.63±4.08	41.39±2.43
HGNTD	75.97±3.24	35.79±2.35	44.90±6.42	65.91±3.59	58.90±4.51	50.65±3.21
LraHGNTD	75.61±3.07	<u>43.61 ± 1.53</u>	43.86±6.27	64.35±3.17	46.32±3.51	47.56±2.99
HyperNTE	<u>77.15 ± 4.04</u>	41.86±1.51	45.41±4.40	<u>71.78 ± 2.23</u>	52.34±2.74	52.66 ± 4.52
TriD	50.57±4.18	31.50±1.53	43.62±4.08	54.30±3.10	74.54±4.25	50.96±5.15
HNTriD	82.11 ± 2.53	47.32 ± 2.06	<u>48.66 ± 6.14</u>	72.99 ± 2.64	83.85 ± 3.27	<u>52.13 ± 4.08</u>

Table 4. Quantitative clustering (ACC%±std%) of different methods on six datasets. Significant values are bold.

Method	Dataset					
	COIL20	GEORGIA	MNIST	ORL	PIE	USPS
NMF	70.82±2.06	60.03±1.13	45.71±2.34	82.77±1.79	85.33±1.39	38.08±2.30
GNMF	82.92±2.58	60.94±1.19	<u>52.02 ± 4.58</u>	84.57±1.62	84.75±1.80	46.08±3.56
HNMF	87.81±1.52	61.32±1.02	49.92±4.55	84.98±1.52	84.69±1.45	46.18±2.97
HSNMF	87.65±1.60	57.74±1.04	49.26±4.52	61.52±1.09	90.56±0.67	37.08±1.69
SHNMF	72.40±2.37	60.02±1.11	45.66±2.48	83.04±1.82	85.19±1.80	38.30±2.27
HGNTD	88.24±1.43	56.07±1.67	49.42±0.58	82.53±1.64	82.81±1.81	52.33±2.30
LraHGNTD	88.15±1.29	<u>62.47 ± 0.91</u>	49.57±4.82	82.18±1.51	75.84±2.23	50.16±2.19
HyperNTE	<u>88.71 ± 7.12</u>	60.78±0.95	48.40±4.18	<u>85.27 ± 0.87</u>	79.19±1.24	54.07 ± 2.49
TriD	66.60±2.42	52.10±1.24	38.35±3.39	73.84±2.13	<u>90.98 ± 1.69</u>	45.82±4.31
HNTriD	91.04 ± 0.83	65.29 ± 1.35	53.22 ± 4.23	86.44 ± 1.22	94.12 ± 1.05	<u>53.95 ± 2.42</u>

Table 5. Quantitative clustering (NMI%±std%) of different methods on six datasets. Significant values are bold.

method	Dataset					
	COIL20	GEORGIA	MNIST	ORL	PIE	USPS
NMF	60.33±3.91	43.27±1.69	52.60±3.33	71.11±2.83	72.09±2.51	43.83±2.63
GNMF	75.21±2.83	44.41±1.80	55.10 ± 5.29	73.25±2.76	71.31±3.05	47.34±3.17
HNMF	80.52±2.29	44.71±1.55	52.10±5.03	73.69±2.34	70.83±2.56	47.39±2.89
HSNMF	80.20±2.31	40.75±1.65	51.25±4.71	44.02±1.94	80.48±1.53	43.32±2.29
SHNMF	61.12±4.06	43.46±1.70	52.62±3.33	71.48±3.11	71.96±2.96	43.33±2.21
HGNTR	81.19±2.25	38.77±2.15	50.50±6.14	69.98±3.04	64.40±3.63	53.60±2.66
LraHGNTR	81.06±2.05	<u>46.71 ± 1.41</u>	50.00±5.75	68.67±2.59	51.16±3.41	50.18±2.62
HyperNTF	<u>81.80 ± 3.30</u>	45.90±1.20	51.20±4.15	<u>74.84 ± 1.70</u>	59.16±1.99	56.73 ± 3.54
TriD	54.16±3.40	34.25±1.51	47.27±3.96	59.55±2.73	<u>80.64 ± 3.17</u>	52.70±4.97
HNTriD	85.65 ± 1.64	52.33 ± 1.62	<u>54.72 ± 5.32</u>	76.36 ± 2.00	87.63 ± 2.35	<u>55.55 ± 3.55</u>

