BIG DATA MINING AND ANALYTICS
ISSN 2096-0654 11/14 pp214–232
DOI: 10.26599/BDMA.2024.9020055
Volume 8, Number 1, February 2025

Robust Non-Negative Matrix Tri-Factorization with Dual Hyper-Graph Regularization

Jiyang Yu, Hangjun Che*, Man-Fai Leung, Cheng Liu, Wenhui Wu, and Zheng Yan

Abstract: Non-negative Matrix Factorization (NMF) has been an ideal tool for machine learning. Non-negative Matrix Tri-Factorization (NMTF) is a generalization of NMF that incorporates a third non-negative factorization matrix, and has shown impressive clustering performance by imposing simultaneous orthogonality constraints on both sample and feature spaces. However, the performance of NMTF dramatically degrades if the data are contaminated with noises and outliers. Furthermore, the high-order geometric information is rarely considered. In this paper, a Robust NMTF with Dual Hyper-graph regularization (namely RDHNMTF) is introduced. Firstly, to enhance the robustness of NMTF, an improvement is made by utilizing the $l_{2,1}$- norm to evaluate the reconstruction error. Secondly, a dual hyper-graph is established to uncover the higher-order inherent information within sample space and feature spaces for clustering. Furthermore, an alternating iteration algorithm is devised, and its convergence is thoroughly analyzed. Additionally, computational complexity is analyzed among comparison algorithms. The effectiveness of RDHNMTF is verified by benchmarking against ten cutting-edge algorithms across seven datasets corrupted with four types of noise.

Key words: Non-negative Matrix Tri-Factorization (NMTF); $l_{2,1}$-norm; dual hyper-graph regularization; co-clustering

1 Introduction

In the field of machine learning and data mining, one is often confronted with extremely high dimensional data. Due to the excessively high dimensionality of the original data, matrix factorization techniques are applied to reduce the dimensionality of high-dimensional data into a low-dimensional subspace and uncover latent structures within the data. The most classic matrix factorization methods include Principal Component Analysis (PCA), Singular Value Decomposition (SVD), and Non-negative Matrix

• Jiyang Yu is with School of Electronic and Information Engineering, Southwest University, Chongqing 400715, China. E-mail: jiyangyuzzz@163.com.
• Hangjun Che is with College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China, and also with Chongqing Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, Chongqing 400715, China. E-mail: hjche123@swu.edu.cn.
• Man-Fai Leung is with School of Computing and Information Science, Faculty of Science and Engineering, Anglia Ruskin University, Cambridge CB1 1PT, UK. E-mail: man-fai.leung@aru.ac.uk.
• Cheng Liu is with Department of Computer Science, Shantou University, Shantou 515063, China. E-mail: chengliu10@gmail.com.
• Wenhui Wu is with College of Electronics and Information Engineering, Shenzhen University, Shenzhen 518060, China. E-mail: wuwenhui@szu.edu.cn.
• Zheng Yan is with Australia Artificial Intelligence Institute, University of Technology Sydney, Sydney 2007, Australia. E-mail: Yan.Zheng@uts.edu.au.
* To whom correspondence should be addressed.
Manuscript received: 2024-04-28; revised: 2024-08-11; accepted: 2024-08-22

Factorization (NMF)[1].

In contrast to alternative matrix factorization methods, NMF factorizes a high-dimensional data matrix into two low-rank matrices that contain only non-negative elements, thus the decomposed matrices are able to approximate to the original matrix. By enforcing a non-negativity constraint, NMF obtains a part-based representation of data[1]. The NMF algorithm and its variants have been utilized in various domains, e.g., medical research[2–4], medium data learning[5–7], community discovery[8–12], gene expression[13–16], text mining[17, 18], and neural network[19, 20].

It is widely recognized that NMF is susceptible to noise and outliers, as NMF uses Euclidean distance to measure the similarity. Additionally, the real-world data often contain Gaussian noise and non-Gaussian noise. Therefore, multiple NMF algorithms based on robust M-estimators have been introduced. Kong et al.[21] introduced a robust NMF utilizing $l_{2,1}$-norm. Unlike NMF algorithm calculates the squared residual difference, $l_{2,1}$-norm measures the residual error without squaring them. Therefore, $l_{2,1}$-norm can mitigate the impact of noise and outliers. After that, Gao et al.[22] proposed a robust capped NMF, which introduces a cap on the residual error of the outliers to improve its robustness. Guan et al.[23] developed truncated Cauchy NMF, which employs Cauchy loss with truncation to limit significant errors and address outliers. Manhattan NMF (MahNMF) utilizes Manhattan distance rather than Euclidean distance to incorporate heavy-tailed Laplacian noise into the model[24]. Lin et al.[25] proposed a robust matrix factorization utilizing l_1-norm, which is solved by majorization minimization technique. Some algorithms incorporate correntropy to NMF to mitigate the influence of non-Gaussian noise and outliers. Yang et al.[26] proposed efficient correntropy-based multi-view clustering to enhance robustness and improve clustering performance. Graph regularized non-negative matrix factorization by maximizing correntropy method utilizes maximizing correntropy technique to handle nonlinear cases[27].

To capture the geometric structure of the data, manifold learning is introduced in the factorization process[28]. Graph regularized NMF (GNMF) is proposed to obtain the geometrical information in original sample space by using k-nearest-neighoor method, and further embedding the learned garph in the objective function[29]. Conventional graph only considers one edge connected between every pair of samples, which ignores the intrinsic information in feature space. Thus graph Dual regularization NMF (DNMF) was proposed to simultaneously explore the manifold embedded in both sample and feature space[30]. Tolić[31] et al. proposed an NMF method that takes into account the nonlinear geometric structure of the data. Unlike a graph, a hyper-graph is constructed using edges that connect multiple samples. As a result, a hyper-graph is capable of capturing high-order relationships within the data, making it widely utilized across various domains (e.g., image classification[32], neural networks[33], and hyperspectral unmixing[34]). Zeng et al.[35] proposed Hyper-graph regularized NMF (HNMF), which utilizes a hyper-graph instead of a traditional graph to capture the distinctive geometric structure of the data. Experiments have shown that NMF combined with manifold learning can achieve a better clustering performance[29, 35].

Non-negative Matrix Tri-Factorization (NMTF) is an extension of NMF. NMTF allows for the simultaneous clustering of both feature and sample spaces, making it a valuable method for co-clustering tasks (e.g., text mining[36] and gene expression[37]). Graph Dual regularized NMTF (DNMTF) combines manifold learning and standard NMTF method[30]. Deng et al.[38] developed tri-regularized non-negative matrix tri-factorization which adds graph regularization, l_1-norm, and NMTF to the objective function. Ding et al.[39] were the first to express the opinion that unconstrained NMTF is equal to NMF, whereas constrained NMTF introduces new features to NMF. Additionally, Ding introduced Orthogonal NMTF (ONMTF). Orthogonality constraints in ONMTF are able to enforce the parts-based representation and lead to a rigorous clustering interpretation. After that, Wang and Huang[40] Proposed Penalized NMF (PNMTF), where three penalty terms are integrated to ensure the near orthogonality of the matrix. Peng et al.[41] proposed the Correntropy based orthogonal NMTF (CNMTF) algorithm. Apply orthogonality constrains and utilize correntropy to measure the similarity is the main feature of CNMTF.

Inspired by the advantages of ONMTF and manifold learning, a robust NMTF with dual hyper-graph regularization is proposed. The primary contribution of the article can be summarized as follows:

(1) Robust NMTF with Dual Hyper-graph regularization (namely RDHNMTF) is proposed. Instead of the widely used NMF framework, orthogonal NMTF framework is adopted to obtain wider degrees of freedom. Moreover, unlike other algorithms that utilize a single graph, dual graph, or a single hyper-graph, RDHNMTF incorporates dual hyper-graph regularization to exploit the intrinsic manifold structure of the sample space and feature space. To enhance robustness without increasing the complexity of the update procedures, RDHNMTF utilizes the l_{21}-norm instead of Euclidean norm.

(2) The update rules are designed for the proposed model within the framework of the alternating iteration algorithm. The objective function is non-increasing under the proposed update rules is theoretically proved. The computational complexity of RDHNMTF and its comparative methods is thoroughly evaluated using three arithmetic operations.

(3) The performance of RDHNMTF is substantiated to be effective by compare RDHNMTF with ten state-of-the-art algorithms on seven datasets with four kinds of noises. The experiments on seven datasets contaminated with noises and outliers demonstrates the robustness of RDHNMTF. The ablation study demonstrates the superiority of incorporating both dual hyper-graph regularization and orthogonality constraints.

The remaining sections of this paper are structured as follows: NMF algorithm, $l_{2,1}$-norm, manifold learning, and hyper-graph regularization are introduced in Section 2. In Section 3, the formulation of RDHNMTF is described. An alternating iteration algorithms is designed thereafter. The computational complexity of RDHNMTF and its comparison algorithms is also comprehensively examined. In Section 4, the clustering results on seven datasets are presented and compared against ten state-of-the-art algorithms. The paper is concluded in Section 5.

