AIMS Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 9(8): 21755–21785.
DOI: 10.3934/math.20241058
Received: 23 May 2024
Revised: 28 June 2024
Accepted: 01 July 2024
Published: 08 July 2024

Research article

Non-negative Tucker decomposition with double constraints for multiway dimensionality reduction

Xiang Gao[1], **Linzhang Lu**[1,2,*] **and Qilong Liu**[1]

[1] School of Mathematical Sciences, Guizhou Normal University, Guiyang 550025, China
[2] School of Mathematical Sciences, Xiamen University, Xiamen 361005, China

* **Correspondence:** Email: llz@gznu.edu.cn.

Abstract: Nonnegative Tucker decomposition (NTD) is one of the renowned techniques in feature extraction and representation for nonnegative high-dimensional tensor data. The main focus behind the NTD-like model was how to factorize the data to get ahold of a high quality data representation from multidimensional directions. However, existing NTD-like models do not consider relationship and properties between the factor matrix of columns while preserving the geometric structure of the data space. In this paper, we managed to capture nonlinear local features of data space and further enhance expressiveness of the NTD clustering method by syncretizing organically approximately orthogonal constraint and graph regularized constraint. First, based on the uni-side and bi-side approximate orthogonality, we flexibly proposed two novel approximately orthogonal NTD with graph regularized models, which not only in part make the factor matrix tend to be orthogonality, but also preserve the geometrical information from high-dimensional tensor data. Second, we developed the iterative updating algorithm dependent on the multiplicative update rule to solve the proposed models, and provided its convergence and computational complexity. Finally, we used numerical experimental results to demonstrate the effectiveness, robustness, and efficiency of the proposed new methods on the real-world image datasets.

Keywords: tensor factorization; non-negative Tucker decomposition; graph regularized; approximate orthogonality
Mathematics Subject Classification: 15A69, 62H30, 68U10

1. Introduction

As the speedy expansion of technology is about acquiring and processing information, most factual data exists in tensor form, e.g., electroencephalography signal data, video volume data, hyperspectral image data, color image data, and functional magnetic resonance imaging data.

Enormous and intricate data is ubiquitous in real-world application scenarios; see [1–3]. Typical dimensional reduction methods, such as singular value decomposition [4], local linear embedding [5, 6], vector quantization [7], principal component analysis [8, 9], nonnegative matrix factorization (NMF) [10–12], etc., extract their low-dimensional representation from high-dimensional ones. A commonality of the abovementioned low rank approximation methods is that converting the data matrices (tensors) of samples into a new large matrix may substantially corrupt the architecture of sample data space. To preserve the internal nature of tensor data and study the low-dimensional representation of high-order domain, many tensor decomposition methods [13–15] have been developed. Kim et al. introduced a nonnegative Tucker decomposition (NTD) [16, 17] by combining Tucker decomposition with nonnegativity constraints on the core tensor and factor matrices. By adding nonnegativity constraints, NTD could not only obtain the parts-based representation like NMF, but also improve the uniqueness of Tucker decomposition. Recent efforts have extended NTD to boost calculation efficiency and meet different demands in actual applications by incorporating suitable constraint conditions [18–20] with NTD, including smoothness, graph Laplacian, sparsity, orthogonality, and supervision, just to name a few.

The model that adds effective and feasible constraints to the NTD model is called the NTD-like model. For example, Liu et al. [21] stated a graph regularized L_p smooth NTD method by adding the graph regularization and L_p smooth constraint into NTD to retain smooth and more accurate solutions of the objective function. Qiu et al. [22] proposed a graph Laplacian-regularized NTD (GNTD) method. GNTD extracts the low-dimensional parts-based representation and preserves the geometrical information simultaneously from the high-dimensional tensor data. Subsequently, Qiu et al. [23] developed an alternating proximate gradient descent method to solve the proposed GNTD framework. Chen et al. [24] designed an adaptive graph regularized NTD model, which adaptively learns the optimal graph to capture local manifold information. Li et al. [25] asserted a manifold regularization NTD (MR-NTD) by employing a manifold regularization term for the core tensor constructed in the NTD to preserve geometric information in tensor data. Huang et al. [26] gave a dynamic hypergraph regularized NTD method by incorporating the hypergraph structure and NTD in a unified framework. Jing et al. [27] recommended a label constrained NTD using partial labels to construct a label matrix. Then, they embedded the label term and graph regularization term into NTD for guiding the algorithm to obtain more correct categories in clustering tasks. To make use of the available label information of sample data, Qiu et al. [28] built up a semi-supervised NTD (SNTD) model by propagating the limited label information and learning the nonnegative tensor representation. This part can only cover a small subset of the many important and interesting ideas that have emerged. There are other related studies; see [29–31] for details.

However, the aforementioned NTD-like model does not take into account the orthogonality constraint. In fact, the orthogonality structure of factor matrices makes sense in practical use. In [32], the equivalence of the orthogonal nonnegative matrix factorization (ONMF) problem and K-means clustering has been well discussed. To maintain this characteristic, Pan et al. [33] developed an orthogonal NTD model by considering the orthogonality on each factor matrix. It can get the clustering information from the factor matrices and their joint connection weight from the core tensor. The orthogonal NTD model not only helps to keep the inherent tensor structure but also performs well in data compression. Lately, drawing on the idea of approximate orthogonality from [34], Qiu et al. [35] affirmed a flexible multi-way clustering model called approximately orthogonal NTD for soft clustering. This model provides extra flexibility to handle crossed memberships while making the orthogonality of the factor matrix adjustable. However, these methods that consider orthogonality still have their shortcomings. In clustering problems, it is common to consider the factor matrix of the last direction as an approximation of the original data. Using graph regularization constraints to the factor matrix of the last direction can capture more internal manifold information of the data tensor. They fail to pay attention to the geometrical structure of the data space and the potential connections between samples.

The geometrical structure and resilient orthogonality of the data space are indispensable. To balance the two purposes, in this paper, we propose two approximately orthogonal NTD with graph regularized (AOGNTD) models by jointly blending the graph regularization and approximately orthogonal constraints into the NTD framework. The new model we propose is essentially an NTD-like model. The main contributions of this paper are fourfold:

- Coupling the graph regularized term, the approximately orthogonal term, and the objective function of NTD, we construct a novel tensor-based frame. Based on whether we add the approximately orthogonal constraint to the N-th factor matrix, two models naturally generated.
- By regulating the quality of approximation and picking the appropriate graph for the learning task, the models allow us to detect more complex, latent, and structural correlations among sample data.
- This algorithm is sensitive to the input of parameters, which means this option determines the performance of clustering results. To overcome the issue, we use the grid method to select competent parameters.
- Numerical experiments on frequently adopted datasets are conducted to illustrate the feasibility of the proposed methods for image clustering and classification tasks. Extensive experimental results show that the AOGNTD models could significantly improve the performance.

The rest of this paper is organized as follows. In Section 2, we review the related models. In Section 3, the two AOGNTD methods are proposed. In Section 4, we have discussed its theoretical convergence, provided a solution process and computational complexity. Finally, experiments for clustering tasks are presented in Section 5, and conclusions are drawn in Section 6.

2. The brief review of NTD and GNTD

NTD could be considered as a special case of NMF with sparser and multi-linear basic vectors. Given a nonnegative data tensor

$$\mathcal{X} \in \mathbb{R}_{+}^{I_1 \times I_2 \times \ldots \times I_{N-1} \times I_N},$$

NTD aims at decomposing the nonnegative tensor $\mathcal{X}$ into a nonnegative core tensor

$$\mathcal{G} \in \mathbb{R}_{+}^{J_1 \times J_2 \times \ldots \times J_N}$$

multiplied by N nonnegative factor matrices

$$\mathbf{A}^{(r)} \in \mathbb{R}_{+}^{I_r \times J_r} (r = 1, 2, \ldots, N)$$

along each mode. To achieve this goal, NTD minimizes the sum of squared residues between the data tensor $\mathcal{X}$ and the multi-linear product of core tensor $\mathcal{G}$ and factor matrices $\mathbf{A}^{(r)}$, which can be formulated as

$$\begin{aligned} \min_{\mathcal{G},\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(N)}} O_{NTD} &= \frac{1}{2}\left\|\mathcal{X} - \mathcal{G}\times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)}\right\|^2, \\ s.t.\mathcal{G} &\geqslant 0,\ \ \mathbf{A}^{(r)} \geqslant 0, \quad r = 1,2,\ldots,N, \end{aligned} \tag{2.1}$$

where in the following, the operator $\times_r$ is referred to as the r-mode product. The r-mode product of a tensor

$$\mathcal{Y} \in \mathbb{R}^{J_1\times J_2 \times \ldots \times J_N}$$

and a matrix $\mathbf{U} \in \mathbb{R}^{I_r \times J_r}$, denoted by $\mathcal{Y}\times_r \mathbf{U}$, is

$$(\mathcal{Y}\times_r \mathbf{U})_{j_1\ldots j_{r-1} i_r j_{r+1}\ldots j_N} = \sum_{j_r=1}^{J_r} y_{j_1\ldots j_{r-1} j_r j_{r+1}\ldots j_N} u_{i_r j_r}.$$

Equation (2.1) can be represented as a matrix form:

$$\begin{aligned} \min_{\mathbf{G}_{(N)},\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(N)}} O_{NTD} &= \frac{1}{2}\left\|\mathbf{X}_{(N)} - \mathbf{A}^{(N)}\mathbf{G}_{(N)}(\otimes_{p\neq N}\mathbf{A}^{(p)})^\top\right\|^2, \\ s.t.\mathbf{G}_{(N)} &\geqslant 0,\ \ \mathbf{A}^{(r)} \geqslant 0, \quad r = 1,2,\ldots,N, \end{aligned}$$

where

$$\mathbf{A}^{(N)} \in \mathbb{R}_+^{I_N\times J_N}, \quad \otimes_{p\neq N}\mathbf{A}^{(p)} = \mathbf{A}^{(N-1)} \otimes \mathbf{A}^{(N-2)} \otimes \cdots \otimes \mathbf{A}^{(1)},$$

$\otimes$ denotes the Kronecker product, and

$$\mathbf{X}_{(N)} \in \mathbb{R}_+^{I_N\times I_1 I_2\cdots I_{N-1}} \quad \text{and} \quad \mathbf{G}_{(N)} \in \mathbb{R}_+^{J_N\times J_1 J_2\cdots J_{N-1}}$$

are the mode-N unfolding matrices of data tensor $\mathcal{X}$ and core tensor $\mathcal{G}$, respectively. Therefore, the NTD regard as the NMF with the encoding matrix

$$\mathbf{A}^{(N)} \in \mathbb{R}_+^{I_N\times J_N}$$

and the basis matrix

$$\mathbf{G}_{(N)}(\otimes_{p\neq N}\mathbf{A}^{(p)})^\top,$$

where I_N and J_N can be regarded as the number of all samples and the dimension of low-dimensional representation of $\mathcal{X}$.

The local geometric structure can be effectively modeled through a nearest neighbor graph on data points, which originates from spectral graph theory and manifold learning. Graphs using the broad range of p nearest neighbors are generated and employed. The most commonly weighted matrices are heat kernel and 0-1 weighting. The heat kernel is

$$\mathbf{W}_{ij} = \begin{cases} e^{-\frac{\left\|\mathbf{y}_i - \mathbf{y}_j\right\|^2}{\sigma^2}}, & \mathbf{y}_j \in \mathcal{N}(\mathbf{y}_i)\ or\ \mathbf{y}_i \in \mathcal{N}(\mathbf{y}_j), \\ 0, & \text{otherwise}, \end{cases}$$

where $\mathcal{N}(\mathbf{y}_i)$ is composed of the p-nearest neighbors of sample $\mathbf{y}_i$, and σ is the Gaussian kernel parameter to control the values of similarity. In our experiment, it was set to 1. Alternatively, 0-1 weighting gave a binarization of weight definition as

$$\mathbf{W}_{ij} = \begin{cases} 1, & \mathbf{y}_j \in \mathcal{N}(\mathbf{y}_i) \ or \ \mathbf{y}_i \in \mathcal{N}(\mathbf{y}_j), \\ 0, & \text{otherwise.} \end{cases}$$

This 0-1 weight is the cosine weight method, which is used in the subsequent image experiments.

To measure the dissimilarity of data points $\mathbf{z}_i, \mathbf{z}_j$ in the low-dimensional representation $\mathbf{A}^{(N)}$,

$$d = (\mathbf{z}_i, \mathbf{z}_j) = \|\mathbf{z}_i - \mathbf{z}_j\|^2$$

is under consideration. The definition of graph Laplacian is

$$\begin{aligned} \frac{1}{2}\sum_{i,j=1}^{n} \|\mathbf{z}_i - \mathbf{z}_j\|^2 \mathbf{W}_{ij} &= \sum_{i=1}^{n} \mathbf{z}_i^{\top}\mathbf{z}_i \sum_{j=1}^{n} \mathbf{W}_{ij} - \sum_{i,j=1}^{n} \mathbf{z}_i^{\top}\mathbf{z}_j \mathbf{W}_{ij} \\ &= \sum_{i=1}^{n} \mathbf{z}_i^{\top}\mathbf{z}_i \mathbf{D}_{ii} - \sum_{i,j=1}^{n} \mathbf{z}_i^{\top}\mathbf{z}_j \mathbf{W}_{ij} \\ &= \mathrm{Tr}(\mathbf{A}^{(N)^T}\mathbf{D}\mathbf{A}^{(N)}) - \mathrm{Tr}(\mathbf{A}^{(N)^T}\mathbf{W}\mathbf{A}^{(N)}) \\ &= \mathrm{Tr}(\mathbf{A}^{(N)^T}\mathbf{L}\mathbf{A}^{(N)}), \end{aligned}$$

where $\mathbf{W}$ is a weight matrix, $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is called the graph Laplacian, and

$$\mathbf{D}_{ii} = \sum_{j} \mathbf{W}_{ij}$$

is a diagonal matrix whose elements are column sums of $\mathbf{W}$. Minimizing the above formula, we hope if $\mathbf{y}_i$ and $\mathbf{y}_j$ are close, $\mathbf{z}_i$ and $\mathbf{z}_j$ are consistent with this trend. If data $\mathbf{z}_i$ and $\mathbf{z}_j$ are similar, the value of $\mathbf{W}_{ij}$ is relatively large. The operation of trace effectively characterizes the smoothness of low-rank representations.

GNTD is obtained by minimizing the following objective function:

$$\begin{aligned} \min_{\mathbf{G}_{(N)},\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(N)}} O_{GNTD} &= \frac{1}{2}\left\|\mathcal{X} - \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \ldots \times_N \mathbf{A}^{(N)}\right\|^2 + \frac{\lambda}{2}\mathrm{Tr}(\mathbf{A}^{(N)^T}\mathbf{L}\mathbf{A}^{(N)}), \\ s.t. \mathcal{G} \geqslant 0, \quad \mathbf{A}^{(r)} &\geqslant 0, \quad r = 1, 2, \ldots, N, \end{aligned}$$

where $\mathrm{Tr}(\cdot)$ represents the trace of a matrix, $\mathbf{L}$ is called the graph Laplacian matrix, and λ is a nonnegative parameter for balancing the importance of a graph regularization term and reconstruction error term. It is worthwhile to integrate the graph regularization into mode-N low-dimensional representation $\mathbf{A}^{(N)}$.