Table 6. Quantitative clustering (PUR%±std%) of different methods on six datasets. Significant values are bold.

receive 100 evaluation index values and calculate the average value as the performance result for each method. Finally, we report the average performance in Tables 4, 5, and 6. Simultaneously, we record the time spent by each method performing clustering tasks on each dataset, and the results are shown in Figure 6.

Tables 4, 5, and 6, present experimental results demonstrating correlation algorithm clustering performance on six datasets. The advanced clustering method can be found in the tables under the quantitative clustering rules. For ease of observation, we highlight the best data in bold and the second ones in a slight underline. The experimental results presented above lead us to the following conclusion: (i) In terms of clustering performance, as measured by ACC, NMI, and PUR, our proposed method outperforms others in the majority of cases, and our experimental results are second-best, if not the best. (ii) The best experimental results in the mass are located in tensor-based methods. Because tensor methods consider more information from the raw data. (iii) The HNTriD method outperforms other tensor-based decomposition methods in most cases. Because it inherits the previous algorithm's excellent characteristics, including TriD, and preserves the data's multi-linear structure. Experiments show that the proposed HNTriD algorithm performs well in clustering tasks.

Statistical significance comparison

A t-test is a statistical technique used to determine if there is a significant difference between two groups of data. It functions as an important tool in hypothesis testing and aids researchers in determining whether two groups are genuinely distinct^{47,48}. Subsequently, we examine the statistical significance of the disparity between HNTriD and some typical approaches using t-test. Similar to⁴⁹, we take the significance level of $p < 0.05$ in the t-test to draw the difference. If our approach outperforms a compared method in a comparison test and the difference is statistically significant (t-test, $p < 0.05$), we record it as significant better or worse for one time. If the difference between our approach and a compared method is not statistically significant, then we say that they are comparable. We use (a, b, c) to display comparison results. Three integers inside the brackets respectively correspond to the number of times that the performance of our method is significantly better than, comparable to, significantly worse than a related method. We compare 10 algorithms (including HNTriD) on both the COIL20

	Metrics	NMF	GNMF	HNMF	HSNMF	SHNMF	HGNTR	LraHGNTR	HyperNTF	TriD
HNTriD	ACC	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(9,1,0)
	NMI	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)
	PUR	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)	(10,0,0)
	overall	(30,0,0)	(30,0,0)	(30,0,0)	(30,0,0)	(30,0,0)	(30,0,0)	(30,0,0)	(30,0,0)	(29,1,0)

Table 7. The t-test comparison results of different methods on COIL20.

	Metrics	NMF	GNMF	HNMF	HSNMF	SHNMF	HGNTR	LraHGNTR	HyperNTF	TriD
HNTriD	ACC	(3,5,2)	(3,5,2)	(5,3,2)	(10,0,0)	(2,4,4)	(3,6,1)	(7,2,1)	(7,1,2)	(5,5,0)
	NMI	(9,1,0)	(6,3,1)	(7,2,1)	(10,0,0)	(8,2,0)	(7,2,1)	(9,1,0)	(8,1,1)	(10,0,0)
	PUR	(6,4,0)	(6,3,1)	(6,6,2)	(10,0,0)	(4,4,2)	(3,6,1)	(6,4,0)	(9,0,1)	(9,1,0)
	overall	(18,10,2)	(15,11,4)	(18,7,5)	(30,0,0)	(14,10,6)	(13,14,3)	(22,7,1)	(24,2,4)	(24,6,0)

Table 8. The t-test comparison results of different methods on MNIST.

and MNIST datasets. In each comparison, we run all the compared algorithms 10 times, and we repeat each group of comparison experiments 10 times. Specifically, the statistical test results are presented in Tables 7 and 8.