2 Related Work

The primary goal of RDHNMTF is to robustly and efficiently cluster data. To achieve this goal, the algorithm utilizes three related techniques, namely NMF, $l_{2,1}$-norm and manifold learning, which are introduced below.

2.1 NMF

The data matrix is represented as $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n] \in \mathbf{R}^{m\times n}$, where m represents the number of features of $\boldsymbol{X}$, n represents the number of samples of $\boldsymbol{X}$, and $\boldsymbol{x}_n$ represents the n-th sample vector of $\boldsymbol{X}$. NMF decomposes $\boldsymbol{X}$ into the product of two matrices with low dimensions and non-negative values $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_k] \in \mathbf{R}^{m\times k}$ and $\boldsymbol{V}^{\mathrm{T}} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \dots, \boldsymbol{v}_n] \in \mathbf{R}^{k\times n}$, where k denotes both the number of samples of the factorized matrix $\boldsymbol{U}$ and the number of features of the fatorized matrix $\boldsymbol{V}$. Frobenius norm is applied to measure the difference between $\boldsymbol{X}$ and $\boldsymbol{U}\boldsymbol{V}^{\mathrm{T}}$. The optimization problem of NMF can be defined as follows[1]:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \left\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{V}^{\mathrm{T}}\right\|_{\mathrm{F}}^{2}, \quad \text{s.t.,} \quad \boldsymbol{U}\geqslant 0,\ \boldsymbol{V}\geqslant 0 \tag{1}$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm. Optimization of Formula (1) can be addressed using the following multiplicative update rules[1]:

$$\boldsymbol{U} \leftarrow \boldsymbol{U}\frac{\boldsymbol{X}\boldsymbol{V}}{\boldsymbol{U}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}} \tag{2}$$

$$\boldsymbol{V} \leftarrow \boldsymbol{V}\frac{\boldsymbol{X}^{\mathrm{T}}\boldsymbol{U}}{\boldsymbol{V}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}} \tag{3}$$

2.2 $l_{2,1}$-norm

The $l_{2,1}$-norm applied in NMF is calculated as follows:

$$\left\|\boldsymbol{X}-\mathbf{U}\mathbf{V}^{\mathrm{T}}\right\|_{2,1} = \sum_{i=1}^{n}\|\boldsymbol{x}_i-\boldsymbol{U}\boldsymbol{v}_i\| \tag{4}$$

the $l_{2,1}$-norm measure the sum of residue error of each data point without squaring them, thus alleviating the influence of outlier data points and noise[21].

2.3 Manifold learning

Manifold learning discovers low-dimensional structures in high-dimensional spaces. Hyper-graph regularized algorithms (e.g., HNMF) incorporate the manifold learning to NMF framework by constructing a hyper-graph of sample space. As show in Fig. 1, the key difference between a simple graph and a hyper-graph is that a graph's edge only connects two nodes, whereas a hyper-graph's edge connects multiple nodes, preserving multivariate relationships without reducing them to binary ones.

Hyper-graph $G = (V, E, \boldsymbol{W})$ comprises a vertex set V, a hyper-edge set E, and a diagonal hyper-edge weight matrix $\boldsymbol{W}$. Given vertex v and hyper-edge e, the incidence matrix $\boldsymbol{H} \in \mathbf{R}^{|V|\times|E|}$ is defined in the following manner:

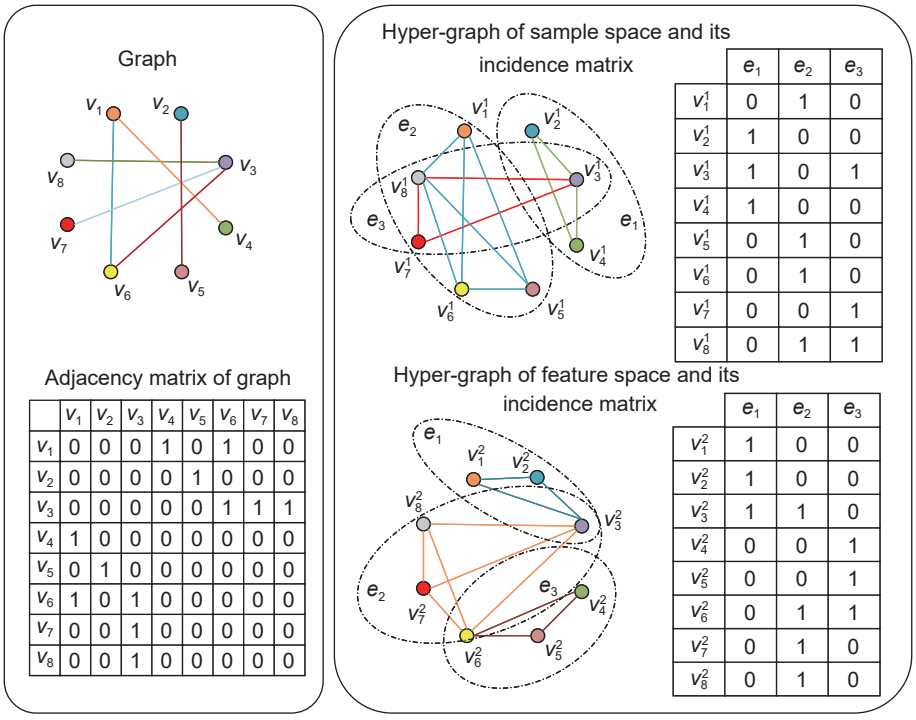

Fig. 1 Comparison between graph and hyper-graph. v_n represents the n-th vertex in graph, v_n^1 represents the n-th vertex in the hyper-graph of sample space, v_n^2 represents the n-th vertex in the hyper-graph of feature space, and e_n denotes the n-th hyper-edge of the hyper-graph.

$$\boldsymbol{H}(v,e)=\begin{cases}1, & \text{if } v\in e;\\ 0, & \text{if } v\notin e\end{cases} \tag{5}$$

where $\boldsymbol{W}_i$ denotes the weight of hyper-edge e_i. Specifically, $\boldsymbol{W}_i$ is defined as

$$\boldsymbol{W}_i=\sum_{v_x,\,v_y\in e_i}\exp\left(-\frac{\|v_x-v_y\|_2^2}{\sigma^2}\right) \tag{6}$$

σ^2 is defined as follows:

$$\sigma^2=\sum_{v_x,\,v_y\in e_i}\frac{\|v_x-v_y\|_2^2}{k_{\text{nearest}}} \tag{7}$$

where k_{nearest} denotes the number of k-nearest neighbors for each vertex. $d(v)$ denotes the degree of a vertex v, which is defined as

$$d(v)=\sum_{e\in E}w(e)\boldsymbol{H}(v,e) \tag{8}$$

where $w(e)$ denotes the weight of hyper-dege e, $f(e)$ denotes the degree of an edge e, which is defined as

$$f(e)=\sum_{v\in V}\boldsymbol{H}(v,e) \tag{9}$$

Assuming that the diagonal matrix $\boldsymbol{D}_v$ comprises $d(v)$ and diagonal matrix $\boldsymbol{D}_e$ comprises $f(e)$, the unnormalized hyper-graph Laplacian matrix $\boldsymbol{L}_{\text{rmhyper}}$ is calculated as $\boldsymbol{L}_{\text{hyper}}=\boldsymbol{D}_v-\boldsymbol{S}$, where $\boldsymbol{S}=\boldsymbol{H}\boldsymbol{W}\boldsymbol{D}_e^{-1}\boldsymbol{H}^{-1}$.

3 Proposed Model

In this section, the novel RDHNMTF is introduced. The convergence and the complexity analysis of RDHNMTF are discussed subsequently.

3.1 Problem formulation

Firstly, $l_{2,1}$-norm is applied to improve the robustness of RDHNMTF. Specifically, $l_{2,1}$-norm calculates the residual error of sample point x_i in the manner of $\|\boldsymbol{x}_i-\boldsymbol{U}\boldsymbol{v}_i\|$ without squaring them, while the Forbenius norm of traditional NMF squares the residual error of sample point x_i in the manner of $\|\boldsymbol{x}_i-\boldsymbol{U}\boldsymbol{v}_i\|^2$. Therefore, when the sample point x_i is an outlier, $l_{2,1}$-norm can decrease the loss of x_i. Besides, the $l_{2,1}$-norm makes the basis matrix sparser. RDHNMTF also utilizes the dual hyper-graph regularization. The dual hyper-graph regularization approach employs two nearest-neighbor hyper-graphs to capture the geometric structure of both the sample space and feature space in a lower-dimensional space. The flowchart of RDHNMTF is displayed in Fig. 2. Ultimately, RDHNMTF is formulated as follows:

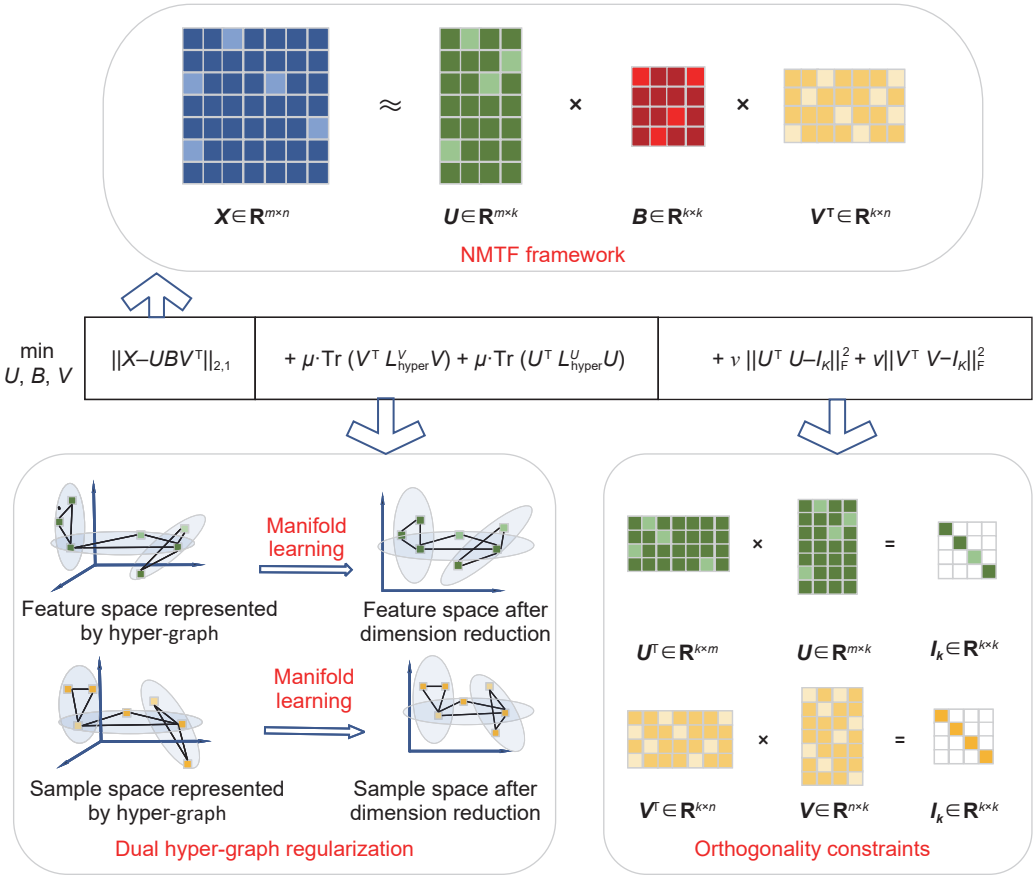

Fig. 2 Flowchart of the proposed RDHNMTF. Dual hyper-graph regularization: Dual hyper-graph regularization helps to discover the low-dimensional geometric structure embedded in the high-dimensional sample space and feature space. Orthogonality constraints: Use of orthogonality constraints allows the proposed algorithm to capture the most distinctive components and enhance the uniqueness of the decomposed outcomes.

$$\begin{aligned}&\min_{\boldsymbol{U},\boldsymbol{B},\boldsymbol{V}}\left\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}}\right\|_{2,1}+\\&\quad\mu\left(\mathrm{Tr}\left(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{L}^{\boldsymbol{V}}_{\mathrm{hyper}}\boldsymbol{V}\right)+\mathrm{Tr}\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{L}^{\boldsymbol{U}}_{\mathrm{hyper}}\boldsymbol{U}\right)\right)+\\&\quad\nu\left(\left\|\boldsymbol{I}_K-\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\right\|^2_{\mathrm{F}}+\left\|\boldsymbol{I}_K-\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}\right\|^2_{\mathrm{F}}\right),\\&\quad\text{s.t.},\quad\boldsymbol{U}\geqslant 0,\ \boldsymbol{B}\geqslant 0,\ \boldsymbol{V}\geqslant 0\end{aligned}\tag{10}$$

where $\left\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}}\right\|_{2,1}$ indicates using $l_{2,1}$-norm to measure the reconstruction error. Besides, $\mu\left(\mathrm{Tr}\left(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{L}^{\boldsymbol{V}}_{\mathrm{hyper}}\boldsymbol{V}\right)+\mathrm{Tr}\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{L}^{\boldsymbol{U}}_{\mathrm{hyper}}\boldsymbol{U}\right)\right)$ denotes the dual hyper-graph regularization, where $\mu\geqslant 0$ denotes dual hyper-graph regularization parameter. $\nu\left(\left\|\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}-\boldsymbol{I}_K\right\|^2_{\mathrm{F}}+\left\|\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}-\boldsymbol{I}_K\right\|^2_{\mathrm{F}}\right)$ indicates orthogonality constraints, where $\nu\geqslant 0$ denotes orthogonality parameter. Orthogonality constraints is vital to HNMTF as orthogonality constraints can lead to a better clustering interpretation[39]. Specifically, orthogonality constraint $\nu\left\|\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}-\boldsymbol{I}_K\right\|^2_{\mathrm{F}}$ leads to a better clustering interpretation, while orthogonality constraint $\nu\left\|\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}-\boldsymbol{I}_K\right\|^2_{\mathrm{F}}$ helps to cluster the rows and columns of the data matrix simultaneously and prevent the mapping $\boldsymbol{U}\leftarrow\boldsymbol{U}\boldsymbol{B}$, without which the proposed NMTF algorithms may degenerate into an NMF method with only one orthogonal constraint imposed on matrix $\boldsymbol{V}$.

3.2 Optimization of RDHNMTF

It is evident that the optimization Formula (10) is non-convex, making it challenging to directly solve the problem. To facilitate the optimization process, the optimization problem is formulated as follows:

$$\begin{aligned}&\min_{\boldsymbol{U},\boldsymbol{B},\boldsymbol{V}}\mathrm{Tr}\left(\left(\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}}\right)\boldsymbol{D}\left(\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}}\right)^{\mathrm{T}}\right)+\\&\quad\alpha\left(\mathrm{Tr}\left(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{L}^{\boldsymbol{V}}_{\mathrm{hyper}}\boldsymbol{V}\right)+\mathrm{Tr}\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{L}^{\boldsymbol{U}}_{\mathrm{hyper}}\boldsymbol{U}\right)\right)+\\&\quad\beta\left(\left\|\boldsymbol{I}_k-\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\right\|^2_{\mathrm{F}}+\left\|\boldsymbol{I}_k-\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}\right\|^2_{\mathrm{F}}\right),\\&\quad\text{s.t.},\quad\boldsymbol{U}\geqslant 0,\ \boldsymbol{B}\geqslant 0,\ \boldsymbol{V}\geqslant 0\end{aligned}\tag{11}$$

where $\alpha=2\mu$, $\beta=2\nu$, and $\boldsymbol{D}$ is a diagonal matrix with its i-th diagonal element calculated as follows:

$$\boldsymbol{D}_{ii}=\frac{1}{\left\|(\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}})_i\right\|} \tag{12}$$

Theorem 1 The updates rules derived from optimization Formula (11) are able to optimize the optimization Formula (10).

The detailed proof of Theorem 1 is provided in Appendix A.

According to Theorem 1, the optimization Formula (10) can be solved by optimizing intermediate Formula (11). Thus, we design an optimization strategy for solving Formula (11) subsequently.

3.3 Update rules of RDHNMF

Considering $\|\boldsymbol{X}\|_{\mathrm{F}}^2=\mathrm{Tr}\,(\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}})$, the optimization Formula (11) can be rewritten in the following manner:

$$\begin{aligned}\min_{\boldsymbol{U},\boldsymbol{B},\boldsymbol{V}}\ &\mathrm{Tr}\,(\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^{\mathrm{T}}-2\boldsymbol{U}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{X}^{\mathrm{T}}+\boldsymbol{U}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{V}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})+\\ &\alpha\,(\mathrm{Tr}\,(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{L}_{\mathrm{hyper}}^{\boldsymbol{V}}\boldsymbol{V})+\mathrm{Tr}\,(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{L}_{\mathrm{hyper}}^{\boldsymbol{U}}\boldsymbol{U}))+\\ &\beta\,(\mathrm{Tr}\,(\boldsymbol{I}_k-2\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}+\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U})+\\ &\mathrm{Tr}\,(\boldsymbol{I}_k-2\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}+\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}))\end{aligned} \tag{13}$$