3. Approximately orthogonal NTD with graph regularized

In this section, an approximately orthogonal NTD with graph regularized model is proposed, the specific updating rules are introduced, the convergence of stated algorithms are proved, and their computational complexity is investigated.

3.1. Approximately orthogonal term

As an orthogonal version of NMF, an orthogonal nonnegative matrix factorization has been proven to be closely related to k-means clustering. For a given nonnegative data matrix $\mathbf{X} \in \mathbb{R}_{+}^{M\times N}$, the ONMF seeks nonnegative factors $\mathbf{U} \in \mathbb{R}_{+}^{M\times R}$ and $\mathbf{V} \in \mathbb{R}_{+}^{N\times R}$ through the following optimization:

$$\begin{aligned} \min_{\mathbf{U}\geqslant 0,\mathbf{V}\geqslant 0} O_{ONMF} &= \frac{1}{2}\left\|\mathbf{X} - \mathbf{U}\mathbf{V}^{\top}\right\|^{2}, \\ s.t.\mathbf{V}^{\top}\mathbf{V} &= \mathbf{I}, \end{aligned}$$

where

$$\mathbf{V} = [v_1, v_2, \ldots, v_r], \ \ (r = 1, 2, \ldots, R).$$

The rigid orthogonality constraint of $\mathbf{V}$ actually consists of two parts: the unit-norm constraints

$$v_r^T v_r = 1$$

and the orthogonality (zero) constraints

$$v_r^T v_j = 0, \ \ r \neq j,$$

where v_i denotes the i-th column of the encoding matrix $\mathbf{V}$. It is reasonable to generalize the orthogonality of NMF to the orthogonality of each directional matrix in NTD:

$$\begin{aligned} \min_{\mathcal{G},\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(N)}} O_{ONTD} &= \frac{1}{2}\left\|\mathcal{X} - \mathcal{G}\times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)}\right\|^{2}, \\ s.t.\mathcal{G}, \mathbf{A}^{(r)} &\geqslant 0, \quad \mathbf{A}^{(r)\top}\mathbf{A}^{(r)} = \mathbf{I}, \quad r = 1, 2, \ldots, N. \end{aligned} \tag{3.1}$$

The orthogonality of the direct absorption about the factor matrix makes model solving more complex, so we cast around for its approximate orthogonal form. The unit-norm constraints are not obligatory. In the paper, we attempted not to put to use this condition. Hence, we only need to focus on the orthogonality constraints that explain the independent relationship between columns. Under the non-negativity constraints, we have

$$\mathbf{a}_i^{(r)\top}\mathbf{a}_j^{(r)} \geqslant 0$$

for any i and j. The form of trace

$$\sum_{i=1}^{R}\sum_{j=1,j\neq i}^{R} \mathbf{a}_i^{(r)\top}\mathbf{a}_j^{(r)} = \mathrm{Tr}(\mathbf{A}^{(r)\top}\mathbf{Q}\mathbf{A}^{(r)})$$

is a convenient mathematical expression of a partially and approximately orthogonality of matrix, where

$$\mathbf{Q} = \mathbf{1}_R\mathbf{1}_R^T - \mathbf{I},$$

$\mathbf{1}_R$ is an all-one R-by-1 column vector, $\mathbf{I}$ is an identity matrix, and R will be assigned different values depending on the corresponding issue. The larger the value of this item, the greater the degree of orthogonality of the matrix.

3.2. The proposed AOGNTD model

Notice that

$$\begin{aligned}&\left\|\mathcal{X}-\mathcal{G}\times_1 \mathbf{A}^{(1)}\times_2 \mathbf{A}^{(2)}\cdots\times_N \mathbf{A}^{(N)}\right\|^2\\ =&\left\|\mathbf{X}_{(N)}-\mathbf{A}^{(N)}\mathbf{G}_{(N)}(\otimes_{p\neq N}\mathbf{A}^{(p)})^\top\right\|_F^2\\ =&\left\|\mathbf{X}_{(N)}-\mathbf{U}\mathbf{S}\mathbf{V}^\top\right\|_F^2\end{aligned}$$

holds, where

$$\mathbf{U}=\mathbf{A}^{(N)},\quad \mathbf{S}=\mathbf{G}_{(N)},\quad \mathbf{V}=\otimes_{p\neq N}\mathbf{A}^{(p)}.$$

We see that the nonnegative 3-factorization:

$$\mathbf{X}_{(N)}\approx\mathbf{U}\mathbf{S}\mathbf{V}^\top$$

gives a good framework for clustering the rows and columns of $\mathbf{X}_{(N)}$. We consider uni-side and bi-side approximately orthogonal cases in the tri-factorization, respectively, so two models are spontaneously born:

- Uni-sided approximately orthogonality:
  $\mathbf{V}^\top\mathbf{V}=\mathbf{I}$ is weakened to $\sum_{i,j,i\neq j}\mathbf{V}_i^T\mathbf{V}_j$ ($\mathbf{V}_i$ and $\mathbf{V}_j$ stand for the different column of V), or simply

  $$\sum_{r=1}^{N-1}\mathrm{Tr}(\mathbf{A}^{(r)^T}\mathbf{Q}\mathbf{A}^{(r)})$$

  for convenience of calculation. The original expression contains the mass of Kronecker product operations, which is too expensive.

- Bi-sided approximately orthogonality:
  $\mathbf{U}^\top\mathbf{U}=\mathbf{I}$ and $\mathbf{V}^\top\mathbf{V}=\mathbf{I}$ are weakened to $\sum_{i,j,i\neq j}\mathbf{U}_i^T\mathbf{U}_j$ and $\sum_{i,j,i\neq j}\mathbf{V}_i^T\mathbf{V}_j$, or simply

  $$\sum_{r=1}^{N}\mathrm{Tr}(\mathbf{A}^{(r)^T}\mathbf{Q}\mathbf{A}^{(r)})$$

  for convenience of calculation.

It is easy to detect that uni-sides situation is a special case of bi-sides situation. Out of such inspiration, we can embed the approximately orthogonal regularization into regular GNTD to develop AOGNTD (or bi-AOGNTD):

$$\begin{aligned}\min_{\mathcal{G},\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(N)}} Obj=&\frac{1}{2}\left\|\mathcal{X}-\mathcal{G}\times_1 \mathbf{A}^{(1)}\times_2 \mathbf{A}^{(2)}\cdots\times_N \mathbf{A}^{(N)}\right\|^2\\ &+\frac{\mu_r}{2}\sum_{r=1}^{N}\mathrm{Tr}(\mathbf{A}^{(r)^T}\mathbf{Q}\mathbf{A}^{(r)})+\frac{\lambda}{2}\mathrm{Tr}(\mathbf{A}^{(N)^T}\mathbf{L}\mathbf{A}^{(N)}),\\ s.t.\mathcal{G}\geqslant&0,\ \ \mathbf{A}^{(r)}\geqslant 0,\quad r=1,2,\ldots,N,\end{aligned}\tag{3.2}$$

where $\mathbf{L}$ is graph Laplacian matrix,

$$\mathbf{Q} = \mathbf{1}_R\mathbf{1}_R^T - \mathbf{I},$$

and μ_r, λ are positive integers. When $\mu_N = 0$ is held, the bi-AOGNTD model could degenerate into uni-AOGNTD. The objective function is:

$$\min_{\mathcal{G},\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(N)}} \frac{1}{2}\left\|\mathcal{X} - \mathcal{G}\times_1 \mathbf{A}^{(1)}\times_2 \mathbf{A}^{(2)}\cdots\times_N \mathbf{A}^{(N)}\right\|^2 + \frac{\mu_r}{2}\sum_{r=1}^{N-1}\mathrm{Tr}(\mathbf{A}^{(r)^T}\mathbf{Q}\mathbf{A}^{(r)}) + \frac{\lambda}{2}\mathrm{Tr}(\mathbf{A}^{(N)^T}\mathbf{L}\mathbf{A}^{(N)}),$$
$$s.t.\mathcal{G}\geqslant 0,\ \ \mathbf{A}^{(r)}\geqslant 0,\quad r = 1,2,\ldots,N-1.$$

In the AOGNTD model,

- $\|\mathcal{X} - \mathcal{G}\times_1 \mathbf{A}^{(1)}\times_2 \mathbf{A}^{(2)}\ldots\times_N \mathbf{A}^{(N)}\|^2$ is the main part, which is the key to feature extraction.
- $\mathrm{Tr}(\mathbf{A}^{(r)^T}\mathbf{Q}\mathbf{A}^{(r)}), r = 1,\ldots,N$ is used to ensure that the first $n-1$ factor matrices are partially orthogonal, which makes global structure more accurate.
- $\mathrm{Tr}(\mathbf{A}^{(N)^T}\mathbf{L}\mathbf{A}^{(N)})$ is the basis for graph optimization, which is achieved by the Laplacian regularizer to conserve the smooth structure of low-rank expression.

We expect that the proposed AOGNTD model can capture more global structure and local manifold information.

4. Optimization algorithm

Solving the global optimal solution of (3.2) is tough, so we adopt the block coordinate descent framework that updates the core tensor or one factor matrix each time while fixing others. This update rule is also known as the multiplicative update rule. It is a validated compromise plan between speed and convenient of implementation.

We use the Lagrange multiplier method and apply the mode-n unfolding form. The Lagrange function is

$$\begin{aligned}\mathcal{L} =&\frac{1}{2}\left\|\mathbf{X}_{(n)} - \mathbf{A}^{(n)}\mathbf{G}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)})^\top\right\|_F^2 + \frac{\mu_r}{2}\sum_{r=1}^{N}\mathrm{Tr}(\mathbf{A}^{(r)^T}\mathbf{Q}\mathbf{A}^{(r)})\\ &+\frac{\lambda}{2}\mathrm{Tr}(\mathbf{A}^{(N)^T}\mathbf{L}\mathbf{A}^{(N)}) + \mathrm{Tr}(\Phi_n\mathbf{G}_{(n)}^T) + \sum_{r=1}^{N}\mathrm{Tr}(\Psi_r\mathbf{A}^{(r)^\top}),\end{aligned}\tag{4.1}$$

where Φ_n and Ψ_r are the Lagrange multipliers matrices of $\mathbf{G}_{(n)}$ and $\mathbf{A}^{(r)}$, respectively. The function (4.1) can be rewritten as

$$\begin{aligned}\mathcal{L} =&\frac{1}{2}\mathrm{Tr}(\mathbf{X}_{(n)}\mathbf{X}_{(n)}^\top) - \mathrm{Tr}(\mathbf{X}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^\top\mathbf{A}^{(n)^\top}) + \frac{1}{2}\mathrm{Tr}(\mathbf{A}^{(n)}\mathbf{G}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)^\top}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^T\mathbf{A}^{(n)^\top})\\ &+\frac{\mu_r}{2}\sum_{r=1}^{N}\mathrm{Tr}(\mathbf{A}^{(r)^T}\mathbf{Q}\mathbf{A}^{(r)}) + \frac{\lambda}{2}\mathrm{Tr}(\mathbf{A}^{(N)^T}\mathbf{L}\mathbf{A}^{(N)}) + \mathrm{Tr}(\Phi_n\mathbf{G}_{(n)}^\top) + \sum_{r=1}^{N}\mathrm{Tr}(\Psi_r\mathbf{A}^{(r)^\top}).\end{aligned}$$

The entire updating rule is divided into tripartite, containing the solutions of factor matrices $\mathbf{A}^{(n)}$, $(n = 1,2,\ldots,N-1)$, the solutions of factor matrices $\mathbf{A}^{(N)}$, and the solutions of core tensor $\mathcal{G}$.

4.1. Updating factor matrix

To update $\mathbf{A}^{(r)}$, $r = 1, \ldots, N$, we fix $\mathcal{G}$ and the other factor matrix.

The partial derivatives of $\mathcal{L}$ in (4.1) with respect to $\mathbf{A}^{(n)}$, $(n = 1, 2, \ldots, N-1)$ are

$$\frac{\partial \mathcal{L}}{\partial \mathbf{A}^{(n)}} = -\mathbf{X}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^{\top} + \mathbf{A}^{(n)}\mathbf{G}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)\top}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^{\top} + \mu_n\mathbf{Q}\mathbf{A}^{(n)} + \Psi_n.$$

By using the Karush-Kuhn-Tucker (KKT) condition, i.e.,

$$\partial\mathcal{L}/\partial\mathbf{A}^{(n)} = 0 \quad \text{and} \quad \mathbf{A}^{(n)} \odot \Psi_n = 0,$$

where in the following, $\odot$ denotes the Hadamard product. According to $\partial\mathcal{L}/\partial\mathbf{A}^{(n)} = 0$, we obtain

$$\Psi_n = \mathbf{X}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^{\top} - \mathbf{A}^{(n)}\mathbf{G}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)\top}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^{\top} - \mu_n\mathbf{Q}\mathbf{A}^{(n)}.$$

By calculating

$$\mathbf{A}^{(n)} \odot \Psi_n = \mathbf{A}^{(n)} \odot (\mathbf{X}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^{\top}) - \mathbf{A}^{(n)} \odot (\mathbf{A}^{(n)}\mathbf{G}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)\top}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^{\top} - \mu_n\mathbf{Q}\mathbf{A}^{(n)}),$$

which together with

$$\mathbf{A}^{(n)} \odot \Psi_n = 0$$

yields the following updating rule for $\mathbf{A}^{(n)}(n = 1, 2, \ldots, N-1)$:

$$\mathbf{A}_{ij}^{(n)} \leftarrow \mathbf{A}_{ij}^{(n)} \frac{P_+[[\mathbf{X}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^{\top} + \mu_n\mathbf{I}\mathbf{A}^{(n)}]_{ij}]}{[\mathbf{A}^{(n)}\mathbf{G}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)\top}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^{\top} + \mu_n\mathbf{1}_R\mathbf{1}_R^T\mathbf{A}^{(n)}]_{ij}}, \tag{4.2}$$

where

$$P_+[\eta] = \max(0, \eta).$$

The partial derivative of $\mathcal{L}$ in (4.1) with respect to $\mathbf{A}^{(N)}$ is

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{A}^{(N)}} =& - \mathbf{X}_{(N)}(\otimes_{p\neq N}\mathbf{A}^{(p)})\mathbf{G}_{(N)}^{\top} + \mathbf{A}^{(N)}\mathbf{G}_{(N)}(\otimes_{p\neq N}\mathbf{A}^{(p)\top}\mathbf{A}^{(p)})\mathbf{G}_{(N)}^{\top} \\ &+ \mu_N\mathbf{Q}\mathbf{A}^{(N)} + \lambda\mathbf{L}\mathbf{A}^{(N)} + \Psi_N.\end{aligned}$$

Similarly, we consider the KKT condition

$$\partial\mathcal{L}/\partial\mathbf{A}^{(N)} = 0 \quad \text{and} \quad \mathbf{A}^{(N)} \odot \Psi_N = 0.$$

As a result, we obtain the following updating rule for $\mathbf{A}^{(N)}$:

$$\mathbf{A}_{ij}^{(N)} \leftarrow \mathbf{A}_{ij}^{(N)} \frac{P_+[[\mathbf{X}_{(N)}(\otimes_{p\neq N}\mathbf{A}^{(p)})\mathbf{G}_{(N)}^{\top} + \mu_N\mathbf{I}\mathbf{A}^{(N)} + \lambda\mathbf{W}\mathbf{A}^{(N)}]_{ij}]}{[\mathbf{A}^{(N)}\mathbf{G}_{(N)}(\otimes_{p\neq N}\mathbf{A}^{(p)\top}\mathbf{A}^{(p)})\mathbf{G}_{(N)}^{\top} + \mu_N\mathbf{1}_R\mathbf{1}_R^T\mathbf{A}^{(N)} + \lambda\mathbf{D}\mathbf{A}^{(N)}]_{ij}}. \tag{4.3}$$

When only considering the unilateral information of the triple decomposition, the iterative format will undergo slight changes and the following updating rule for $\mathbf{A}^{(N)}$ is

$$\mathbf{A}_{ij}^{(N)} \leftarrow \mathbf{A}_{ij}^{(N)} \frac{P_+[[\mathbf{X}_{(N)}(\otimes_{p\neq N}\mathbf{A}^{(p)})\mathbf{G}_{(N)}^{\top} + \lambda\mathbf{W}\mathbf{A}^{(N)}]_{ij}]}{[\mathbf{A}^{(N)}\mathbf{G}_{(N)}(\otimes_{p\neq N}\mathbf{A}^{(p)\top}\mathbf{A}^{(p)})\mathbf{G}_{(N)}^{\top} + \lambda\mathbf{D}\mathbf{A}^{(N)}]_{ij}}. \tag{4.4}$$

4.2. Updating core tensor

To update $\mathcal{G}$, we fix $\mathbf{A}^{(r)}$, $r = 1, \ldots, N$.