According to comparison tests show in Table 7, our method is significantly superior to the compared methods on the COIL20 dataset. The experimental results on MNIST demonstrate a clear decline in performance compared to the experimental findings on COIL20. However, from the overview of all metrics' evaluation, the results still demonstrate a high level of performance when compared to other approaches. Based on the information provided in Tables 7 and 8, it is evident that our method demonstrates significant statistical advancements compared to the listed methods in most cases. The statistical test findings indicate that our method has a significantly bigger advantage over the other compared ones.

Visualization on clustering tasks

In order to visually demonstrate the clustering performance of HNTriD, we present cluster visualizations of several comparable approaches to assess the data learning capability of HNTriD. In this experiment, we choose the COIL20 dataset as a representative example to conduct comparative tests on clustering tasks. We specifically select 10 categories of data for analysis. The data analysis is shown in a two-dimensional space using t-SNE, and the cluster results are displayed in Figure 5 for visual comparison.

Figure 5 demonstrates that the HNTriD method, when applied to the multiway dataset, is capable of effectively discerning the differences between data samples. HNTriD outperforms other approaches in visually separating sample clusters in the COIL20 dataset, while some methods fail to completely separate samples from other clusters. This strategy enhances the reliability of the clustering data comparison experiment mentioned above and confirms that the inclusion of HNTriD improves the learning capability of multiway data.

Running time comparison

From the previous experimental results (including numerical experiments, statistical significance comparison, and visualization on clustering tasks), the HNTriD model shows better data analysis performance. However, it is important to take into account the time cost when applying mathematical models in real-life situations. This means that if we can improve the efficiency of calculations while preserving the quality of data analysis, the mathematical model will be more effective in practical applications. Based on this background, we figure out the time cost and use Figure 6 to record the running time of clustering tasks for each method on six related datasets. On each dataset, we compare the computational time required by each method to complete the same numerical tests described in Subsection 4.6. Each bar in the Figure 6 represents the total time needed for a method to complete the cluster analysis of a dataset, and different colors represent different algorithms. For example, for each dataset, the time cost of HNTriD is represented in yellow.

We can deduce the following statements from the bar graph: (i) Matrix-based decomposition methods are almost always faster than tensor-based ones. Matrix-based methods have an obvious advantage in terms of running speed for there are few factors that needed to be updated due to their special arithmetic expression. (ii) When compared to general methods, manifold learning ones take longer to complete clustering tasks in most cases. This occurs because manifold learning algorithms require updating more parameters in clustering data. (iii) When compared to matrix-based algorithms, the HNTriD algorithm takes longer to cluster tasks. Given the computational complexity of the algorithm, the experimental results are consistent with our expectations. The increase in computational time is due to the construction of the hypergraph and the depiction of raw data. (iv) Among the tensor-based methods, the HNTriD algorithm's computation speed does not fall behind while maintaining its superior performance.

Conclusions

In this paper, the proposed HNTriD method performs well in multiway data learning because it combines the advantages of hypergraph learning and TriD. By constructing hypergraphs, it can reveal the complex structural information of more complex variables hidden among raw data. When combined with the TriD model, it can retain the multi-linear structure of high-order data while mining the potential information within the data and has strong data clustering abilities. Furthermore, we use the multiplicative update method to optimize the proposed HNTriD model, and experiments show that the new algorithm is convergent. The proposed algorithm is applied to six real-world datasets for clustering analysis, including COIL20, GEORGIA, MNIST, ORL, PIE, and USPS, and the data clustering results are compared to those of several existing algorithms. The experimental results demonstrate that the proposed HNTriD method is efficient and saves time in data analysis. In our current work, our hypergraph does not change once it is generated, which may result in a less-than-ideal hypergraph learned in some data with unexpected noise. The solution to this problem, however, is outside the scope of our current work, and we hope to improve it in the future.