Optimization Formula (13) can be solved by multiplicative iteration method. Let $\boldsymbol{\phi}=[\phi_{jk}]\in\mathbf{R}_{\geqslant 0}^{m\times k}$, $\boldsymbol{\omega}=[\omega_{kk'}]\in\mathbf{R}_{\geqslant 0}^{k\times k}$, and $\boldsymbol{\psi}=[\psi_{ik}]\in\mathbf{R}_{\geqslant 0}^{n\times k}$ be the Lagrange multipliers for $\boldsymbol{U}$, $\boldsymbol{B}$, and $\boldsymbol{V}$, respectively, Lagrange function $\mathcal{L}$ is obtained as follows:

$$\begin{aligned}\mathcal{L}=&\mathrm{Tr}\,(\boldsymbol{X}\boldsymbol{D}\boldsymbol{X}^{\mathrm{T}}-2\boldsymbol{U}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{X}^{\mathrm{T}}+\boldsymbol{U}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{V}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}})+\\ &\alpha\,(\mathrm{Tr}\,(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{L}_{\mathrm{hyper}}^{\boldsymbol{V}}\boldsymbol{V})+\mathrm{Tr}\,(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{L}_{\mathrm{hyper}}^{\boldsymbol{U}}\boldsymbol{U}))+\\ &\beta\,(\mathrm{Tr}\,(\boldsymbol{I}_k-2\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}+\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U})+\\ &\mathrm{Tr}\,(\boldsymbol{I}_k-2\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}+\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}))+\\ &\mathrm{Tr}\,(\boldsymbol{\phi}\boldsymbol{U}^{\mathrm{T}})+\mathrm{Tr}\,(\boldsymbol{\omega}\boldsymbol{B}^{\mathrm{T}})+\mathrm{Tr}\,(\boldsymbol{\psi}\boldsymbol{V}^{\mathrm{T}})\end{aligned} \tag{14}$$

Next, we can derive the partial derivatives of $\mathcal{L}$ with respect to $\boldsymbol{U}$, $\boldsymbol{B}$, and $\boldsymbol{V}$ as follows:

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\boldsymbol{U}}=&-2\boldsymbol{X}\boldsymbol{D}\boldsymbol{V}\boldsymbol{B}^{\mathrm{T}}+2\boldsymbol{U}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{V}\boldsymbol{B}^{\mathrm{T}}+\\ &\boldsymbol{\phi}+4\beta\,(\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}-\boldsymbol{U})+2\alpha\boldsymbol{L}_{\mathrm{hyper}}^{\boldsymbol{U}}\end{aligned} \tag{15}$$

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{B}}=-2\boldsymbol{U}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{D}\boldsymbol{V}+\boldsymbol{\omega}+2\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{V} \tag{16}$$

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\boldsymbol{V}}=&-2\boldsymbol{D}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{B}+2\boldsymbol{D}\boldsymbol{V}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{B}+\\ &\boldsymbol{\psi}+4\beta\,(\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V}-\boldsymbol{V})+2\alpha\boldsymbol{L}_{\mathrm{hyper}}^{\boldsymbol{V}}\end{aligned} \tag{17}$$

Given that $\boldsymbol{L}_{\mathrm{hyper}}$ is not a non-negative matrix, we refer to $\boldsymbol{L}=\boldsymbol{D}-\boldsymbol{S}$, let $\boldsymbol{L}_{\mathrm{hyper}}=\boldsymbol{D}_{\mathrm{hyper}}-\boldsymbol{S}_{\mathrm{hyper}}$, where $\boldsymbol{D}_{\mathrm{hyper}}$ and $\boldsymbol{S}_{\mathrm{hyper}}$ are non-negative matrices. Using the KKT condition, the multiplicative update rules of RDHNMTF are obtained as follows:

$$u_{jk}\leftarrow u_{jk}\frac{(\boldsymbol{X}\boldsymbol{D}\boldsymbol{V}\boldsymbol{B}^{\mathrm{T}}+\alpha\boldsymbol{S}_{\mathrm{hyper}}^{\boldsymbol{U}}\boldsymbol{U}+2\beta\boldsymbol{U})_{jk}}{(\boldsymbol{U}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{V}\boldsymbol{B}^{\mathrm{T}}+\alpha\boldsymbol{D}_{\mathrm{hyper}}^{\boldsymbol{U}}\boldsymbol{U}+2\beta\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U})_{jk}} \tag{18}$$

$$b_{kk'}\leftarrow b_{kk'}\frac{(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{X}\boldsymbol{D}\boldsymbol{V})_{kk'}}{(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{D}\boldsymbol{V})_{kk'}} \tag{19}$$

$$v_{ik}\leftarrow v_{ik}\frac{(\boldsymbol{D}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{B}+\alpha\boldsymbol{S}_{\mathrm{hyper}}^{\boldsymbol{V}}\boldsymbol{V}+2\beta\boldsymbol{V})_{ik}}{(\boldsymbol{D}\boldsymbol{V}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\boldsymbol{B}+\alpha\boldsymbol{D}_{\mathrm{hyper}}^{\boldsymbol{V}}\boldsymbol{V}+2\beta\boldsymbol{V}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V})_{ik}} \tag{20}$$

The overall process of the RDHNMTF algorithm is presented in Algorlthm 1.

3.4 Computational complexity analysis

In this subsection, the computational complexity analysis for the mentioned algorithms are presented. It is widely adopted to analyze the computational complexity with big O notation. However, big O notation is not precise enough to differentiate the computational complexity between RDHNMTF and its comparison algorithms. To enhance the accuracy of computational complexity analysis, three arithmetic operations, namely addition, multiplication, and division, are utilized. Table 1 displays the computational complexity of each algorithm during each iteration. From Table 1, it can be observed that NMTF based algorithms (i.e., DNMTF and CNMTF) require more arithmetic operations than NMF based algorithms (i.e., DNMF and CHNMF) due to the extra decomposed matrix and orthogonality constraints. When assessed using big O notation, the overall costs for all algorithms amount to $O(mnk)$. In addition, RDHNMTF requires the feature hyper-graph and

Algorithm 1 Iterative algorithm for RDHNMTF

Input: Data matrix $\boldsymbol{X}\in\mathbf{R}^{m\times n}$, parameter α, β, and k

Output: Decomposed matrices $\boldsymbol{U}\in\mathbf{R}^{m\times k}$ and $\boldsymbol{V}\in\mathbf{R}^{n\times k}$

1: Initialize matrices $\boldsymbol{U}\geqslant 0$, $\boldsymbol{B}\geqslant 0$, and $\boldsymbol{V}\geqslant 0$ randomly;
2: Repeat:
3: Calculate matrix $\boldsymbol{D}$ based on Eq. (12);
4: Update $\boldsymbol{U}$ by Formula (18);
5: Update $\boldsymbol{B}$ by Formula (19);
6: Update $\boldsymbol{V}$ by Formula (20);
7: Until converges;
8: Return decomposed matrices $\boldsymbol{V}$;
9: Utilize the K-means to obtain learned cluster label information for matrices $\boldsymbol{V}$.

Table 1 Computational complexity comparison.

Method	Floating point addition	Floating point multiplication	Floating point division	Computational complexity
RDHNMTF	$3mnk+8mk^2+5nk^2+6k^3+5mk+11nk+(m+n)pk$	$3mnk+8mk^2+5nk^2+6k^3+2mk+8nk+k^2+(m+n)pk$	$mk+nk+k^2$	$O(mnk)$
HNMF	$2mnk+2mk^2+2nk^2+mk+3nk+mpk$	$2mnk+2mk^2+2nk^2+mk+2nk+npk$	$mk+nk$	$O(mnk)$
CNMTF	$5mnk+9mk^2+6nk^2+2k^3+11mk+5nk+(m+n)pk$	$5mnk+9mk^2+6nk^2+2k^3+8mk+2nk+k^2+(m+n)pK$	$mk+nk+k^2$	$O(mnk)$
CHNMF	$2mnk+2mk^2+2nk^2+4mk+3nk+npk$	$2mnk+2mk^2+2nk^2+5mk+2nk+npk$	$mk+nk$	$O(mnk)$
HGSNMF	$2mnk+2mk^2+2nk^2+3mk+3nk+mpk$	$2mnk+2mk^2+2nk^2+mk+3nk+npk$	$mk+nk$	$O(mnk)$
DNMF	$2mnk+2mk^2+2nk^2+3mk+3nk+(m+n)pk$	$2mnk+2mk^2+2nk^2+2mk+2nk+(m+n)pk$	$mk+nk$	$O(mnk)$
SHNMF-MCC	$2mnk+2mk^2+2nk^2+4mk+4nk+npk$	$2mnk+2mk^2+2nk^2+5mk+3nk+npk$	$mk+nk$	$O(mnk)$
RHNMF	$2mnk+2mk^2+2nk^2+7nk+npk$	$2mnk+2mk^2+2nk^2+mk+6nk+npp$	$mk+nk$	$O(mnk)$
RDGDNMF	$2mnk+2mk^2+2nk^2+3mk+3nk+(m+n)pk$	$2mnk+2mk^2+2nk^2+2mk+2nk+(m+n)pk$	$mk+nk$	$O(mnk)$
SDGNMF-BO	$2mnk+2mk^2+2nk^2+2k^3+4mk+3nk+(m+n)pk$	$2mnk+2mk^2+2nk^2+2k^3+mk+2nk+(m+n)pk$	$mk+nk+k^2$	$O(mnk)$

sample hyper-graph to capture the manifold structure, whose computational complexity are $O(mn^2)$ and $O(n^2m)$ respectively. The total computational complexity of RDHNMTF can be expressed as $O(mnk+mn^2+nm^2)$. We can draw the conclusion that RDHNMTF can achieve a better clustering performance without much increase in run time, the reasons are as follows:

(1) Given that $k << \min(m, n)$, term mnk dominates the computational analysis and term mnk is considered to estimate the computational complexity. It is evident in Table 1 that the coefficient of mnk terms in RDHNMTF is 3, which is not large, as the the coefficient of mnk terms in comparison algorithms varies from 2 to 5.