The objective function in (4.1) about $\mathcal{G}$ can be changed into

$$\mathcal{L} = \frac{1}{2} \left\| vec(\mathcal{X}) - \mathbf{F} vec(\mathcal{G}) \right\|_2^2 + vec(\mathcal{G})^T vec(\Phi), \tag{4.5}$$

where

$$\mathbf{F} = \mathbf{A}^{(N)} \otimes \mathbf{A}^{(N-1)} \otimes \ldots \otimes \mathbf{A}^{(1)} \in \mathbb{R}^{I_1 I_2 \ldots I_N \times J_1 J_2 \ldots J_N},$$

$vec(\mathcal{X})$ expands the tensor $\mathcal{X}$ into a 1-dimension vector and $vec(\Phi)$ represents the Lagrange multiplier of $vec(\mathcal{G})$. The partial derivative of $\mathcal{L}$ in (4.5) with respect to $vec(\mathcal{G})$ is

$$\frac{\partial \mathcal{L}}{\partial vec(\mathcal{G})} = \mathbf{F}^\top \mathbf{F} vec(\mathcal{G}) - \mathbf{F}^\top vec(\mathcal{X}) + vec(\Phi).$$

By applying

$$\partial \mathcal{L} / \partial vec(\mathcal{G}) = 0 \quad \text{and} \quad (vec(\mathcal{G}))_i (vec(\Phi))_i = 0,$$

we obtain the following updating rule:

$$(vec(\mathcal{G}))_i \leftarrow (vec(\mathcal{G}))_i \frac{P_+[(\mathbf{F}^\top vec(\mathcal{X}))_i]}{(\mathbf{F}^\top \mathbf{F} vec(\mathcal{G}))_i}. \tag{4.6}$$

At this point, the optimization problem has been solved. According to the above iteration rules, we summarize the process of AOGNTD method, as shown in Algorithm 1.

Algorithm 1. Algorithm of the AOGNTD method.

```
Require: Data matrix $\mathcal{X}$, cluster number $k$; parameter $\lambda$, $\mu$.
Ensure: Core tensor $\mathcal{G}$, nonnegative factor matrices $\mathbf{A}^{(r)}, r = 1, 2, \ldots, N$.
Initialize $\mathbf{A}^{(r)}$ as random matrices and $\mathcal{G}$ as an arbitrary positive tensor.
Calculate the weight matrix $\mathbf{W}$.
repeat
    Update $\mathbf{A}^{(n)}$, by (4.2), where $n = 1, 2, \ldots, N-1$.
    Update $\mathbf{A}^{(N)}$ by (4.3) or (4.4).
    Update $\mathcal{G}$ by (4.6).
until the algorithm convergence condition is satisfied.
```

4.3. Theoretical investigation

In [11], we are well aware that if $G(u, u^{'})$ is an auxiliary function for $F(u)$, then $F(u)$ is nonincreasing under the updating rule

$$u^{t+1} = \underset{u}{\operatorname{argmin}} G(u, u^t). \tag{4.7}$$

The equality

$$F(u^{t+1}) = F(u^t)$$

holds only if u^t is a local minimum of $G(u, u^t)$. Each subproblem has an optimal solution, which is the fundamental basis and theorem support about our algorithm. The construction and design of auxiliary functions are of utmost importance in this part. Before expounding the local convergence theorem, we offer three lemmas about diverse auxiliary functions.

For any element $a_{ij}^{(n)}$ in $\mathbf{A}^{(n)}(n = 1, 2, \ldots, N-1)$, let $F_{ij}(a_{ij}^{(n)})$ denote the part of the objective function in (3.2) relevant to $a_{ij}^{(n)}$. The iterative process is element-wise, so it is necessary to imply that each $F_{ij}(a_{ij}^{(n)})$ is nonincreasing under the iteration rule. The first-order derivative of the $F_{ij}(a_{ij}^{(n)})$ with respect to $a_{ij}^{(n)}$ is

$$F_{ij}^{'}(a_{ij}^{(n)}) = [-\mathbf{X}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^{\top} + \mathbf{A}^{(n)}\mathbf{G}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)^{\top}}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^{\top} + \mu_n\mathbf{1}_R\mathbf{1}_R^T\mathbf{A}^{(n)}]_{ij},$$

and

$$F_{ij}^{''}(a_{ij}^{(n)^t}) = [\mathbf{G}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)^{\top}}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^{\top} + \mu_n\mathbf{1}_R\mathbf{1}_R^T]_{jj}$$

is the second order derivative of the $F_{ij}(a_{ij}^{(n)})$ relevant to $a_{ij}^{(n)^t}$.

Lemma 1. *The function*

$$\begin{aligned} G(a_{ij}^{(n)}, a_{ij}^{(n)^t}) =& F_{ij}(a_{ij}^{(n)^t}) + F_{ij}^{'}(a_{ij}^{(n)^t})(a_{ij}^{(n)} - a_{ij}^{(n)^t}) \\ &+ \frac{1}{2}\frac{[\mathbf{A}^{(n)^t}\mathbf{G}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)^{\top}}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^{\top} + \mu_n\mathbf{1}_R\mathbf{1}_R^T\mathbf{A}^{(n)^t}]_{ij}}{a_{ij}^{(n)^t}} \times (a_{ij}^{(n)} - a_{ij}^{(n)^t})^2 \end{aligned} \tag{4.8}$$

is an auxiliary function for $F_{ij}(a_{ij}^{(n)})$, with the matrix

$$\mathbf{A}^{(n)^t} = (a_{ij}^{(n)^t}).$$

Proof. Basically, $G(u, u^{'})$ is an auxiliary function for $F(u)$ if the conditions

$$G(u, u^{'}) \geq F(u) \quad \text{and} \quad G(u, u) = F(u)$$

are satisfied.

$$G(a_{ij}^{(n)^t}, a_{ij}^{(n)^t}) = F_{ij}(a_{ij}^{(n)^t})$$

clearly holds, so we only need to testify that

$$G(a_{ij}^{(n)}, a_{ij}^{(n)^t}) \geq F_{ij}(a_{ij}^{(n)}).$$

The Taylor expansion of $F_{ij}(a_{ij}^{(n)})$ at $a_{ij}^{(n)^t}$ is

$$F_{ij}(a_{ij}^{(n)}) = F_{ij}(a_{ij}^{(n)^t}) + F_{ij}^{'}(a_{ij}^{(n)^t})(a_{ij}^{(n)} - a_{ij}^{(n)^t}) + \frac{1}{2}F_{ij}^{''}(a_{ij}^{(n)^t})(a_{ij}^{(n)} - a_{ij}^{(n)^t})^2. \tag{4.9}$$

Comparing (4.8) with (4.9), we can see that

$$G(a_{ij}^{(n)}, a_{ij}^{(n)^t}) \geq F_{ij}(a_{ij}^{(n)})$$

is equivalent to

$$\frac{[\mathbf{A}^{(n)^t}\mathbf{G}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)^{\top}}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^{\top}]_{ij}}{a_{ij}^{(n)^t}} \geq [\mathbf{G}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)^{\top}}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^{\top}]_{jj}.$$

We have

$$[\mathbf{A}^{(n)^t}\mathbf{G}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)^\top}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^\top]_{ij} = \sum_{l=1}^{J_n} a_{il}^{(n)^t}[\mathbf{G}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)^\top}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^\top]_{lj}$$
$$\geq a_{ij}^{(n)^t}[\mathbf{G}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)^\top}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^\top]_{jj},$$

thus, the inequality

$$G(a_{ij}^{(n)}, a_{ij}^{(n)^t}) \geq F_{ij}(a_{ij}^{(n)})$$

holds. □

Let $F_{ij}(a_{ij}^{(N)})$ denote the part of the objective function in (3.2) relevant to $a_{ij}^{(N)}$in $\mathbf{A}^{(N)}$.

Lemma 2. *The function*

$$\begin{aligned} G(a_{ij}^{(N)}, a_{ij}^{(N)^t}) =& F_{ij}(a_{ij}^{(N)^t}) + F_{ij}^{'}(a_{ij}^{(N)^t})(a_{ij}^{(N)} - a_{ij}^{(N)^t}) \\ &+ \frac{1}{2}\frac{[\mathbf{A}^{(N)^t}\mathbf{G}_{(N)}(\otimes_{p\neq N}\mathbf{A}^{(p)^\top}\mathbf{A}^{(p)})\mathbf{G}_{(N)}^\top + \lambda\mathbf{D}\mathbf{A}^{(N)^t} + \mu_N\mathbf{1}_R\mathbf{1}_R^T\mathbf{A}^{(N)^t}]_{ij}}{a_{ij}^{(N)^t}} \times (a_{ij}^{(N)} - a_{ij}^{(N)^t})^2 \end{aligned} \tag{4.10}$$

is an auxiliary function for $F_{ij}(a_{ij}^{(N)})$.

Lemma 3. *Let g_i denote the element of $vec(\mathcal{G})$ and $F_i(g_i)$ denote the part of the objective function in (4.1) relevant to g_i. The function*

$$G(g_i, g_i^t) = F_i(g_i^t) + F_i^{'}(g_i^t)(g_i - g_i^t) + \frac{(\mathbf{F}^\top\mathbf{F}vec(\mathcal{G}^t))_i}{g_i^t}(g_i - g_i^t)^2 \tag{4.11}$$

is an auxiliary function for $F_i(g_i)$.

Due to the similarity between the proofs of Lemmas 1–3, they are omitted here.

Theorem 4. *The objective function in (4.1) is nonincreasing under the updating rules in (4.2), (4.3), and (4.6). The objective function is invariant under these updates if, and only if,* $\mathbf{A}^{(r)}$*, $r = 1, 2, \ldots, N$, $\mathcal{G}$ are at a stationary point.*

Proof. Replacing $G(u, u^{'})$ in (4.7) by (4.8), we obtain

$$\frac{\partial G(a_{ij}^{(n)}, a_{ij}^{(n)^t})}{\partial a_{ij}^{(n)}} = F_{ij}^{'}(a_{ij}^{(n)^t}) + \frac{[\mathbf{A}^{(n)^t}\mathbf{G}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)^\top}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^\top + \mu_n\mathbf{1}_R\mathbf{1}_R^T\mathbf{A}^{(n)^t}]_{ij}}{a_{ij}^{(n)^t}}(a_{ij}^{(n)} - a_{ij}^{(n)^t}) = 0,$$

which yields

$$a_{ij}^{(n)^{t+1}} = a_{ij}^{(n)^t}\frac{[\mathbf{X}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^\top + \mu_n\mathbf{I}\mathbf{A}^{(n)^t}]_{ij}}{[\mathbf{A}^{(n)^t}\mathbf{G}_{(n)}(\otimes_{p\neq n}\mathbf{A}^{(p)^\top}\mathbf{A}^{(p)})\mathbf{G}_{(n)}^\top + \mu_n\mathbf{1}_R\mathbf{1}_R^T\mathbf{A}^{(n)^t}]_{ij}}.$$

According to Lemma 1, $F_{ij}(a_{ij}^{(n)})$ is nonincreasing under the updating rules (4.2) for $\mathbf{A}^{(n)}$. Then, putting $G(a_{ij}^{(N)}, a_{ij}^{(N)^t})$ of (4.10) into (4.7), we can obtain

$$a_{ij}^{(N)^{t+1}} = a_{ij}^{(N)^t}\frac{[\mathbf{X}_{(N)}(\otimes_{p\neq N}\mathbf{A}^{(p)})\mathbf{G}_{(N)}^\top + \lambda\mathbf{W}\mathbf{A}^{(N)^t} + \mu_N\mathbf{I}\mathbf{A}^{(N)^t}]_{ij}}{[\mathbf{A}^{(N)^t}\mathbf{G}_{(N)}(\otimes_{p\neq N}\mathbf{A}^{(p)^\top}\mathbf{A}^{(p)})\mathbf{G}_{(N)}^\top + \lambda\mathbf{D}\mathbf{A}^{(N)^t} + \mu_N\mathbf{1}_R\mathbf{1}_R^T\mathbf{A}^{(N)^t}]_{ij}}.$$

From Lemma 2, $F_{ij}(a_{ij}^{(N)})$ is nonincreasing under the updating rules (4.3) for $\mathbf{A}^{(N)}$. Finally, putting $G(g_i, g_i^t)$ of (4.11) into (4.7),

$$g_i^{t+1} = g_i^t \frac{(F^\top vec(\mathcal{X}))_i}{(F^\top F vec(\mathcal{G}^t))_i}.$$

In accordance with Lemma 3, $F_i(g_i)$ is nonincreasing under the updating rule (4.6).

The proof of Theorem 4 is completed. □

4.4. Computational complexity analysis

In this subsection, we discuss the computational cost of our proposed algorithms in comparison to other methods. The common way to express the complexity of one algorithm is using big O notation, but it is not precise enough to differentiate the complexities. Thus, we additionally count the arithmetic operations for each algorithm, including fladd (a floating-point addition), flmlt (a floating-point multiplication) and fldiv (a floating-point division). Specifically, we provide the computational complexity analysis of uni-AOGNTD and bi-AOGNTD models. There are many Kronecker product operations in our update rules, which needs high time cost. So, we will replace the Kronecker product with tensor operation during the calculation process.