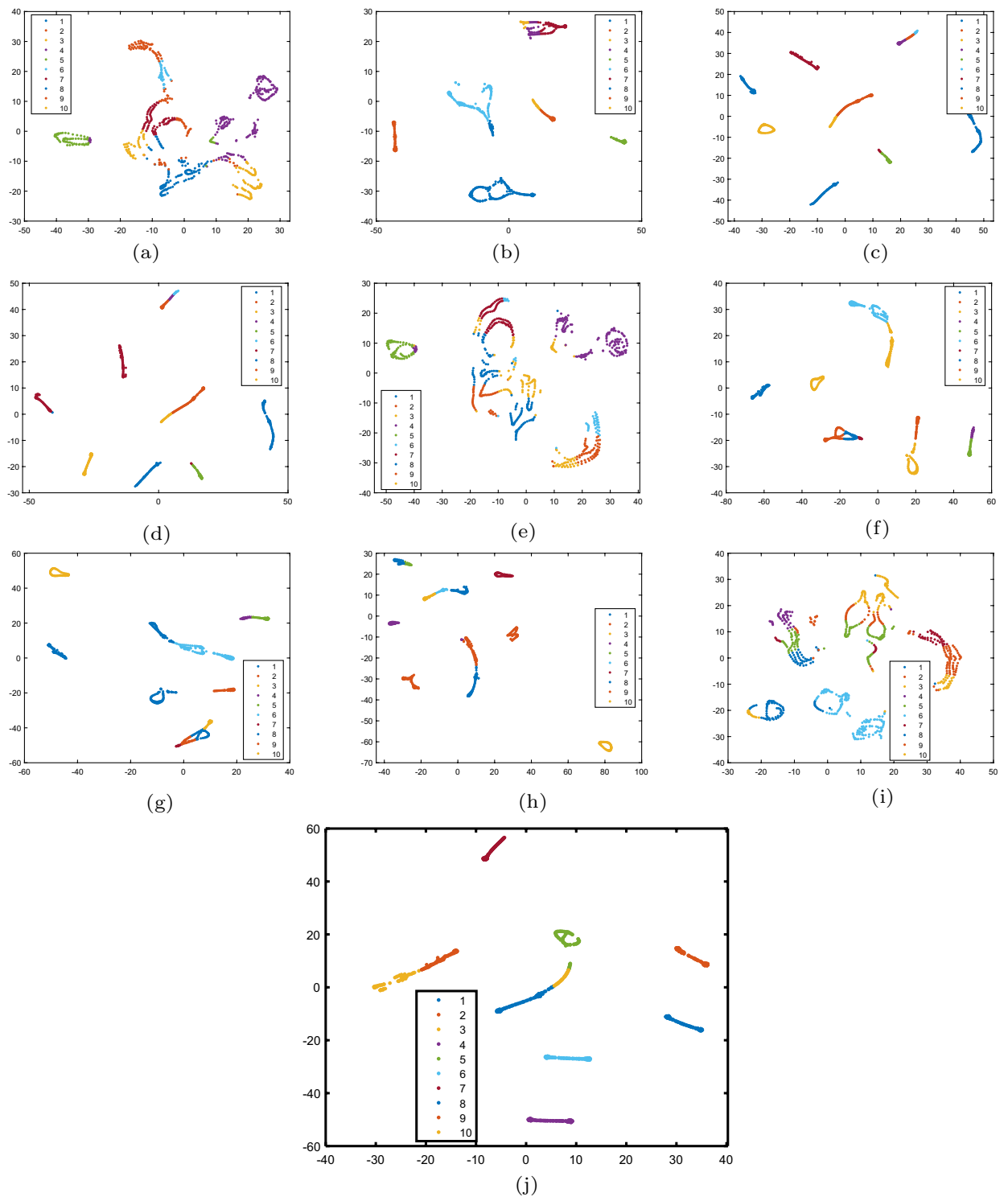


Figure 5. 2-D visualizations of the clustering results of several algorithms using t-SNE on the COIL20 dataset. (a) NMF. (b) GNMF. (c) HNMF. (d) HSNMF. (e) SHNMF. (f) HGNTR. (g) LraHGNT. (h) HyperNTE. (i) TriD. (j) HNTriD.