(2) Some NMF methods, such as CHNMF and CNMTF, require exponential operations when calculating similarity, whereas RDHNMTF does not require exponential arithmetic operation, which also saves run time.

(3) As feature hyper-graph and sample hyper-graph are constructed only once in the initial phase, the extra run time to construct hyper-graph is not large.

3.5 Convergence analysis

The convergence of RDHNMTF is analyzed in this subsection and the following theorem is derived.

Theorem 2 The optimization Formula (11) is monotonically decreasing under the update rules of Formulas (18)–(20).

The detailed proof of Theorem 2 is given in Appendix B.

4 Experiment

RDHNMTF algorithm is compared with ten state-of-the-art algorithms across seven datasets (i.e., COIL20, ORL[†], YALE[‡], PIE[§], MSRA25[¶], PENDIGITS[©], and Mpeg7) to demonstrate the efficacy and robustness of the introduced methods.

4.1 Experiments setting

4.1.1 Datasets description

A variety of datasets are utilized in the experiment, Table 2 summarizes the details of the used datasets.

4.1.2 Comparison algorithms

RDHNMTF is compared with ten state-of-the-art algorithms to demonstrate the superior performance and robustness of RDHNMTF.

(1) GNMF[29]. Graph regularized non-negative matrix factorization adopts graph regularization with the aim of constructing a affinity graph to capture the geometric information of the datasets.

(2) HNMF[35]. Hyper-graph regularized non-negative matrix factorization constructs a hyper-graph to learn

[†] http://www.cl.cam.ac.uk/Research/DTG/attarchive:pub/data/att_faces.tar.Z

[‡] http://cvc.cs.yale.edu/cvc/projects/yalefaces/yalefaces.html

[§] https://www.ri.cmu.edu/publications/the-cmu-pose-illumination-and-expression-pie-database-of-human-faces/

[¶] http://www.escience.cn/people/fpnie/index.html

[©] http://archive.ics.uci.edu/ml/datasets/pen-based+recognition+of+handwritten+digits

Table 2 Description of the used datasets.

Dataset	Number of samples	Number of features	Number of classes	Data type
COIL20	1440	1024	20	Object image
ORL	400	10 304	40	Face image
YALE	165	1024	15	Face image
PIE	1166	1024	53	Face image
MSRA25	1799	256	12	Face image
PENDIGITS	10 992	16	10	Handwritten digit
Mpeg7	1400	6000	70	Shape image

geometrical structure.

(3) CNMTF[41]. Correntropy based orthogonal non-negative matrix tri-factorizaion uses correntropy to measure the similarity.

(4) CHNMF[13]. Correntropy based hyper-graph regularized non-negative matrix factorization integrates hyper-graph regularization and correntropy method into NMF algorithm.

(5) HGSNMF[42]. Algorithm called hyper-graph regularized l_p smooth NMF adds hyper-graph regularization terms and l_p smoothing constraint term into conventional NMTF.

(6) DNMF[30]. Graph dual regularization non-negative matrix factorization considers the data manifolds and feature manifolds simultaneously.

(7) SHNMF-MCC[43]. Maximizing correntropy is utilized in sparse hyper-graph regularized non-negative matrix factorization to measure similarity, while hyper-graph regularization and sparse constraint are employed to enhance clustering performance.

(8) RHNMF[14]. Robust hyper-graph regularized non-negative matrix factorization uses $l_{2,1}$-norm instead of Euclidean distance to calculate the difference and it absorbs hyper-graph regularization.

(9) SDGNMF-BO[44]. SDGNMF-BO is a semi-supervised dual graph regularized NMF with biorthogonal constraints. In our unsupervised task experiment, the label matrix of SDGNMF-BO is fixed as the identity matrix.

(10) RDGDNMF[45]. RDGDNMF is a robust supervised NMF algorithm with dual graph regularization. The coefficient of the label information terms is set to 0 for the unsupervised task.

4.1.3 Evaluation metrics

With the aim of evaluating the performance of the mentioned methods in a fair way, evaluation metrics Purity (PUR), Normalized Mutual Information (NMI), and Accuracy (ACC) are introduced.

Purity measures the extent to which samples within each cluster belong to the same class, purity is defined in the following manner:

$$\mathrm{PUR} = \frac{1}{N}\sum_{j=1}^{N}\max\left(n_i^j\right) \tag{21}$$

where n_i^j represents the sample in cluster i that also is a member of original class j, N denotes the total number of samples in the dataset.

The NMI evaluates the common information shared by two clusters and gauges the degree of consensus between them. Given the ground truth cluster C and the cluster obtained from the clustering algorithm result $\overline{C}$, NMI is defined as follows:

$$\mathrm{NMI} = \frac{\mathrm{MI}\left(C, \overline{C}\right)}{\max\left(H\left(C\right), H\left(\overline{C}\right)\right)} \tag{22}$$

where $\mathrm{MI}\left(C, \overline{C}\right)$ represents the mutual information between cluster C and $\overline{C}$, and $H\left(C\right)$ denotes the entropies of C.

The accuracy explores the one-to-one relationship between the ground truth label l_i and the learned cluster label r_i, and calculates the percentage of the correct label. Accuracy is defined int the following manner:

$$\mathrm{ACC} = \frac{\sum_{n=1}^{N}\delta\left(l_n, \mathrm{map}\left(r_n\right)\right)}{n} \tag{23}$$

where $\delta\left(x, y\right) = 1$ if $x = y$ and $\delta\left(x, y\right) = 0$ otherwise. $\mathrm{map}\left(\cdot\right)$ is a mapping function which can find a optimal match between l_n and r_n. In our experiments, Hungarian algorithm is adopted[46], which is a effective mapping function to solve the mapping problem. For the mentioned evaluation metrics PUR, NMI, and ACC, a higher value indicates better performance of the clustering algorithm.

4.1.4 Experimental setup

In this subsection, the experiments setup is discussed comprehensively. The dimension k of the $\boldsymbol{B}^{k\times k}$ is set to match the real class in the datasets. In the graph regularized algorithms and hyper-graph regularized

algorithms, we utilize k-nearest neighbor method to construct the graph and the hyper-graph. The value of nearest neighbor p in k-nearest neighbor method is empirically adjusted to be 5. After experimental verification, dual hyper-graph regularization parameter α of the proposed algorithm is set to 100 and orthogonality parameter β is adjusted to be 0. 01. The parameters regarding the different compared algorithm are set as default. Matrices $\boldsymbol{U}$, $\boldsymbol{B}$, and $\boldsymbol{V}$ are randomly initialized, and each approach is executed 20 times on different datasets.

4.2 Experiment results

Tables 3–5 show the accuracy, purity, and NMI of each algorithms on different datasets, respectively. The optimal values of the experiments are marked with boldface. Figure 3 shows the convergent results of RDHNMTF on seven datasets, which confirms the convergence proof in Appendix B. Overall speaking, RDHNMTF has a better clustering performance than other state-of-the-art algorithms. Specifically, three critical factors are summarized and explained as follows:

(1) Algorithms that combine hyper-graph regularization (e.g., HNMF, CHNMF, and RHNMF) outperform algorithms combine conventional graph regularization (e.g., GNMF and DNMF). Moreover, the proposed RDHNMTF performs better than other single hyper-graph regularized algorithms (e.g., CHNMF and RHNMF) and graph regularized algorithms (e.g., GNMF). The above results show that the dual hyper-graph regularization is superior to single hyper-graph

Table 3 Accuracy (ACC ± standard deviation) on different datasets.

(%)

Algorithm	COIL	ORL	YALE	PIE	MSRA25	PENDIGITS	Mpeg7
GNMF[29]	73.64±3.04	67.50±2.31	43.07±1.77	73.82±2.21	58.68±2.36	72.51±4.54	53.01±1.21
HNMF[35]	76.30±3.35	67.38±2.13	43.88±2.53	77.24±2.07	56.12±2.41	74.00±4.61	53.77±1.24
CNMTF[41]	66.32±2.51	64.11±1.68	45.51±3.47	67.75±1.57	**59.66±3.27**	71.53±4.47	47.58±2.29
CHNMF[13]	75.05±3.18	68.59±2.10	45.69±2.58	73.13±1.66	58.10±2.48	71.43±7.06	51.37±2.17
HGSNMFF[42]	76.84±3.22	72.17±2.53	45.45±2.14	76.63±1.76	57.01±2.19	73.37±3.87	55.15±0.86
DNMF[30]	66.11±2.65	68.15±1.01	43.55±2.44	77.47±1.50	57.98±2.27	74.23±3.84	52.61±1.76
SHNMF-MCC[43]	69.87±2.50	71.85±1.91	46.97±2.96	65.05±1.32	58.27±1.86	70.69±7.75	54.51±1.53
RHNMF[14]	77.34±2.57	70.36±1.94	44.85±2.15	71.27±3.01	53.93±2.18	70.51±5.43	54.87±1.03
RDGDNMF[45]	66.04±1.19	63.97±2.45	45.57±2.89	75.98±1.85	56.89±2.65	73.66±2.44	56.02±0.77
SDGNMF-BO[44]	71.09±2.00	62.07±2.55	40.06±2.08	75.19±2.02	56.08±1.75	70.49±4.70	51.97±0.45
RDHNMTF	**78.12±1.85**	**72.29±2.66**	**47.96±2.08**	**80.85±2.10**	59.04±3.80	75.61±3.11	**58.41+0.40**

Note: The best results are in bold.