To facilitate the estimation of computational complexity, we consider the case where $r = 3$ in the models. Taking uni-AOGNTD as an example, the update rules, respectively, are

$$\mathbf{A}_{ij}^{(1)} \leftarrow \mathbf{A}_{ij}^{(1)} \frac{[(\mathcal{X} \times_2 \mathbf{A}^{(2)^\top} \times_3 \mathbf{A}^{(3)^\top})_{(1)} \mathbf{G}_{(1)}^\top + \mu \mathbf{I} \mathbf{A}^{(1)}]_{ij}}{[\mathbf{A}^{(1)} \mathbf{G}_{(1)} (\mathcal{X} \times_2 \mathbf{A}^{(2)^\top} \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)^\top} \mathbf{A}^{(3)})_{(1)} \mathbf{G}_{(1)}^\top + \mu \mathbf{1}_R \mathbf{1}_R^T \mathbf{A}^{(1)}]_{ij}},$$

$$\mathbf{A}_{ij}^{(2)} \leftarrow \mathbf{A}_{ij}^{(2)} \frac{[(\mathcal{X} \times_1 \mathbf{A}^{(1)^\top} \times_3 \mathbf{A}^{(3)^\top})_{(2)} \mathbf{G}_{(2)}^\top + \mu \mathbf{I} \mathbf{A}^{(2)}]_{ij}}{[\mathbf{A}^{(2)} \mathbf{G}_{(2)} (\mathcal{X} \times_1 \mathbf{A}^{(1)^\top} \mathbf{A}^{(1)} \times_3 \mathbf{A}^{(3)^\top} \mathbf{A}^{(3)})_{(2)} \mathbf{G}_{(2)}^\top + \mu \mathbf{1}_R \mathbf{1}_R^T \mathbf{A}^{(2)}]_{ij}},$$

$$\mathbf{A}_{ij}^{(3)} \leftarrow \mathbf{A}_{ij}^{(3)} \frac{[(\mathcal{X} \times_1 \mathbf{A}^{(1)^\top} \times_2 \mathbf{A}^{(2)^\top})_{(3)} \mathbf{G}_{(3)}^\top + \lambda \mathbf{W} \mathbf{A}^{(3)}]_{ij}}{[\mathbf{A}^{(3)} \mathbf{G}_{(3)} (\mathcal{X} \times_1 \mathbf{A}^{(1)^\top} \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)^\top} \mathbf{A}^{(2)})_{(3)} \mathbf{G}_{(3)}^\top + \lambda \mathbf{D} \mathbf{A}^{(3)}]_{ij}}$$

and

$$\mathcal{G}_{ijk} \leftarrow \mathcal{G}_{ijk} \frac{[\mathcal{X} \times_1 \mathbf{A}^{(1)^\top} \times_2 \mathbf{A}^{(2)^\top} \times_3 \mathbf{A}^{(3)^\top}]_{ijk}}{[\mathcal{G} \times_1 \mathbf{A}^{(1)^\top} \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)^\top} \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)^\top} \mathbf{A}^{(3)}]_{ijk}}.$$

where

$$\mathcal{X}_{(1)}(\mathbf{A}^{(2)} \otimes \mathbf{A}^{(3)}) = \mathcal{X} \times_2 \mathbf{A}^{(2)^\top} \times_3 \mathbf{A}^{(3)^\top}$$

and so on.

The frequently used expressions and the specific calculations for the two methods are provided in Table 1. By calculating the computational complexity of the four formats and considering the fact $I_i \gg J_i$, we can gain the complexity of the proposed uni-AOGNTD as $O(I_1 I_2 I_3 J)$, in which

$$J = 3(J_1 + J_2 + J_3).$$

Then, we also can acquire the complexity of the bi-AOGNTD method as $O(I_1 I_2 I_3 J)$. According to the analysis of other compared methods in corresponding literature, Table 2 generalizes the complexity of compared algorithms. The new algorithm will inevitably increase a certain amount of computation, but it can be ignored in terms of overall computational complexity.

Table 1. Computational operation counts.

	Fladd	Flmlt	Fldiv
$\mathcal{Y} \times_r \mathbf{U}$	$I_r J_1 J_2 \dots J_N$	$I_r J_1 J_2 \dots J_N$	–
$\mathcal{X} \times_2 \mathbf{A}_2^{\mathbf{T}} \times_3 \mathbf{A}_3^{\mathbf{T}}$	$I_1 I_2 I_3 J_2 + I_1 I_3 J_2 J_3$	$I_1 I_2 I_3 J_2 + I_1 I_3 J_2 J_3$	–
$\mathcal{G} \times_2 \mathbf{A}_2^T \mathbf{A}_2 \times_3 \mathbf{A}_3^T \mathbf{A}_3$	$J_1 J_2^2 J_3 + J_1 J_2 J_3^2 + I_2 J_2^2 + I_3 J_3^2$	$J_1 J_2^2 J_3 + J_1 J_2 J_3^2 + I_2 J_2^2 + I_3 J_3^2$	–
uni-AOGNTD			
$\mathbf{A}^{(1)}$	$I_1 I_2 I_3 J_2 + I_1 I_3 J_2 J_3 + J_1 J_2^2 J_3 + J_1 J_2 J_3^2$ $+ I_2 J_2^2 + I_3 J_3^2 + 3 I_1 J_1 J_2 J_3 + 2 I_1 J_1 (I_1 + 2)$	$I_1 I_2 I_3 J_2 + I_1 I_3 J_2 J_3 + J_1 J_2^2 J_3 + J_1 J_2 J_3^2 + I_2 J_2^2$ $+ I_3 J_3^2 + 3 I_1 J_1 J_2 J_3 + 2 I_1 J_1 (I_1 + 1) + I_1 J_1$	$I_1 J_1$
$\mathbf{A}^{(2)}$	$I_1 I_2 I_3 J_1 + I_2 I_3 J_1 J_3 + J_1^2 J_2 J_3 + J_1 J_2 J_3^2$ $+ I_1 J_1^2 + I_3 J_3^2 + 3 I_2 J_1 J_2 J_3 + 2 I_2 J_2 (I_2 + 2)$	$I_1 I_2 I_3 J_1 + I_2 I_3 J_1 J_3 + J_1^2 J_2 J_3 + J_1 J_2 J_3^2 + I_1 J_1^2$ $+ I_3 J_3^2 + 3 I_2 J_1 J_2 J_3 + 2 I_2 J_2 (I_2 + 1) + I_2 J_2$	$I_2 J_2$
$\mathbf{A}^{(3)}$	$I_1 I_2 I_3 J_1 + I_2 I_3 J_1 J_2 + J_1^2 J_2 J_3 + J_1 J_2^2 J_3$ $+ I_1 J_1^2 + I_2 J_2^2 + 3 I_3 J_1 J_2 J_3 + 2 I_3 J_3 (I_3 + 2)$	$I_1 I_2 I_3 J_1 + I_2 I_3 J_1 J_2 + J_1^2 J_2 J_3 + J_1 J_2^2 J_3 + I_1 J_1^2$ $+ I_2 J_2^2 + 3 I_3 J_1 J_2 J_3 + 2 I_3 J_3 (I_3 + 2) + I_3 J_3$	$I_3 J_3$
$\mathcal{G}$	$I_1 I_2 I_3 J_1 + I_2 I_3 J_1 J_2 + J_1^2 J_2 J_3 + J_1 J_2^2 J_3$ $+ J_1 J_2 J_3^2 + I_1 J_1^2 + I_2 J_2^2 + I_3 J_3^2 + I_3 J_1 J_2 J_3$	$I_1 I_2 I_3 J_1 + I_2 I_3 J_1 J_2 + J_1^2 J_2 J_3 + J_1 J_2^2 J_3 + I_1 J_1^2$ $+ I_2 J_2^2 + 3 I_3 J_1 J_2 J_3 + J_1 J_2 J_3$	$J_1 J_2 J_3$
bi-AOGNTD			
$\mathbf{A}^{(1)}$	$I_1 I_2 I_3 J_2 + I_1 I_3 J_2 J_3 + J_1 J_2^2 J_3 + J_1 J_2 J_3^2$ $+ I_2 J_2^2 + I_3 J_3^2 + 3 I_1 J_1 J_2 J_3 + 2 I_1 J_1 (I_1 + 2)$	$I_1 I_2 I_3 J_2 + I_1 I_3 J_2 J_3 + J_1 J_2^2 J_3 + J_1 J_2 J_3^2 + I_2 J_2^2$ $+ I_3 J_3^2 + 3 I_1 J_1 J_2 J_3 + 2 I_1 J_1 (I_1 + 1) + I_1 J_1$	$I_1 J_1$
$\mathbf{A}^{(2)}$	$I_1 I_2 I_3 J_1 + I_2 I_3 J_1 J_3 + J_1^2 J_2 J_3 + J_1 J_2 J_3^2$ $+ I_1 J_1^2 + I_3 J_3^2 + 3 I_2 J_1 J_2 J_3 + 2 I_2 J_2 (I_2 + 2)$	$I_1 I_2 I_3 J_1 + I_2 I_3 J_1 J_3 + J_1^2 J_2 J_3 + J_1 J_2 J_3^2 + I_1 J_1^2$ $+ I_3 J_3^2 + 3 I_2 J_1 J_2 J_3 + 2 I_2 J_2 (I_2 + 1) + I_2 J_2$	$I_2 J_2$
$\mathbf{A}^{(3)}$	$I_1 I_2 I_3 J_1 + I_2 I_3 J_1 J_2 + J_1^2 J_2 J_3 + J_1 J_2^2 J_3$ $+ I_1 J_1^2 + I_2 J_2^2 + 3 I_3 J_1 J_2 J_3 + 4 I_3 J_3 (I_3 + 2)$	$I_1 I_2 I_3 J_1 + I_2 I_3 J_1 J_2 + J_1^2 J_2 J_3 + J_1 J_2^2 J_3 + I_1 J_1^2$ $+ I_2 J_2^2 + 3 I_3 J_1 J_2 J_3 + 4 I_3 J_3 (I_1 + 1) + I_3 J_3$	$I_3 J_3$
$\mathcal{G}$	$I_1 I_2 I_3 J_1 + I_2 I_3 J_1 J_2 + J_1^2 J_2 J_3 + J_1 J_2^2 J_3$ $+ J_1 J_2 J_3^2 + I_1 J_1^2 + I_2 J_2^2 + I_3 J_3^2 + I_3 J_1 J_2 J_3$	$I_1 I_2 I_3 J_1 + I_2 I_3 J_1 J_2 + J_1^2 J_2 J_3 + J_1 J_2^2 J_3 + I_1 J_1^2$ $+ I_2 J_2^2 + 3 I_3 J_1 J_2 J_3 + J_1 J_2 J_3$	$J_1 J_2 J_3$

Table 2. Computational operation counts.

Model	Objective function	Overall	Supplement
NMF	$\min_{\mathbf{U},\mathbf{V},} \frac{1}{2} \left\Vert \mathbf{X} - \mathbf{U}\mathbf{V} \right\Vert^2$ s.t. $\mathbf{U}, \mathbf{V} \geqslant 0$	$\mathcal{O}(MNK)$	
NTD	$\min_{\mathcal{G},\mathbf{A}^{(1)},\dots,\mathbf{A}^{(N)}} \frac{1}{2} \left\Vert \mathcal{X} - \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} \right\Vert^2$ s.t. $\mathcal{G} \geqslant 0, \mathbf{A}^{(r)} \geqslant 0$	$\mathcal{O}(I_1 I_2 I_3 J)$	$J = 3(J_1 + J_2 + J_3)$
GNTD	$\min_{\mathcal{G},\mathbf{A}^{(1)},\dots,\mathbf{A}^{(N)}} \frac{1}{2} \left\Vert \mathcal{X} - \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} \right\Vert^2$ $+ \frac{\lambda}{2} \mathrm{Tr}(\mathbf{A}^{(N)^T} \mathbf{L} \mathbf{A}^{(N)})$ s.t. $\mathcal{G} \geqslant 0, \mathbf{A}^{(r)} \geqslant 0$	$\mathcal{O}(I_1 I_2 I_3 J)$	$J = 3(J_1 + J_2 + J_3)$
uni-AOGNTD	$\min_{\mathcal{G},\mathbf{A}^{(1)},\dots,\mathbf{A}^{(N)}} \frac{1}{2} \left\Vert \mathcal{X} - \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} \right\Vert^2$ $+ \frac{\mu_r}{2} \sum_{r=1}^{N-1} \mathrm{Tr}(\mathbf{A}^{(r)^T} \mathbf{Q} \mathbf{A}^{(r)}) + \frac{\lambda}{2} \mathrm{Tr}(\mathbf{A}^{(N)^T} \mathbf{L} \mathbf{A}^{(N)})$ s.t. $\mathcal{G} \geqslant 0, \mathbf{A}^{(r)} \geqslant 0$	$\mathcal{O}(I_1 I_2 I_3 J)$	$J = 3(J_1 + J_2 + J_3)$
bi-AOGNTD	$\min_{\mathcal{G},\mathbf{A}^{(1)},\dots,\mathbf{A}^{(N)}} \frac{1}{2} \left\Vert \mathcal{X} - \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \cdots \times_N \mathbf{A}^{(N)} \right\Vert^2$ $+ \frac{\mu_r}{2} \sum_{r=1}^{N} \mathrm{Tr}(\mathbf{A}^{(r)^T} \mathbf{Q} \mathbf{A}^{(r)}) + \frac{\lambda}{2} \mathrm{Tr}(\mathbf{A}^{(N)^T} \mathbf{L} \mathbf{A}^{(N)})$ s.t. $\mathcal{G} \geqslant 0, \mathbf{A}^{(r)} \geqslant 0$	$\mathcal{O}(I_1 I_2 I_3 J)$	$J = 3(J_1 + J_2 + J_3)$

5. Experiments

In this Section, according to [36], two indicators, accuracy (AC) and normalized mutual information (NMI), are used to evaluate the clustering performance. The AC is defined as

$$\mathrm{AC} = \frac{\delta(\mathrm{k_i}, \mathrm{map}(\mathrm{c_i}))}{\mathrm{n}},$$

where c_i denotes the label from clustering method, k_i is the true label of object x_i, $\delta(x, y)$ is the indicator function, n is the total number of objects, and $map(c_i)$ is the mapping function. The NMI is defined as

$$\mathrm{NMI} = \frac{\sum_{\mathrm{u=1}}^{\mathrm{c}} \sum_{\mathrm{v=1}}^{\mathrm{k}} \log(\frac{\mathrm{nn_{uv}}}{\mathrm{n_u n_v'}})}{\sqrt{(\sum_{u=1}^{c} n_u \log(\frac{n_u}{n}))(\sum_{v=1}^{k} n_v' \log(\frac{n_v'}{n}))}},$$

where n_{uv} is the number of cluster, and n_i and n_i' are the data points in the true label cluster and the clustering label cluster, respectively.