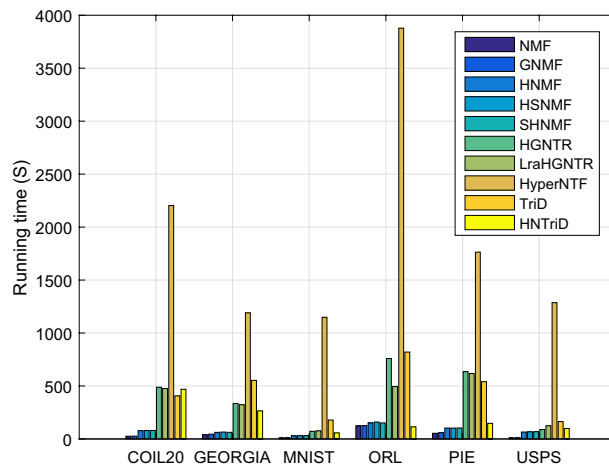


Figure 6. Comparison report of the running time in regard to the different methods on six datasets.

Data availability statements

The open datasets used in this manuscript have been linked in related parts, and the datasets generated during the current study are available from the corresponding author on reasonable request.

Received: 9 January 2024; Accepted: 9 April 2024

Published online: 20 April 2024

References

- Wold, S., Esbensen, K. & Geladi, P. Principal component analysis. *Chemometr. Intell. Lab. 2*(1–3), 37–52 (1987).
- Stewart, G. W. On the early history of the singular value decomposition. *SIAM Rev.* **35**(4), 551–566 (1993).
- Beh, E. J., & Lombardo, R. Multiple and Multiway Correspondence Analysis. *Wiley Interdiscip. Rev. Comput. Stat.* **11** e1464. MR3999531, (2019). <https://doi.org/10.1002/wics.1464>
- Martinez, A. M. & Kak, A. C. PCA versus LDA. *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(2), 228–233 (2001).
- Carroll, J. D. & Chang, J. J. Analysis of individual differences in multidimensional scaling via an n-way generalization of “eckart-young” decomposition. *Psych.* **35**(3), 283–319 (1970).
- Domanov, I. & Lathauwer, L. D. Canonical polyadic decomposition of third-order tensors: Reduction to generalized eigenvalue decomposition. *SIAM J. Matrix Anal. App.* **35**(2), 636–660 (2014).
- Kolda, T. G. & Bader, B. W. Tensor decompositions and applications. *SIAM Rev.* **51**(3), 455–500 (2009).
- Ceulemans, E. & Kiers, H. A. Selecting among three-mode principal component models of different types and complexities: A numerical convex hull based method. *Br. J. Math. Stat. Psychol.* **59**(1), 133–150 (2006).
- Kroonenberg, P. M. Applied Multiway Data Analysis. Wiley Series in Probability and Statistics, Wiley Interscience, Hoboken, NJ. MR2378349 (2008). <https://doi.org/10.1002/9780470238004>
- Kiers, H. A. L. *Three-way methods for the analysis of qualitative and quantitative two-way data* (DSWO Press, Leiden, NL, 1989).
- Kroonenberg, P. M. Multiway extensions of the SVD. *Advanced studies in behaviormetrics and data science T. Imaizumi, A. Nakayama, S. Yokoyama, (eds.)* 141–157 (2020)
- Lombardo, R., Velden, M. & Beh, E. J. Three-way correspondence analysis in R. *R J.* **15**(2), 237–262 (2023).