Table 4 Purity (PUR± standard deviation) on different datasets.

(%)

Algorithm	COIL	ORL	YALE	PIE	MSRA25	PENDIGITS	Mpeg7
GNMF[29]	75.97±2.22	71.12±2.03	43.99±1.88	76.71±1.53	61.56±0.99	74.12±2.65	56.77±0.78
HNMF[35]	80.08±2.45	71.01±1.45	45.46±2.16	80.83±1.92	58.53±2.80	75.22±2.91	58.28±0.80
CNMTF[41]	69.30±2.24	67.19±1.48	46.54±3.29	71.69±1.19	**63.26±1.94**	73.72±3.23	51.12±1.34
CHNMF[13]	77.73±2.58	73.11±1.96	47.04±2.31	76.68±1.15	60.35±3.96	73.21±5.01	54.92±1.84
HGSNMF[42]	79.16±1.78	75.40±1.80	46.12±2.18	80.36±1.30	60.14±2.33	74.46±2.28	59.27±0.89
DNMF[30]	68.28±2.27	71.65±1.28	44.78±2.11	81.35±1.13	62.01±3.36	74.76±2.86	56.87±1.20
SHNMF-MCC[43]	71.50±1.76	75.55±0.91	47.818±2.42	68.11±0.88	60.48±1.29	72.37±5.71	58.45±0.63
RHNMF[14]	**81.39±1.67**	74.02±1.59	47.23±1.97	75.27±2.46	56.88±1.61	72.48±4.38	58.57±0.70
RDGDNMF[45]	69.47±1.33	67.5±1.58	46.97±2.38	79.75±1.92	60.92±1.42	70.00±3.50	58.84±0.37
SDGNMF-BO[44]	72.81±2.02	66.97±1.93	42.12±2.05	79.14±1.46	60.61±0.57	72.26±3.50	55.62±0.35
RDHNMTF	80.31±1.24	**75.98±1.95**	**48.62±2.46**	**83.71±1.15**	60.24±1.78	**75.65±2.03**	**60.41±0.36**

Note: The best results are in bold.

Table 5 NMI (NMI± standard deviation) on different datasets, the best results are highlighted in bold.

(%)

Algorithm	COIL	ORL	YALE	PIE	MSRA25	PENDIGITS	Mpeg7
GNMF[29]	83.16±1.70	84.03±1.31	47.81±1.47	84.46±0.90	73.11±0.79	69.92±1.23	73.03±0.61
HNMF[35]	87.23±1.81	83.60±0.69	48.95±1.75	87.98±0.91	63.53±3.445	70.01±2.10	73.34±0.539
CNMTF[41]	74.83±2.95	81.43±0.68	50.45±2.99	82.04±0.92	**74.52±1.82**	**70.23±2.61**	68.81±0.49
CHNMF[13]	83.93±2.61	85.41±1.41	50.1±1.83	85.03±0.77	66.48±3.06	67.22±3.65	72.22±1.24
HGSNMF[42]	**87.82±0.75**	**89.41±0.77**	47.78±2.33	88.02±0.74	68.75±2.41	69.92±1.42	74.00±0.39
DNMF[30]	76.46±1.12	83.73±1.13	48.18±1.71	88.85±1.08	74.11±2.23	68.37±2.93	72.83±0.52
SHNMF-MCC[43]	78.69±1.19	89.69±0.37	51.61±2.41	78.69±0.68	66.47±1.97	67.90±2.38	73.87±0.06
RHNMF[14]	88.84±1.28	85.76±0.87	48.62±2.46	84.67±1.41	61.21±1.61	66.96±4.36	73.81±0.46
RDGDNMF[45]	78.21±1.22	81.12±1.02	50.27±2.30	87.89±1.73	73.55±1.28	65.47±0.96	74.98±0.32
SDGNMF-BO[44]	79.91±1.39	80.30±1.20	45.98±2.44	87.35±0.87	73.14±0.72	67.78±2.34	71.87±0.15
RDHNMTF	84.30±1.21	86.95±0.58	**51.65±2.04**	**89.64±0.82**	65.30±2.59	69.69±2.15	**76.24±0.38**

Note: The best results are in bold.

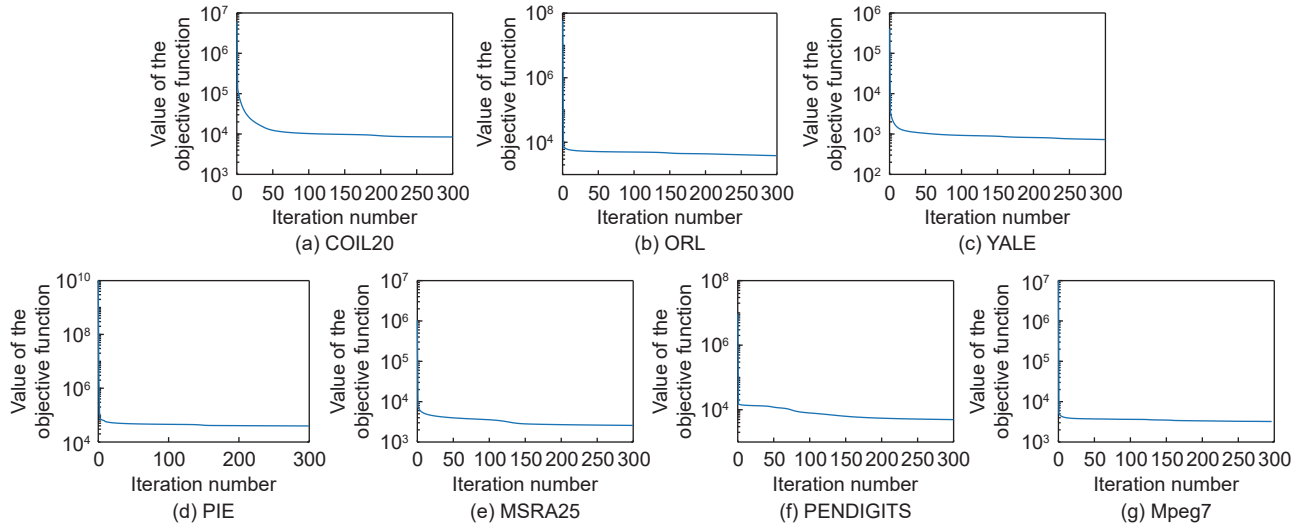

Fig. 3 Convergence results on seven datasets.

regularization and single graph regularization. The main reason is that dual hyper-graph regularization contributes to obtaining a better geometric information in sample manifold and feature manifold.

(2) $l_{2,1}$-norm based algorithms (e.g., RHNMF and RDHNMTF) have a better performance than those using Euclidean distance (e.g., NMF, GNMF, and DNMF) to measure the reconstruction error. It indicates that $l_{2,1}$ norm helps alleviate the influence of noise and outliers and thus improve the clustering performance.

(3) The dual graph regularized algorithms without NMTF framework (e.g., DNMF and SDGNMF-BO) performs similarly to the single-graph regularized algorithm (GNMF), whereas RDHNMTF outperforms other algorithms with single hyper-graph regularization, which indicates that the NMTF framework helps the proposed algorithm to obtain wider degrees of freedom. The adopted orthogonality condition contributes to find unique solution. Therefore, the used orthogonal NMTF framework improves the clustering performance of RDHNMTF.

4.3 Robustness analysis

To demonstrate the robustness of RDHNMTF, we further conducted additional experiments on contaminated datasets using the aforementioned state-of-the-art algorithms. The contaminated datasets consist of ORL, MSRA25, and PIE datasets with four different noises, i.e., Salt & Pepper noise, Block noise,

Speckle noise, and Gaussian noise. Specifically, we add Salt & Pepper noise and Block noise to ORL dataset, Speckle noise to MSRA25 dataset, and Gaussian noise to PIE dataset. The intensity of several kinds of noises applied is gradually increased. Figure 4 shows the examples from the contaminated datasets. The clustering results on contaminated datasets are demonstrated in Figs. 5 and 6. The clustering results suggest that the robust M-estimators based NMF (e.g., RHNMF, CNMF, CHNM, and RDHNMTF) methods can mitigate the adverse impact of noise and outliers. Additionally, the proposed RDHNMTF algorithm outperforms other state-of-the-art algorithms in most cases. This is due to $l_{2,1}$-norm reducing the influence of noise and outliers while orthogonality constrains and dual hyper-graph regularization improve the clustering performance.