All the experiments are run with MATLAB on a PC equipped with Intel i5-10210U and 8GB of RAM.

5.1. Compared methods

To evaluate the effectiveness of the proposed AOGNTD methods, we compare the clustering performance of our methods with the other six state of the art and previously mentioned methods, which are k-means [36], NMF [11], graph regularized NMF (GNMF) [36], graph dual regularized NMF (GDNMF) [37, 38] , NTD [17], and GNTD [22, 23]. In order to distinguish DNMF of the original article from DNMF with the label information [27], we have written DNMF [37, 38] as GDNMF.

5.2. Datasets description

Three databases are used to conduct the experiments in Figure 1, and the details of public datasets are stated as follows (the COIL20 dataset means the Columbia University image library dataset, the ORL dataset means the Oxford-rotating dataset):

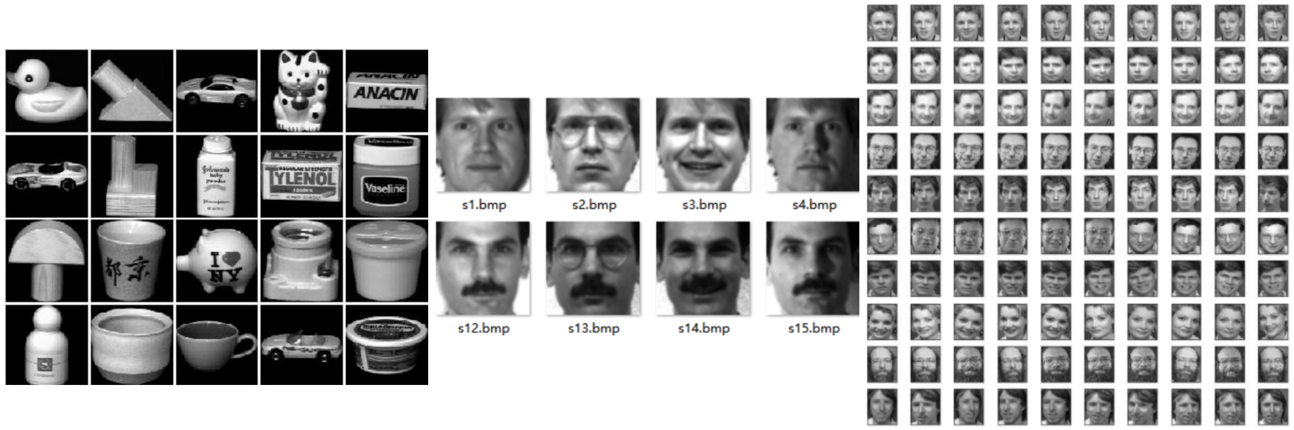

Figure 1. COIL20 (left), Yale (middle), and ORL dataset (right).

5.2.1. COIL20 dataset

The COIL20 dataset collects 1440 grayscale images of 20 subjects. The objects were placed on a motorized turntable, which was rotated through 360 degrees to vary object pose with respect to a fixed camera. Images of the objects were taken at pose intervals of 5 degrees, which corresponds to 72 images per object. In our experiments, each image was resized to 32×32 pixels and all images are stacked into a tensor $\mathcal{X} \in \mathbb{R}^{32\times32\times1440}$.

5.2.2. Yale dataset

The Yale faces dataset contains 165 grayscale images in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration. Each image was resized into 32×32 pixels. In the experiments, all images form a tensor $\mathcal{X} \in \mathbb{R}^{32\times32\times165}$.

5.2.3. ORL dataset

The ORL dataset collects 400 grayscale 112×92 faces images which consist of 40 different subjects with 10 distinct images. For some subjects, the images were taken at different lighting, times, and facial expressions. For each image, we resized it to be 32×32 pixels in our experiment. These images construct a tensor of $\mathcal{X} \in \mathbb{R}^{32\times32\times400}$.

5.3. Parameter selection

Identifying and adjusting the parameter combination of the algorithm has always been a striking matter in machine learning. The size of core tensor plays a prominent role in our algorithms. For NTD-based methods, the size of the core tensor is set to $\{10 \times 10 \times s, 20 \times 20 \times s, 30 \times 30 \times s\}$ in three datasets, where s is positive integer.

There are multiple regularization parameters λ and $\mu_r, r = 1, 2, \ldots, N$ balancing the effects of the reconstruction term, graph Laplacian term, and approximately orthogonal term in our model. To reduce parameter adjustment pressure, we study the influence on the clustering performance when the parameters λ and $\mu_r = \mu$ and the ratio of sampling vary. Since it is infeasible to consider the changes of these parameters disposable, we use the grid control method to determine the relatively optimal parameters of the process.

To begin, determining the numerical range of two parameters is $[5e - 5, 1e + 4]$, then select t values within this range, and finally select t^2 results to achieve the best clustering effect. For the COIL20 dataset, we set $\{1e - 1, 1, 1e + 1, 1e + 2\}$ as selections of the parameters λ and μ in uni-AOGNTD format, where $t = 4$, and choose the best among 16 values. It is crucial to weigh the two measurement criteria reasonably when comparing results.

In principle, the numerical value of AC (accuracy) is the main factor, supplemented by NMI. More details can be found in Figures 2–7. Figures 2–7 shows the specific parameter selection, and the optimal parameter setting is circled in black. The symbol dataset-k, say, COIL20-4, represents the categories extracted from a certain dataset. During the procedure of parameter option, we found strong graph regularity and weak orthogonality of the AOGNTD models.

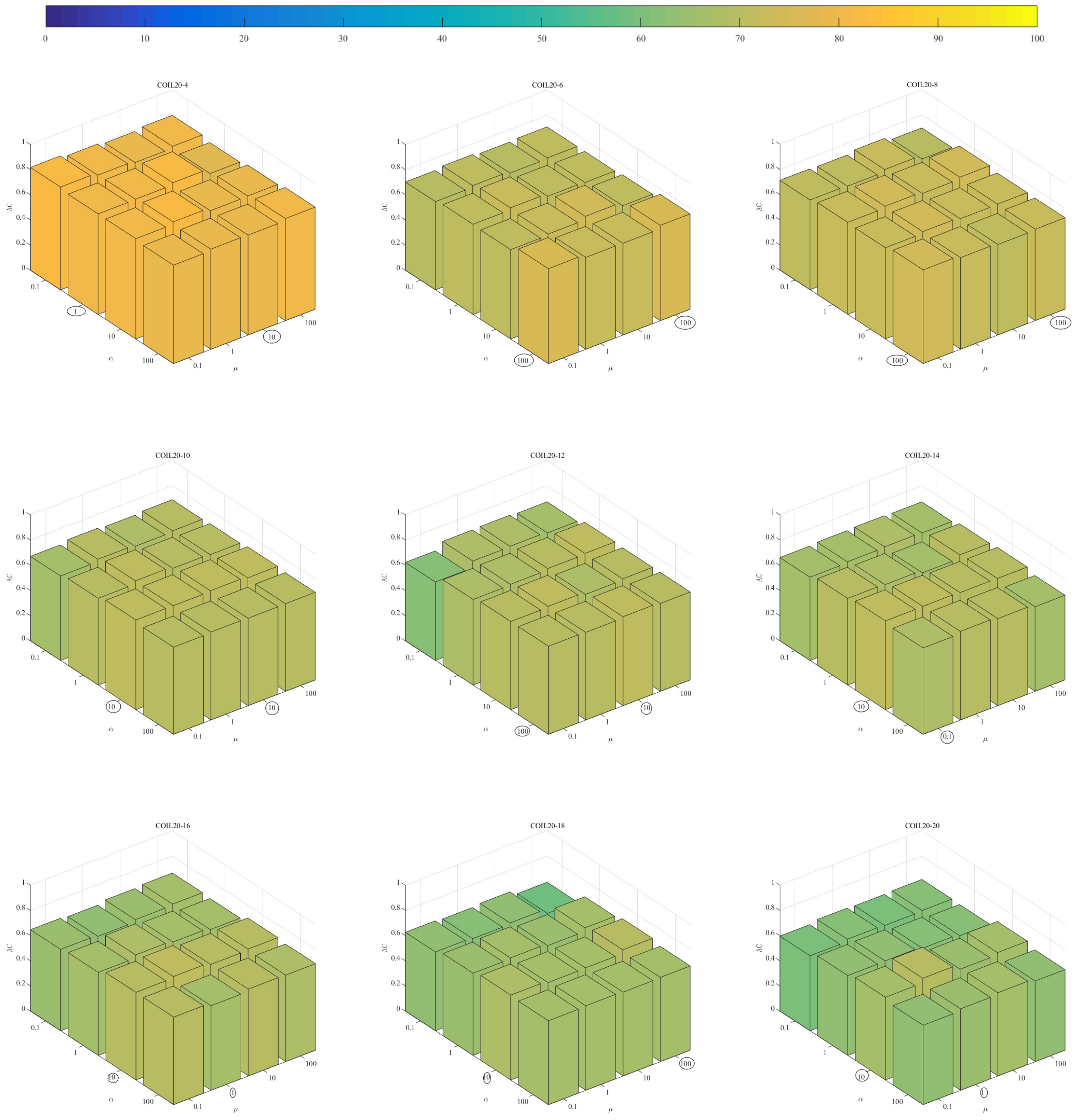

Figure 2. The clustering performance of the uni-AOGNTD method on the COIL20 datasets.

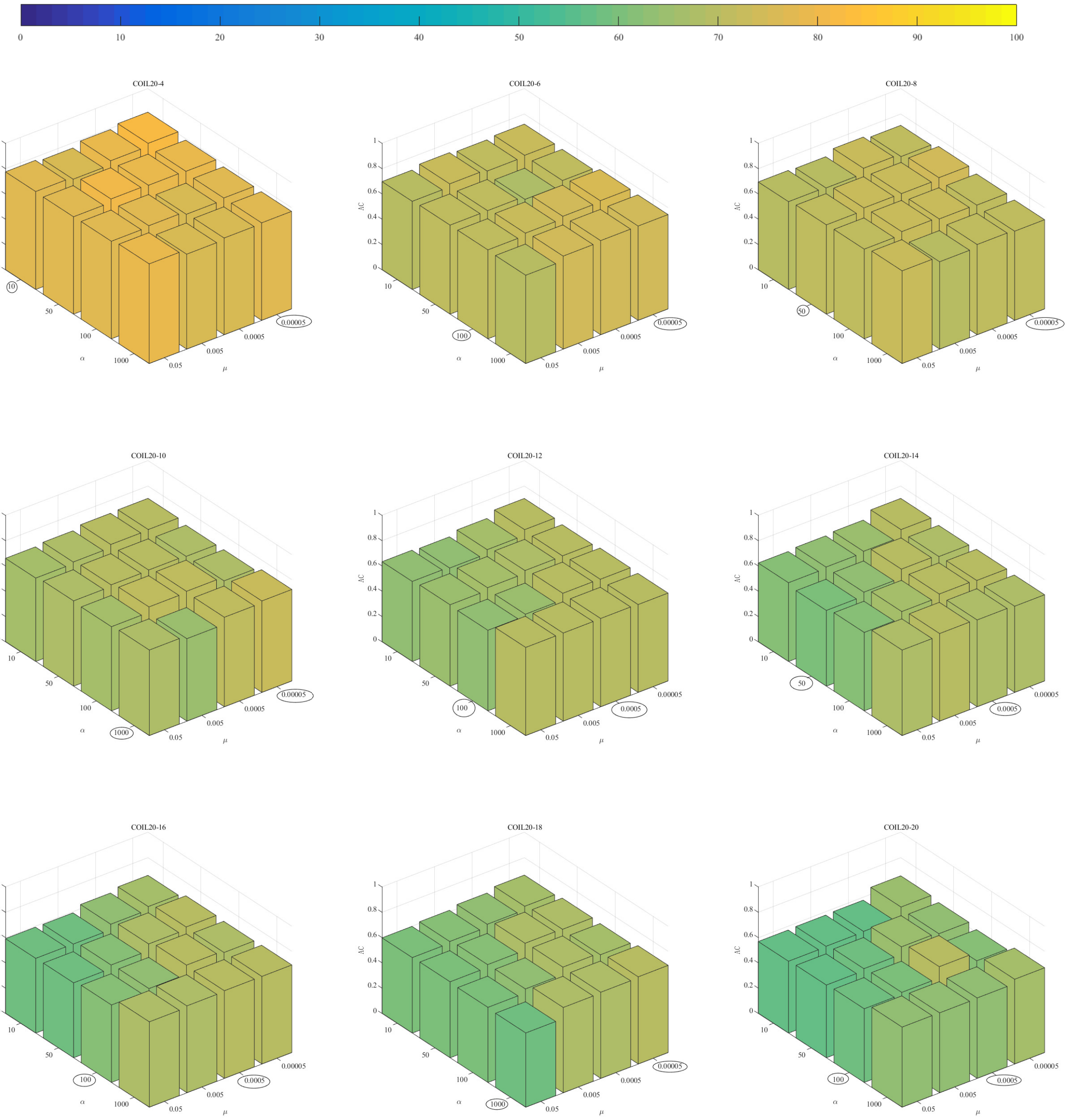

Figure 3. The clustering performance of the bi-AOGNTD method on the COIL20 datasets.