- Xu, Y. Y. Alternating proximal gradient method for sparse nonnegative Tucker decomposition. *Math. Program. Comput.* **7**, 39–70 (2015).
- Yokota, T., Zdunek, R., Cichocki, A. & Yamashita, Y. Smooth nonnegative matrix and tensor factorizations for robust multi-way data analysis. *Signal Process.* **113**, 234–249 (2015).
- Wu, Q., Zhang, L. Q. & Cichocki, A. Multifactor sparse feature extraction using convolutive nonnegative Tucker decomposition. *Neurocomputing* **129**, 17–24 (2014).
- Tan, H. C., Yang, Z. X., Feng, G., Wang, W. H. & Ran, B. Correlation analysis for tensor-based traffic data imputation method. *Procedia Soc. Behav. Sci.* **96**, 2611–2620 (2013).
- Qi, L. Q., Chen, Y. N., Bakshi, M. & Zhang, X. Z. Triple decomposition and tensor recovery of third order tensors. *SIAM J. Matrix Anal. Appl.* **42**(1), 299–329 (2021).
- Cai, D., He, X. F., Han, J. W. & Huang, T. S. Graph regularized nonnegative matrix factorization for data representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(8), 1548–1560 (2011).
- Chen, K. Y., Che, H. J., Li, X. Q. & Leung, M. F. Graph non-negative matrix factorization with alternative smoothed l0 regularizations. *Neural Comput. Appl.* **35**(14), 9995–10009 (2023).
- Deng, P. et al. Tri-regularized nonnegative matrix tri-factorization for co-clustering. *Knowl-Based Syst.* **226**, 107101 (2021).
- Li, C. L., Che, H. J., Leung, M. F., Liu, C. & Yan, Z. Robust multi-view non-negative matrix factorization with adaptive graph and diversity constraints. *Inf. Sci.* **634**, 587–607 (2023).
- Ly, L. S., Bardou, D., Hu, P., Liu, Y. Q. & Yu, G. H. Graph regularized nonnegative matrix factorization for link prediction in directed temporal networks using pagerank centrality. *Chaos Solitons Fractals* **159**, 112107 (2022).
- Nasiri, E., Berahmand, K. & Li, Y. F. Robust graph regularization nonnegative matrix factorization for link prediction in attributed networks. *Multimed. Tools Appl.* **82**(3), 3745–3768 (2023).
- Wang, Q., He, X., Jiang, X. & Li, X. L. Robust bi-stochastic graph regularized matrix factorization for data clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(1), 390–403 (2020).
- Li, X. T., Ng, M. K., Cong, G., Ye, Y. M. & Wu, Q. Y. MR-NTD: Manifold regularization nonnegative Tucker decomposition for tensor data dimension reduction and representation. *IEEE Trans. Neural Netw. Learn. Syst.* **28**(8), 1787–1800 (2016).