4.4 Parameter selection

Parameter selection has a significant impact on the experimental results. The main parameters in RDHNMTF are α and β. Relatively small parameter values result in reduced impact of the dual hyper-graph regularization and orthogonality constraints. Also, comparatively large parameters may decrease the impact of rest terms in the optimization function and deteriorate the clustering performance. Unfortunately, it is hard to find optimal values for α and β. Hence, grid searching method is adopted to find relatively ideal values for α and β, which range in $\{10^{-3}, 10^{-2}, \ldots, 1, \ldots 10^{3}\}$. Although gird searching method can not find the optimal parameters for RDHNMTF, RDHNMTF with parameters found by gird searching can achieve a good clustering performance. Figure 7 demonstrates that DHNMTF has a ideal performance when $\alpha = 100$ and $\beta = 0.01$. Hence, 100 and 0.01 are set as the default values of α and β, respectively.

4.5 Ablation study

Ablation study is conducted to illustrate the benefits of dual hyper-graph regularization and orthogonality constraints. The results of ablation study are recorded in Table 6 for four cases: (1) without dual hyper-graph regularization and orthogonality constraint; (2) with only dual hyper-graph regularization; (3) with only orthogonality constraints; (4) with dual hyper-graph regularization and orthogonality constraints. To realize the above four cases, α and β are tuend respectively as: (1) 0 and 0; (2) 100 and 0; (3) 0 and 0.01; and (4) 100 and 0.01. It is evident that the performance of RDHNMTF with orthogonality constraints alone is significantly better than the performance of RDHNMTF with dual hyper-graph regularization alone (e.g., COIL20, YALE, PIE, and MSRA25). Additionally, when adding dual hyper-graph regularization, the performance of DHNMTF without dual orthogonality dramatically degrades on all datasets, whereas the performance of orthogonality constrained DHNMTF is greatly improved. The main reason to explain the experiment results is that the orthogonality constraints enables the proposed

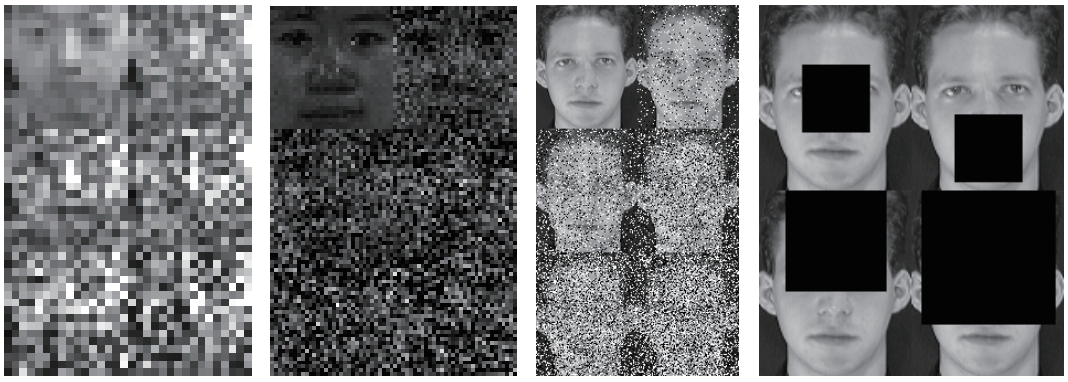

Fig. 4 Sample images of different datasets in various noisy conditions. (a) Illustrative image extracted from the MSRA25 dataset along with its corrupted version by Speckle noise having a mean of 0 and variances V = 0.04, 0.08, 0. 12, 0. 16, and 0.20; (b) Illustrative image extracted from the PIE dataset along with its corrupted version by Gaussian noise having a mean of 0 and variance V = 0.01, 0.02, 0.03, 0.04, 0.05; (c) Illustrative image from the ORL dataset along with its corrupted version by Salt & Pepper noise having a density D = 10%, 20%, 30%, 40%, and 50%; and (d) Illustrative image from the ORL dataset along with its corrupted version by $a \times a$-blocks noise with a = 25, 30, 35, and 40.

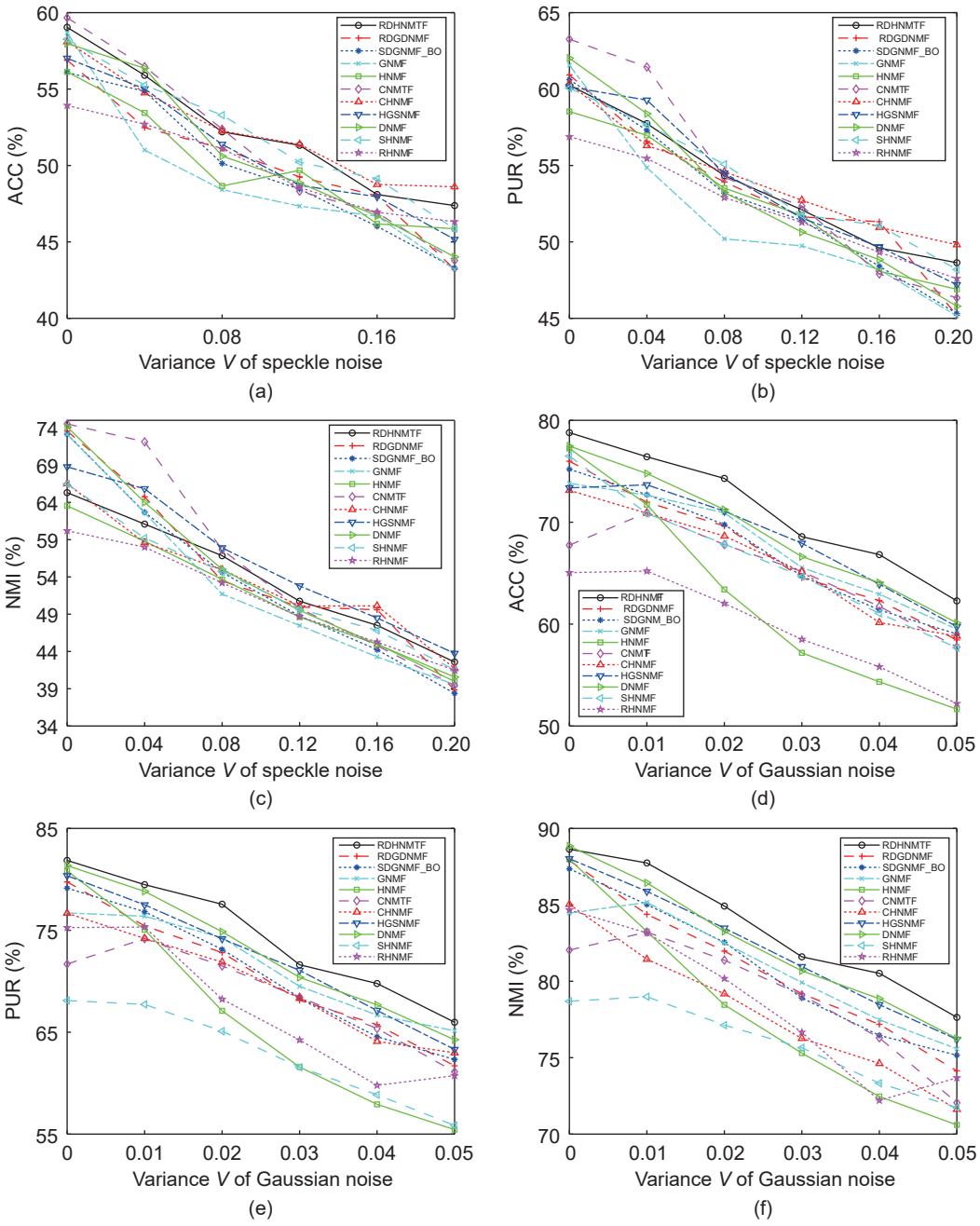

Fig. 5 Clustering performance on MSRA25 and PIE datasets with different noise types: (a)–(c) show the clustering performance on MSRA25 dataset corrupted by Speckle noise, while (d)–(f) show the clustering performance on PIE dataset corrupted by Gaussian noise.

algorithms to have a distinct data representation. The dual hyper-graph regularization without orthogonality constraints make the learned sample manifold and feature manifold lose uniqueness and thus downgrade the performance. Ablation experiments suggest that the combination of dual hyper-graph regularization and orthogonality constraints is beneficial to the RDHNMTF model.