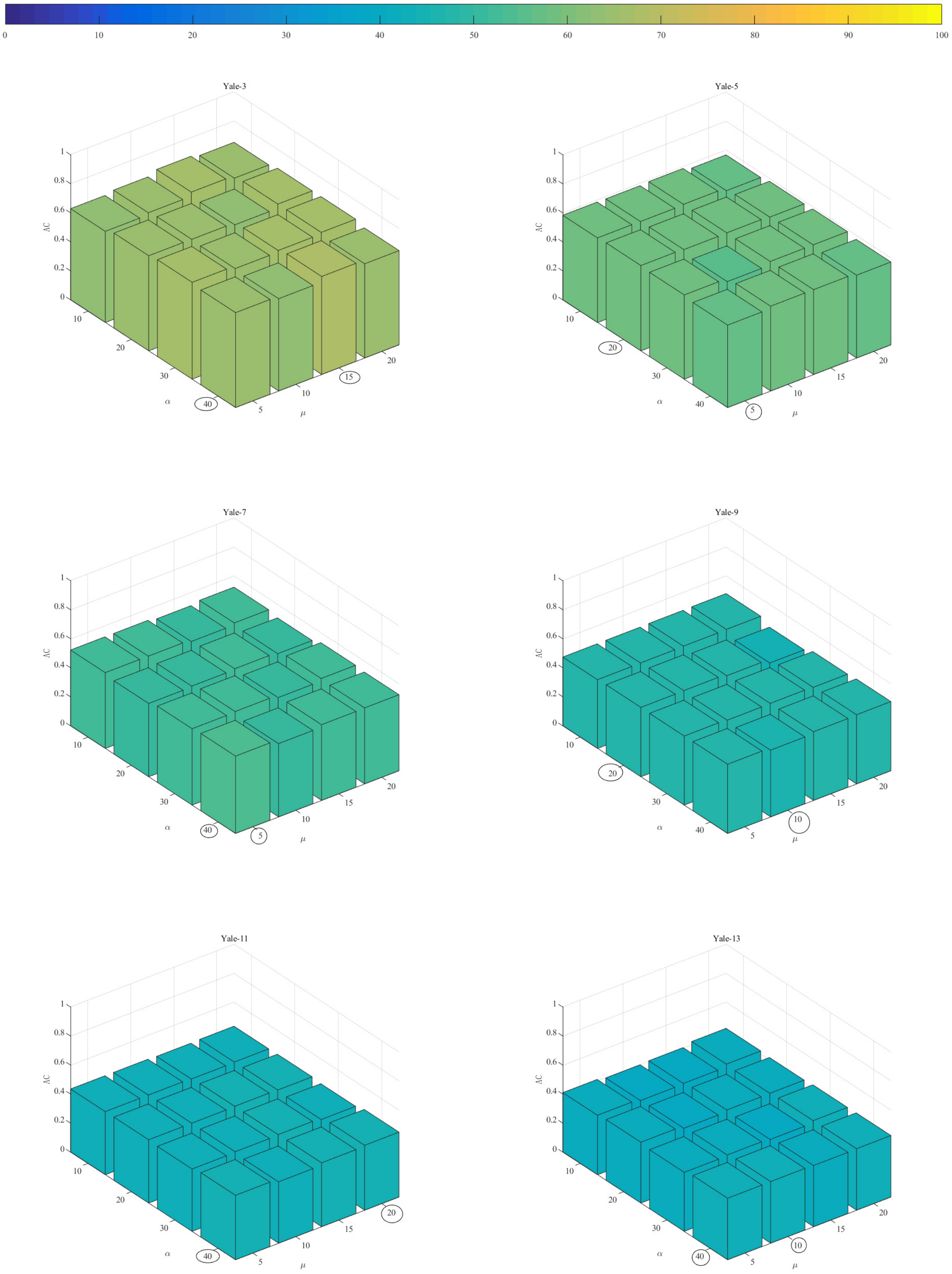

Figure 4. The clustering performance of the uni-AOGNTD method on the Yale datasets.

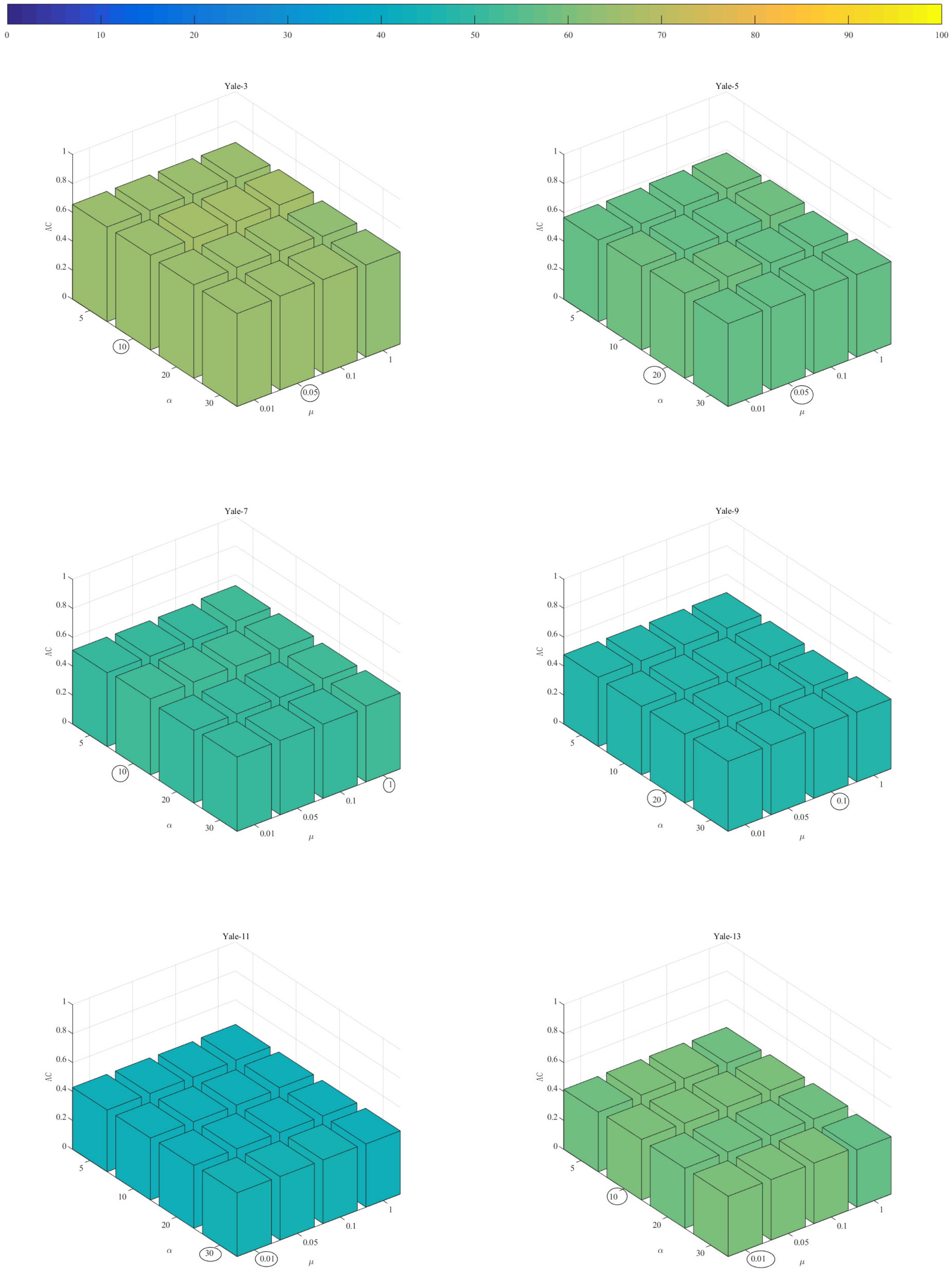

Figure 5. The clustering performance of the bi-AOGNTD method on the Yale datasets.

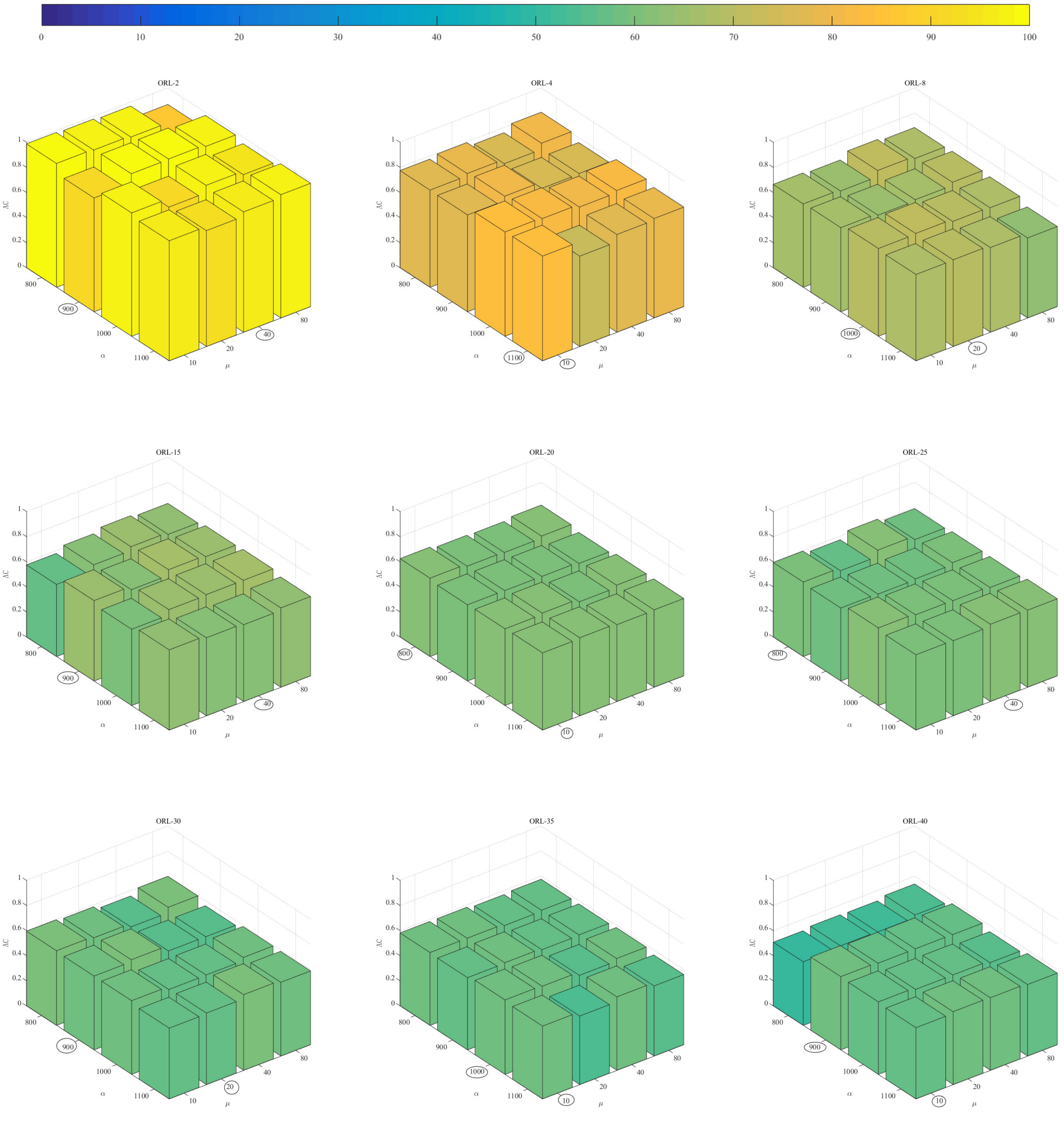

Figure 6. The clustering performance of the uni-AOGNTD method on the ORL datasets.

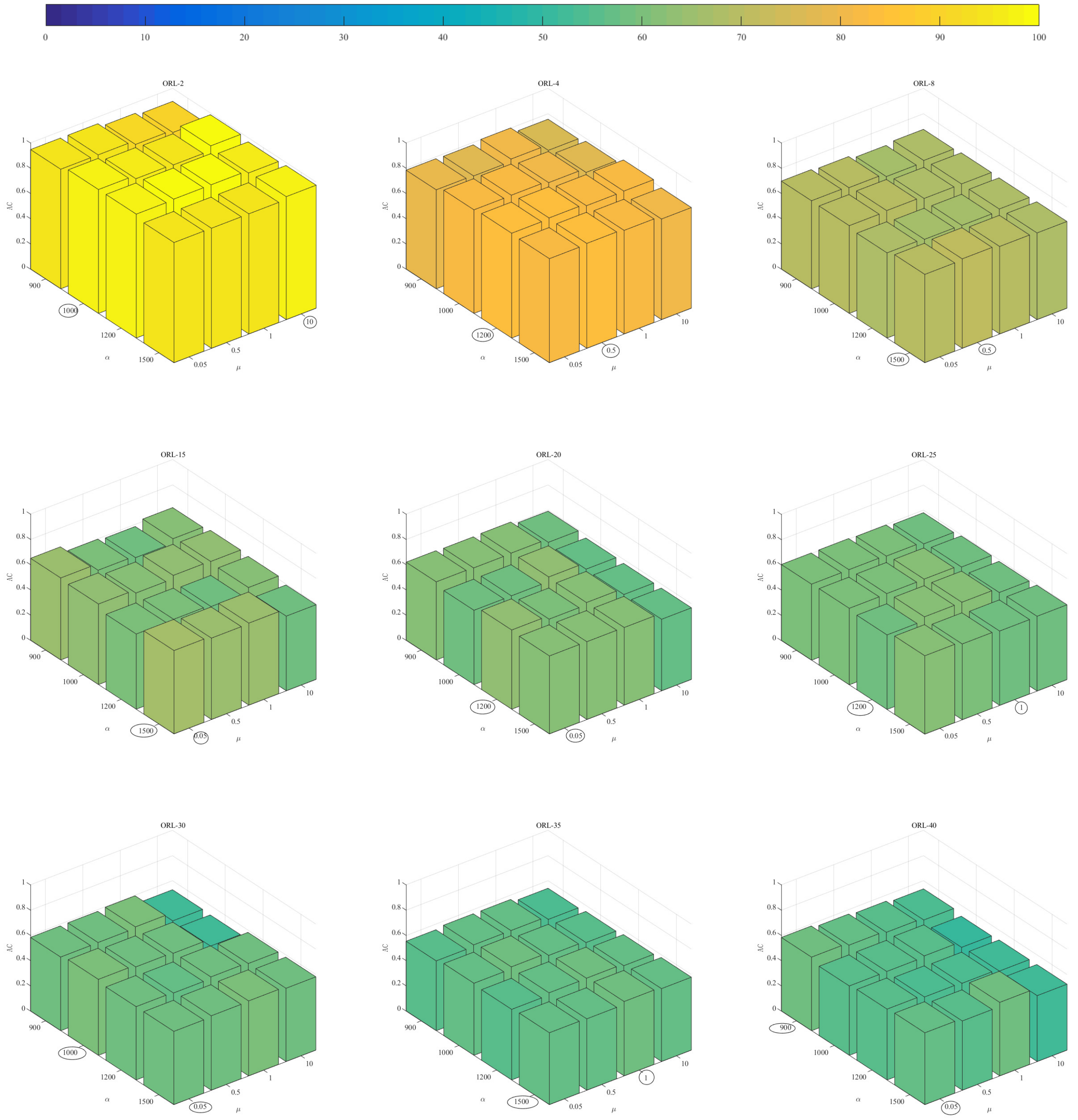

Figure 7. The clustering performance of the bi-AOGNTD method on the ORL datasets.

5.4. *Experiments for effectiveness and analysis*

Tables 3–8 show the average clustering results of the compared algorithms on three datasets, respectively. We bold mark the best results with each clustering number k on each dataset. In each experiment, we randomly selected k categories as the evaluated data. We construct the graph Laplacian using p-nearest neighbors in which the neighborhood size p is set to 5. We run the experiment and apply K-means 10 times on the low-dimensional representation matrix of each method. To make the experimental result validated, we repeat the above operations 10 times and calculate the averages, so there are a total of 100 calculations.

From Tables 3–8, our AOGNTD algorithms present better performance on the three datasets with varying category counts. In the comparison of data, the following phenomena can also be observed:

(1) For the COIL20 dataset, its performance is the most outstanding. The proposed uni-AOGNTD method attains 9%, 8.27%, 7.66%, 6.69%, 8.48%, 3.39% improvement corresponding to AC and 12.39%, 10.92%, 10.17%, 9.14%, 12.51%, 3.15% improvement corresponding to NMI in comparison with K-means, NMF, GNMF, GDNMF, NTD, GNTD, respectively. In the same way, the proposed bi-AOGNTD algorithm attains 8.65%, 7.92%, 7.31%, 6.34%, 8.13%, 3.04% improvement corresponding to AC and 12.41%, 10.96%, 10.22%, 9.18%, 12.55%, 3.19% improvement corresponding to NMI in comparison with K-means, NMF, GNMF, GDNMF, NTD, GNTD, respectively.