26. Qiu, Y. N., Zhou, G. X., Wang, Y. J., Zhang, Y. & Xie, S. L. A generalized graph regularized non-negative Tucker decomposition framework for tensor data representation. *IEEE T. Cybern.* **52**(1), 594–607 (2020).
27. Liu, Q., Lu, L. & Chen, Z. Non-negative Tucker decomposition with graph regularization and smooth constraint for clustering. *Pattern Recognit.* **148**, 110207 (2024).
28. Wu, F. S., Li, C. Q. & Li, Y. T. Manifold regularization nonnegative triple decomposition of tensor sets for image compression and representation. *J. Optimiz. Theory App.* **192**(3), 979–1000 (2022).
29. Zeng, K., Yu, J., Li, C. H., You, J. & Jin, T. Image clustering by hyper-graph regularized non-negative matrix factorization. *Neuro-computing* **138**, 209–217 (2014).
30. Wang, W. H., Qian, Y. T. & Tang, Y. Y. Hypergraph-regularized sparse NMF for hyperspectral unmixing. *IEEE J. Sel. Topics Appl. Earth Obs. Remote Sens.* **9**(2), 681–694 (2016).
31. Huang, S. *et al.* Improved hypergraph regularized nonnegative matrix factorization with sparse representation. *Pattern Recognit. Lett.* **102**, 8–14 (2018).
32. Yin, W. G., Qu, Y. Z., Ma, Z. M. & Liu, Q. Y. HyperNMF: A hypergraph regularized nonnegative tensor factorization for dimensionality reduction. *Neurocomputing* **512**, 190–202 (2022).
33. Zhao, X. H., Yu, Y. Y., Zhou, G. X., Zhao, Q. B. & Sun, W. J. Fast hypergraph regularized nonnegative tensor ring decomposition based on low-rank approximation. *Appl. Intell.* **52**(15), 17684–17707 (2022).
34. Huang, Z. H., Zhou, G. X., Qiu, Y. N., Yu, Y. Y. & Dai, H. A dynamic hypergraph regularized non-negative Tucker decomposition framework for multiway data analysis. *Int. J. Mach. Learn. Cybern.* **13**(12), 3691–3710 (2022).
35. Kim, Y. D., & Choi, S. Nonnegative Tucker decomposition. In *IEEE Comput. Vis. Pattern Recognit.*, pp. 1–8 (2007). IEEE
36. Gao, Y. *et al.* Hypergraph learning: Methods and practices. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**(5), 2548–2566 (2020).
37. Bretto, A. *Hypergraph Theory* (Springer, New York, 2013).
38. Zhang, Z. H., Bai, L., Liang, Y. H. & Hancock, E. Joint hypergraph learning and sparse regression for feature selection. *Pattern Recognit.* **63**, 291–309 (2017).
39. Lee, D. D. & Seung, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature* **401**(6755), 788–791 (1999).
40. Zhou, D. Y., Huang, J. Y., & Schölkopf, B. Learning with hypergraphs: Clustering, classification, and embedding. *Adv. Neural Inf. Process. Syst.* **19** (2006)
41. Boyd, S., Boyd, S. P. & Vandenberghe, L. *Convex Optimization* (Cambridge Univ. Press, Cambridge, 2004).
42. Lee, D. D. & Seung, H. S. Algorithms for non-negative matrix factorization. *Proc. Adv. Neural Inf. Process. Syst.* **1**, 556–562 (2001).
43. Wang, C. Y. *et al.* Dual hyper-graph regularized supervised NMF for selecting differentially expressed genes and tumor classification. *IEEE ACM Trans. Comput. Biol. Bioinf.* **18**(6), 2375–2383 (2020).
44. Razak, F. A. The derivation of mutual information and covariance function using centered random variables. In *AIP Conference Proceedings*, vol. 1635, pp. 883–889 (2014). AIP
45. Yin, M., Gao, J. B., Xie, S. L. & Guo, Y. Multiview subspace clustering via tensorial t-product representation. *IEEE Trans. Neural Netw. Learn. Syst.* **30**(3), 851–864 (2018).
46. Li, S., Li, W., Lu, H. & Li, Y. Semi-supervised non-negative matrix tri-factorization with adaptive neighbors and block-diagonal learning. *Eng. Appl. Artif. Intell.* **121**, 106043 (2023).
47. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **7**, 1–30 (2006).
48. Huang, D., Wang, C. D. & Lai, J. H. Locally weighted ensemble clustering. *IEEE Trans. Cybern.* **48**(5), 1460–1473 (2017).
49. Zhang, G. Y., Zhou, Y. R., He, X. Y., Wang, C. D. & Huang, D. One-step kernel multi-view subspace clustering. *Knowl. Based Syst.* **189**, 105126 (2020).

Acknowledgements

Fatimah Abdul Razak is supported by Universiti Kebangsaan Malaysia under Grant GUP-2021-046. Qingshui Liao is supported by Scientific Research Foundation of Higher Education Institutions for Young Talents of Department of Education of Guizhou Province under Grant QJJ[2022]129. Qilong Liu is supported by Guizhou Provincial Basic Research Program (Natural Science) under Grant QKHJC-ZK[2023]YB245.

Author contributions

The idea of this article is proposed by Fatimah Abdul Razak, the exact algorithms are provided by Qingshui Liao, the numerical tests are conducted by Qingshui Liao, the English writing is finished by Qingshui Liao and polished by Qilong Liu and Fatimah Abdul Razak.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Q.L. or F.A.R.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International

License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024