(2) For the Yale dataset, the uni-AOGNTD method attains 4.87%, 3.71%, 3.4%, 2.85%, 2.72%, and 2.02% improvement corresponding to AC and 5.11%, 4.63%, 4.33%, 3.09%, 3.38%, 2.38% improvement corresponding to NMI in comparison with K-means, NMF, GNMF, GDNMF, NTD, GNTD, respectively. The uni-AOGNTD and bi-AOGNTD models are about equal in problem-solving skills.

(3) For the ORL dataset, the uni-AOGNTD method attains 7.94%, 2.48%, 2.02%, 1.86%, 12.33%, and 0.72% improvement corresponding to AC and 9.48%, 4.31%, 3.88%, 3.72%, 15.29%, 1.1% improvement corresponding to NMI in comparison with K-means, NMF, GNMF, GDNMF, NTD, GNTD, respectively. The clustering effect of uni-AOGNTD method and bi-AOGNTD method are roughly the same.

Table 3. AC (%) of different algorithms on COIL20 dataset.

k	k-means	NMF	GNMF	GDNMF	NTD	GNTD	uni-AOGNTD	bi-AOGNTD
2	89.10±12.07	97.60±4.96	97.65±4.86	98.33±5.03	90.55±12.09	98.33±5.03	**100.00**	**100.00**
4	75.86±15.01	71.68±16.99	72.47±17.14	77.31±14.95	71.42±13.90	76.29±18.45	**81.94±15.35**	**81.46±17.55**
6	65.63±9.12	68.11±11.97	68.65±13.00	68.58±11.39	70.79±12.56	73.61±13.04	**75.73±14.88**	**75.10±13.77**
8	66.61±10.32	64.37±10.37	65.60±10.77	65.90±8.09	69.49±8.54	71.00±10.92	**74.73±10.22**	**73.80±10.85**
10	64.92±8.72	63.01±8.14	65.58±8.19	65.63±7.19	66.41±7.64	66.01±10.52	**71.64±9.21**	**72.29±10.04**
12	62.38±8.40	63.84±8.26	64.16±7.52	62.73±7.78	62.37±8.93	66.59±9.24	**70.55±8.57**	**70.21±9.82**
14	62.08±7.40	60.09±5.76	60.52±5.85	62.66±5.80	60.92±7.08	68.92±8.03	**70.46±8.12**	**70.21±7.94**
16	59.84±5.81	62.55±5.64	63.33±5.73	63.70±6.06	61.38±6.09	69.24±7.52	**70.69±7.05**	**70.04±7.39**
18	60.41±5.19	60.03±4.91	60.56±4.64	62.54±5.37	59.28±4.88	65.89±6.89	**69.16±7.27**	**68.91±6.34**
20	57.49±4.71	60.29±4.49	59.22±4.70	60.01±4.64	56.90±4.03	64.47±5.75	**69.35±5.20**	**68.79±6.28**
Avg.	66.43	67.16	67.77	68.74	66.95	72.04	**75.43**	**75.08**

Table 4. NMI (%) of different algorithms on COIL20 dataset.

k	k-means	NMF	GNMF	GDNMF	NTD	GNTD	uni-AOGNTD	bi-AOGNTD
2	65.77±30.18	91.29±17.74	91.38±17.53	94.59±16.31	72.77±31.77	94.59±16.31	**100.00**	**100.00**
4	69.61±15.08	69.05±16.24	69.99±17.15	74.89±11.95	69.64±13.21	71.24±17.77	**81.51±12.92**	**85.18±13.08**
6	66.49±8.37	69.69±11.15	70.68±12.54	66.91±10.45	70.39±12.97	78.68±10.49	**81.21±10.79**	**80.28±10.74**
8	71.64±7.72	69.25±9.14	70.60±9.13	70.62±6.76	73.10±7.25	79.90±7.88	**81.54±7.51**	**82.70±6.87**
10	73.85±6.80	70.39±7.17	72.18±7.40	72.95±5.68	72.72±6.05	76.73±10.00	**81.72±5.41**	**83.51±6.06**
12	72.32±6.53	71.76±6.11	72.15±5.94	71.83±6.84	70.77±7.14	79.16±7.36	**81.94±5.37**	**81.80±6.01**
14	74.36±5.58	70.62±4.84	71.11±4.53	72.82±4.20	71.16±5.16	82.12±4.70	**82.89±4.27**	**81.45±4.77**
16	72.96±3.57	73.17±3.64	73.99±3.57	74.48±3.75	71.84±4.21	82.81±3.99	**82.85±3.72**	**82.84±3.97**
18	74.37±3.02	71.92±2.81	72.27±2.62	74.66±3.37	71.42±2.85	81.00±3.86	**82.31±4.54**	**82.28±3.54**
20	73.43±2.42	72.40±2.36	72.64±2.54	73.59±2.43	69.76±2.39	81.00±3.14	**82.72±2.58**	**81.94±3.32**
Avg.	71.48	72.95	73.70	74.73	71.36	80.72	**83.87**	**83.91**

Table 5. AC (%) of different algorithms on Yale dataset.

k	k-means	NMF	GNMF	GDNMF	NTD	GNTD	uni-AOGNTD	bi-AOGNTD
3	62.03±10.93	59.73±9.23	59.97±9.35	62.12±9.66	60.82±11.96	61.55±12.06	**67.48±12.52**	**66.06±13.70**
5	51.42±8.72	54.42±8.37	55.56±8.64	56.00±6.62	56.02±11.44	56.16±11.52	**58.82±8.18**	**58.76±8.16**
7	47.58±7.31	48.91±6.24	49.03±6.05	48.56±6.94	49.45±7.26	50.83±7.78	**53.26±10.38**	**52.74±9.37**
9	42.70±5.72	45.03±5.05	45.16±5.33	45.32±5.91	47.44±6.05	47.59±5.55	**48.02±6.16**	**48.23±5.83**
11	40.41±4.78	41.51±4.55	41.74±4.67	43.31±4.80	41.94±4.60	43.96±4.57	**44.69±4.68**	**44.02±4.62**
13	39.08±4.53	41.54±3.67	41.72±3.86	41.59±4.01	41.80±4.67	41.60±4.52	**42.45±4.97**	**41.92±4.02**
15	38.43±3.71	38.64±3.80	38.73±3.68	38.88±3.10	39.25±4.00	39.94±3.64	**41.02±2.73**	**40.36±2.97**
Avg.	45.95	47.11	47.42	47.97	48.10	48.80	**50.82**	**50.30**

Table 6. NMI (%) of different algorithms on Yale dataset.

k	k-means	NMF	GNMF	GDNMF	NTD	GNTD	uni-AOGNTD	bi-AOGNTD
3	34.72±15.81	30.16±13.95	30.37±14.32	38.05±14.35	31.74±17.86	32.85±16.57	**39.26±17.48**	**38.44±19.12**
5	38.18±9.08	39.38±8.69	40.46±8.64	40.90±8.35	40.66±13.87	40.80±14.01	**45.91±12.66**	**46.92±11.59**
7	40.82±7.59	39.88±6.68	39.99±6.64	40.44±7.54	40.79±7.32	44.21±9.39	**46.67±11.39**	**46.14±10.51**
9	41.03±5.87	41.75±4.54	41.82±4.27	41.47±5.64	45.16±6.19	44.82±5.55	**45.26±6.01**	**45.36±5.80**
11	41.51±4.42	41.79±4.35	42.28±4.16	43.61±4.49	42.65±4.43	43.99±4.20	**44.81±3.99**	**44.05±4.28**
13	43.11±4.13	44.55±3.18	44.56±3.38	43.63±3.53	44.38±4.24	44.74±3.99	**45.63±4.21**	**45.26±3.70**
15	38.43±3.71	43.70±3.19	43.78±3.20	43.85±2.54	44.58±2.96	45.53±3.05	**46.06±2.59**	**45.68±2.58**
Avg.	39.69	40.17	40.47	41.71	41.42	42.42	**44.80**	**44.55**

Table 7. AC (%) of different algorithms on ORL dataset.

k	k-means	NMF	GNMF	GDNMF	NTD	GNTD	uni-AOGNTD	bi-AOGNTD
2	88.65±16.89	90.35±12.84	90.25±13.58	91.35±10.80	86.45±15.46	98.20±5.80	**99.00±3.02**	**98.70±3.23**
4	73.73±14.28	82.08±13.92	82.48±14.08	80.23±16.09	71.43±12.83	83.10±16.65	**83.40±16.44**	**83.80±16.16**
6	68.57±10.69	72.20±11.02	72.37±11.83	72.75±11.64	61.32±13.98	72.68±10.59	**75.30±12.14**	**76.35±10.86**
8	62.29±10.23	69.91±10.22	68.49±9.91	67.98±10.06	54.05±13.90	70.48±9.09	**70.55±9.87**	**70.88±9.39**
10	62.22±7.44	65.46±7.37	66.89±8.69	67.88±7.73	58.00±11.64	68.78±8.97	**69.15±8.91**	**69.05±9.14**
15	57.78±5.38	64.19±6.03	64.83±6.80	65.03±7.26	56.13±11.52	65.82±6.07	**65.99±6.88**	**65.99±6.31**
20	54.68±5.25	61.11±6.01	61.92±6.36	62.27±5.24	50.25±11.76	62.36±5.63	**62.49±6.80**	**62.79±5.67**
25	55.38±4.56	61.10±4.23	61.70±4.82	62.21±5.44	52.83±10.18	61.91±5.63	**62.05±5.14**	**62.20±5.68**
30	52.07±3.78	58.80±4.44	59.56±4.46	59.06±4.21	45.35±14.04	59.55±5.04	**60.44±4.03**	**60.23±3.71**
35	51.86±3.07	57.15±3.52	57.28±3.63	58.69±3.59	49.01±10.39	57.00±4.35	**59.31±3.85**	**58.91±3.99**
40	51.48±3.35	56.46±3.42	58.03±3.73	58.17±3.36	45.58±9.31	58.25±3.27	**58.38±3.15**	**58.36±3.36**
Avg.	61.70	67.16	67.62	67.78	57.31	68.92	**69.64**	**69.75**

Table 8. NMI(%) of different algorithms on ORL dataset.

k	k-means	NMF	GNMF	GDNMF	NTD	GNTD	uni-AOGNTD	bi-AOGNTD
2	70.82±40.76	70.64±34.38	71.39±34.53	73.87±32.41	62.85±35.80	93.53±15.58	**96.10±11.76**	**94.62±12.59**
4	68.42±16.36	79.91±12.97	80.06±13.49	77.86±15.48	64.81±14.60	80.18±19.54	**82.13±15.94**	**82.46±16.15**
6	69.52±9.93	74.09±9.67	74.01±11.38	74.10±11.53	60.51±17.75	76.41±8.52	**79.18±8.17**	**79.69±8.01**
8	65.60±10.31	74.99±8.75	73.82±8.16	72.66±9.05	55.10±17.20	75.27±7.38	**75.52±7.61**	**75.73±7.08**
10	68.35±6.79	73.18±5.33	73.80±6.37	75.33±5.98	63.76±13.26	75.73±7.47	**77.98±7.79**	**77.82±8.17**
15	69.53±4.25	76.00±4.69	76.38±5.10	75.83±4.99	66.67±12.51	76.64±4.86	**76.64±5.09**	**76.93±4.93**
20	69.21±4.42	74.10±4.32	74.67±4.72	75.24±3.62	64.33±11.76	75.59±4.15	**75.98±4.65**	**76.01±4.15**
25	71.68±3.25	75.77±2.66	76.62±3.26	76.92±3.62	68.37±9.31	77.04±3.91	**77.11±3.69**	**77.15±3.59**
30	70.58±2.41	75.39±2.97	76.13±2.65	75.66±2.68	62.90±12.97	76.28±2.88	**76.66±2.81**	**76.55±2.45**
35	71.17±1.94	74.71±2.11	75.35±2.09	76.02±2.11	67.46±9.66	75.39±2.77	**76.86±2.18**	**76.48±2.30**
40	71.84±2.01	74.91±1.90	76.03±1.90	76.55±1.83	66.07±7.95	76.79±1.79	**76.88±1.63**	**76.90±1.85**
Avg.	69.70	74.87	75.30	75.46	63.89	78.08	**79.18**	**79.12**

As can be seen, the uni-AOGNTD and bi-AOGNTD methods are superior to the compared methods, due to innovating and balancing the approximately orthogonal regularization and the graph regularization in conjunction with the unified NTD for tensor data representation. The local minimum solution is compressed to a better solution space under the adjustment of parameters. Intuitively, the nonnegativity of the factor matrix and the model constraint reduce the feasible solution domain to near local minima.

The clustering results of methods containing graph regularization terms for $\mathbf{A}^{(N)}$ are extended in

Figure 8. In short, the fresh approaches produced the highest and stablest AC and NMI.

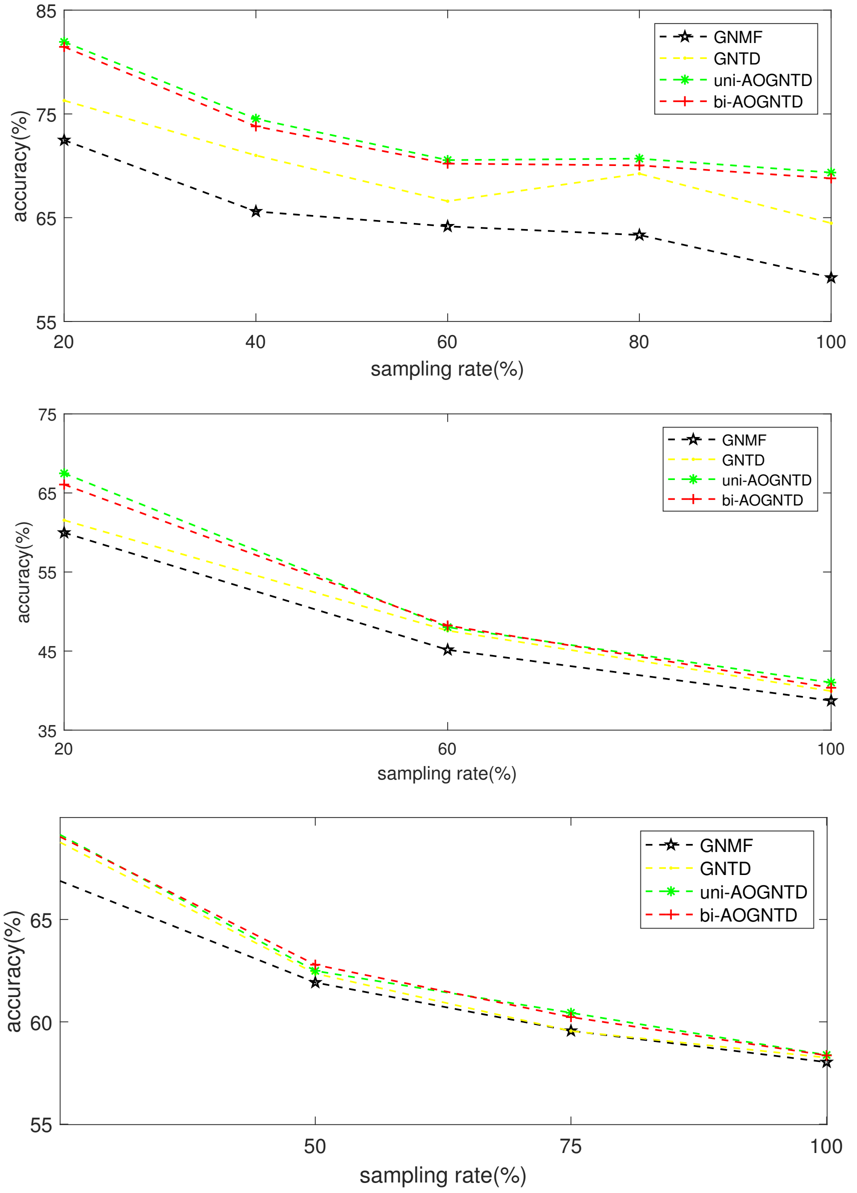

Figure 8. The clustering performance of four methods on COIL-20, Yale, and ORL datasets.

5.5. *Convergence study*

As described in Section 3, the convergence of the proposed algorithms has been theoretically proved. In this subsection, we experimentally study the convergence of the proposed algorithms by assuming link between the number of iterations and the value of the objective function in Figure 9. The trend intuitively suggests that the objective function can converge effectively under the multiplication iteration rules and explains the correctness of Theorem 4.

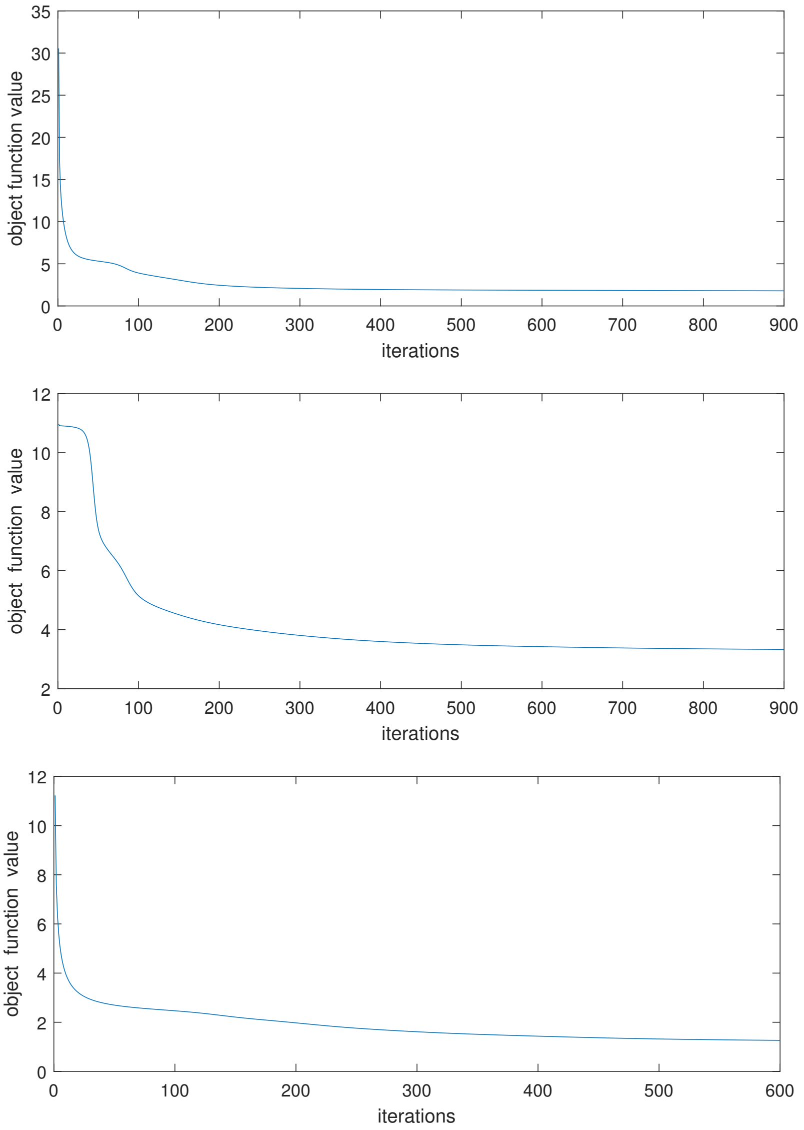

Figure 9. Convergence curves of AOGNTD on COIL-20, Yale, and ORL datasets.

6. Conclusions

In this paper, we built up two novel AOGNTD methods for tensor data representation. The convergence analysis is given. The AOGNTD method presents more competent representation and achieves better clustering performance on the publicly available real-world datasets through two adjustable regularization parameters. Although AOGNTD methods perform well in image clustering tasks, it can be further improved in two possible directions in the future. First, the graph in our algorithm is fixed, but it is thought-provoking whether to obtain joint information from multiple graphs or learn the optimal graph. In addition, a large number of NTD-based methods rely on K-means, and designing independent classification methods without any additional clustering procedure is the main point of further research.

Author contributions

Xiang Gao: material preparation, data collection and analysis, writing the first draft, commenting; Linzhang Lu: material preparation, data collection and analysis, commenting; Qilong Liu: material preparation, data collection and analysis, commenting. All authors have read and approved the final version of the manuscript for publication.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

Linzhang Lu is supported by the National Natural Science Foundation of China under Grant (12161020 and 12061025), and Guizhou Normal University Academic Talent Foundation under Grant (QSXM[2022]04). Qilong Liu is supported by Guizhou Provincial Basis Research Program (Natural Science) under Grant (QKHJC-ZK[2023]YB245).

Conflict of interest

The authors declare that they have no conflicts of interest in this paper.

References

1. P. Deng, T. Li, H. Wang, S. Horng, Z. Yu, X. Wang, Tri-regularized non-negative matrix tri-factorization for co-clustering, *Knowl. Based Syst.*, **226** (2021), 107101. https://doi.org/10.1016/j.knosys.2021.107101
2. S. Li, W. Li, J. Hu, Y. Li, Semi-supervised bi-orthogonal constraints dual-graph regularized NMF for subspace clustering, *Appl. Intell.*, **52** (2022), 3227–3248. https://doi.org/10.1007/s10489-021-02522-z
3. B. Cai, G. Lu, Tensor subspace clustering using consensus tensor low-rank representation, *Inf. Sci.*, **609** (2022), 46–59. https://doi.org/10.1016/j.ins.2022.07.049
4. M. Wall, A. Rechtsteiner, L. Rocha, Singular value decomposition and principal component analysis, In: D. P. Berrar, W. Dubitzky, M. Granzow, *A practical approach to microarray data analysis*, Springer, 2003, 91–109. https://doi.org/10.1007/0-306-47815-3_5
5. S. Roweis, L. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science*, **290** (2000), 2323–2326. https://doi.org/10.1126/science.290.5500.2323
6. W. Yin, Z. Ma, LE and LLE regularized nonnegative Tucker decomposition for clustering of high dimensional datasets, *Neurocomputing*, **364** (2019), 77–94. https://doi.org/10.1016/j.neucom.2019.06.054
7. A. Gersho, R. Gray, *Vector quantization and signal compression*, Springer, 2012. https://doi.org/10.1007/978-1-4615-3626-0

8. S. Wold, K. Esbensen, P. Geladi, Principal component analysis, *Chemometr. Intell. Lab. Syst.*, **2** (1987), 37–52. https://doi.org/10.1016/0169-7439(87)80084-9

9. Y. Zhao, C. Jiao, M. Wang, J. Liu, J. Wang, C. Zheng, Htrpca: hypergraph regularized tensor robust principal component analysis for sample clustering in tumor omics data, *Interdiscip. Sci.*, **14** (2022), 22–33. https://doi.org/10.1007/s12539-021-00441-8

10. D. Lee, H. Seung, Learning the parts of objects by non-negative matrix factorization, *Nature*, **401** (1999), 788–791. https://doi.org/10.1038/44565

11. D. Lee, H. Seung, Algorithms for non-negative matrix factorization, *Adv. Neural Inf. Process. Syst.*, **13** (2000), 556–562.

12. P. De Handschutter, N. Gillis, A consistent and flexible framework for deep matrix factorizations, *Pattern Recogn.*, **134** (2023), 109102. https://doi.org/10.1016/j.patcog.2022.109102

13. Z. Wang, P. Dellaportas, I. Kosmidis, Bayesian tensor factorisations for time series of counts, *Mach. Learn.*, **113** (2023), 3731–3750. https://doi.org/10.1007/s10994-023-06441-7

14. B. Chen, J. Guan, Z. Li, Unsupervised feature selection via graph regularized non-negative CP decomposition, *IEEE Trans. Pattern Anal. Mach. Intell.*, **45** (2022), 2582–2594. https://doi.org/10.1109/TPAMI.2022.3160205

15. M. Che, Y. Wei, Randomized algorithms for the approximations of Tucker and the tensor train decompositions, *Adv. Comput. Math.*, **45** (2019), 395–428. https://doi.org/10.1007/s10444-018-9622-8

16. T. Kolda, B. Bader, Tensor decompositions and applications, *SIAM Rev.*, **51** (2009), 455–500. https://doi.org/10.1137/07070111X

17. Y. Kim, S. Choi, Nonnegative Tucker decomposition, *2007 IEEE Conference on Computer Vision and Pattern Recognition*, 2007. https://doi.org/10.1109/CVPR.2007.383405

18. H. Huang, Z. Ma, G. Zhang, Dimensionality reduction of tensors based on manifold-regularized tucker decomposition and its iterative solution, *Int. J. Mach. Learn. Cybern.*, **13** (2022), 509–522. https://doi.org/10.1007/s13042-021-01422-5

19. J. Zhang, Y. Han, J. Jiang, Semi-supervised tensor learning for image classification, *Multimedia Syst.*, **23** (2017), 63–73. https://doi.org/10.1007/s00530-014-0416-7

20. X. Zhang, M. Ng, Sparse nonnegative Tucker decomposition and completion under noisy observations, *ArXiv*, 2022. https://doi.org/10.48550/arXiv.2208.08287

21. Q. Liu, L. Lu, Z. Chen, Nonnegative Tucker decomposition with graph regularization and smooth constraint for clustering, *Pattern Recogn.*, **148** (2023), 110207. https://doi.org/10.1016/j.patcog.2023.110207

22. Y. Qiu, G. Zhou, Y. Zhang, S. Xie, Graph regularized nonnegative Tucker decomposition for tensor data representation, *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, 8613–8617. https://doi.org/10.1109/ICASSP.2019.8683766

23. Y. Qiu, G. Zhou, Y. Wang, Y. Zhang, S. Xie, A generalized graph regularized non-negative Tucker decomposition framework for tensor data representation, *IEEE Trans. Cybern.*, **52** (2020), 594–607. https://doi.org/10.1109/TCYB.2020.2979344

24. D. Chen, G. Zhou, Y. Qiu, Y. Yu, Adaptive graph regularized non-negative Tucker decomposition for multiway dimensionality reduction, *Multimedia Tools Appl.*, **83** (2024), 9647–9668. https://doi.org/10.1007/s11042-023-15622-4

25. X. Li, M. Ng, G. Cong, Y. Ye, Q. Wu, MR-NTD: manifold regularization nonnegative tucker decomposition for tensor data dimension reduction and representation, *IEEE Trans. Neural Networks Lear. Syst.*, **28** (2016), 1787–1800. https://doi.org/10.1109/TNNLS.2016.2545400

26. Z. Huang, G. Zhou, Y. Qiu, Y. Yun, Y. Dai, A dynamic hypergraph regularized non-negative Tucker decomposition framework for multiway data analysis, *Int. J. Mach. Learn. Cybern.*, **13** (2022), 3691–3710. https://doi.org/10.1007/s13042-022-01620-9

27. W. Jing, L. Lu, Q. Liu, Graph regularized discriminative nonnegative Tucker decomposition for tensor data representation, *Appl. Intell.*, **53** (2023), 23864–23882. https://doi.org/10.1007/s10489-023-04738-7

28. Y. Qiu, G. Zhou, X. Chen, D. Zhang, X. Zhao, Q. Zhao, Semi-supervised non-negative Tucker decomposition for tensor data representation, *Sci. China Technol. Sci.*, **64** (2021), 1881–1892. https://doi.org/10.1007/s11431-020-1824-4

29. L. Ren, R. Hu, Y. Liu, D. Li, J. Wu, Y. Zang, et al., Improving fraud detection via imbalanced graph structure learning, *Mach. Learn.*, **113** (2023), 1069–1090. https://doi.org/10.1007/s10994-023-06464-0

30. M. Zhao, W. Li, L. Li, P. Ma, Z. Cai, R. Tao, Three-order tensor creation and Tucker decomposition for infrared small-target detection, *IEEE Trans. Geosci. Remote Sens.*, **60** (2021), 1–16. https://doi.org/10.1109/TGRS.2021.3057696

31. T. Jiang, M. K. Ng, J. Pan, G. Song, Nonnegative low rank tensor approximations with multidimensional image applications, *Numer. Math.*, **153** (2023), 141–170. https://doi.org/10.1007/s00211-022-01328-6

32. C. Ding, X. He, H. Simon, On the equivalence of non-negative matrix factorization and spectral clustering, *Proceedings of the 2005 SIAM International Conference on Data Mining*, 2005, 606–610. https://doi.org/10.1137/1.9781611972757.70

33. J. Pan, M. Ng, Y. Liu, X. Zhang, H. Yan, Orthogonal nonnegative Tucker decomposition, *SIAM J. Sci. Comput.*, **43** (2021), B55–B81. https://doi.org/10.1137/19M1294708

34. B. Li, G. Zhou, A. Cichocki, Two efficient algorithms for approximately orthogonal nonnegative matrix factorization, *IEEE Signal Process. Lett.*, **22** (2015), 843–846. https://doi.org/10.1109/LSP.2014.2371895

35. Y. Qiu, W. Sun, Y. Zhang, X. Gu, G. Zhou, Approximately orthogonal nonnegative Tucker decomposition for flexible multiway clustering, *Sci. China Technol. Sci.*, **64** (2021), 1872–1880. https://doi.org/10.1007/s11431-020-1827-0

36. D. Cai, X. He, J. Han, T. Huang, Graph regularized nonnegative matrix factorization for data representation, *IEEE Trans. Pattern Anal. Mach. Intell.*, **33** (2010), 1548–1560. https://doi.org/10.1109/TPAMI.2010.231

37. F. Shang, L. Jiao, J. Shi, F. Wang, M. Gong, Fast affinity propagation clustering: a multilevel approach, *Pattern Recogn.*, **45** (2012), 474–486. https://doi.org/10.1016/j.patcog.2011.04.032

38. F. Shang, L. Jiao, F. Wang, Graph dual regularization non-negative matrix factorization for co-clustering, *Pattern Recogn.*, **45** (2012), 2237–2250. https://doi.org/10.1016/j.patcog.2011.12.015