5 Conclusion and Feature work

In this paper, the robustness of non-negative matrix tri-factorization is enhanced by introducing $l_{2,1}$-norm to measure the construction error. Furthermore, hyper-graph is utilized to discover the high order geometric information for improving clustering performance. To address the formulated optimization problem, an alternating optimization algorithm is designed and

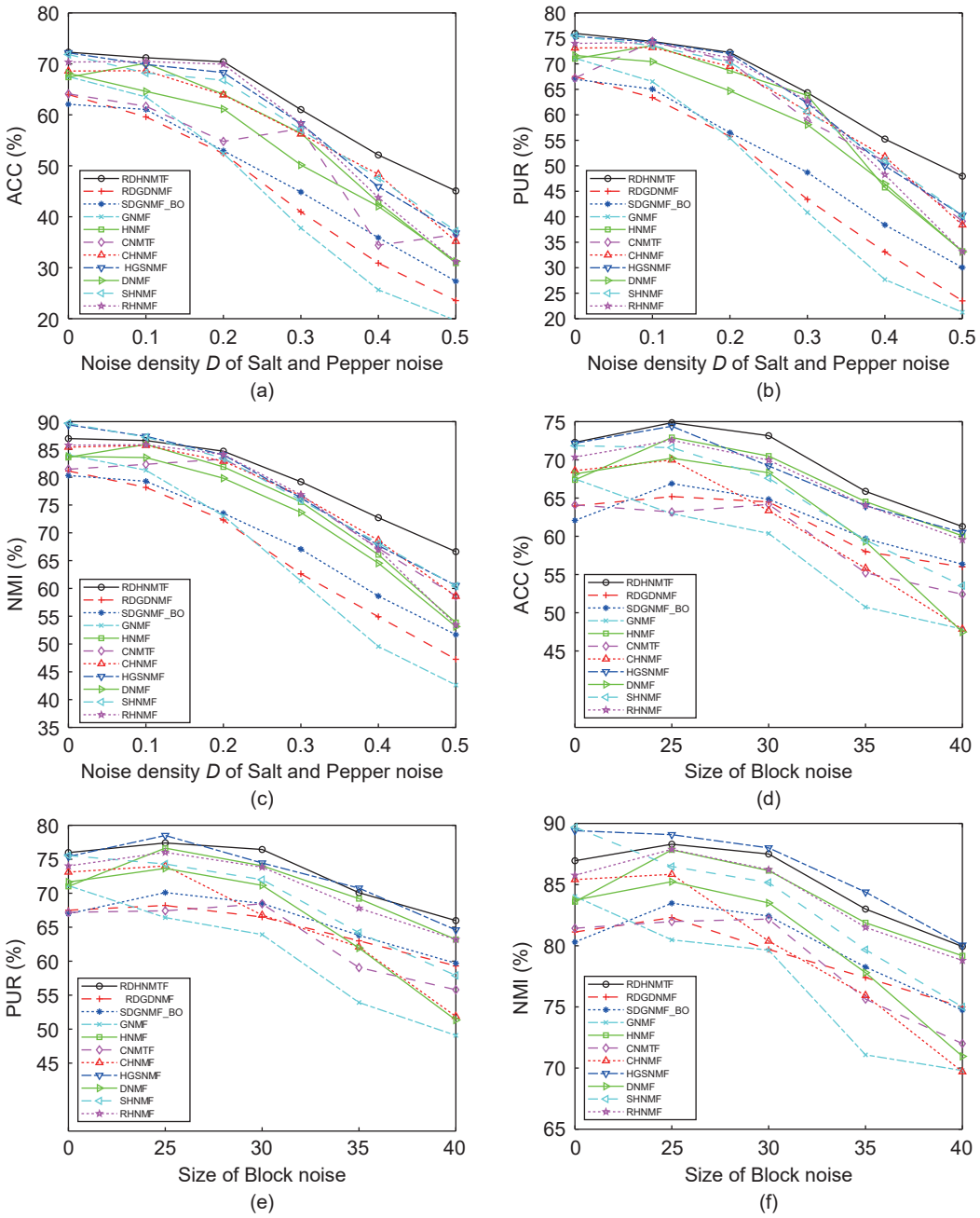

Fig. 6 Clustering performance on ORL dataset with different noise types: (a)–(c) show the clustering performance on ORL dataset corrupted by Salt and Pepper noise, while subfigures (d)–(f) show the clustering performance on ORL dataset corrupted by Block noise.

analyzed. Abundant experiments show the superiority of the proposed algorithm. Although RDHNMTF has good clustering performance, it also has rooms for improvements. First, the parameter is set empirically, thus a algorithm to find optimal parameters is needed. Second, RDHNMTF can be improved to solve multi-view clustering problem instead of single-view clustering problem.

Acknowledgment

The work was supported by the National Natural Science Foundation of China (No. 62003281), the Natural Science Foundation of Chongqing, China (No. cstc2021jcyjmsxmX1169), and the Science and Technology Research Program of Chongqing Municipal Education Commission (No. KJQN202200207).

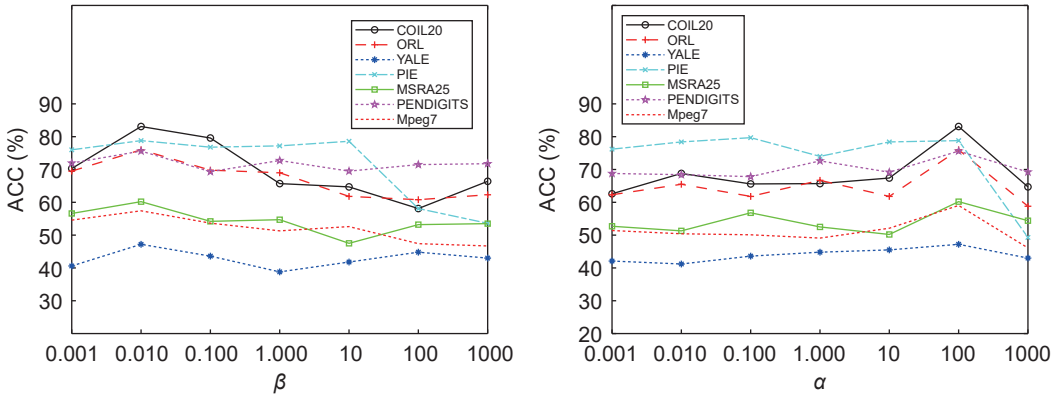

Fig. 7 Performance of RDHNMTF with the regularization parameters.

Table 6 Ablation experiments of RDHNMTF.

(%)

Setting	Dataset	ACC	PUR	NMI
RDHNMTF without the use of dual hyper-graph regularization and constraints for orthogonality	COIL20	64.17±1.64	67.44±1.91	76.18±1.41
	ORL	65.00±2.93	68.43±2.80	81.63±1.77
	YALE	43.5±2.49	44.97±2.49	48.74±1.71
	PIE	76.03±1.86	80.13±1.86	88.48±1.25
	MSRA25	52.26±2.22	54.97±2.37	57.08±2.64
	PENDIGITS	71.96±5.77	66.80±4.79	70.92±6.46
	Mpeg7	50.27±0.79	53.75±0.46	70.19±0.51
RDHNMTF solely utilizing dual hyper-graph regularization	COIL20	50.94±2.68	55.58±0.98	64.26±0.96
	ORL	26.57±3.51	30.43±3.36	51.91±4.54
	YALE	21.02±0.40	22.88±0.62	27.17±0.50
	PIE	22.17±2.55	27.93±2.26	45.23±3.59
	MSRA25	38.90±7.23	42.51±6.80	46.34±9.44
	PENDIGITS	70.28±3.12	66.78±2.91	68.22±3.63
	Mpeg7	22.20±0.47	25.31±0.71	50.15±0.81
RDHNMTF solely utilizing orthogonality constraints	COIL20	66.05±1.52	68.28±1.29	76.06±0.90
	ORL	63.84±1.65	67.11±1.87	81.42±1.15
	YALE	45.89±2.02	46.62±2.17	49.96±1.58
	PIE	77.20±1.78	80.56±1.41	88.30±0.71
	MSRA25	54.53±1.63	56.67±2.64	59.73±3.44
	PENDIGITS	71.73±3.33	66.68±2.56	69.91±4.40
	Mpeg7	49.38±1.40	59.41±0.36	56.97±0.76
RDHNMTF with the use of dual hyper-graph regularization and orthogonality constraints	COIL20	78.12±1.85	80.31±1.24	84.30±1.21
	ORL	72.29±2.66	75.98±1.95	86.95±0.58
	YALE	47.96±2.08	48.62±2.46	51.65±2.04
	PIE	80.85±2.10	83.71±1.15	89.64±0.82
	MSRA25	59.04±3.80	60.24±1.78	65.30±2.59
	PENDIGITS	75.61±3.11	75.65±2.03	69.69±2.15
	Mpeg7	58.41+0.40	60.41±0.36	76.24±0.38

Appendix

A Proof of Theorem 1

To prove Theorem 1, Lemmas 1–3 are needed to be proved firstly. We define $\boldsymbol{U}^{\mathrm{T}}$ as the value of $\boldsymbol{U}$ at the t-th iteration, so are $\boldsymbol{B}^{\mathrm{T}}$ and $\boldsymbol{V}^{\mathrm{T}}$.

Lemma 1

$$\begin{aligned}&\left\|\boldsymbol{X}-\boldsymbol{U}^{t+1}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}}\right\|_{2,1}-\left\|\boldsymbol{X}-\boldsymbol{U}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}}\right\|_{2,1}\leqslant\\&\frac{1}{2}\left(\mathrm{Tr}\left((\boldsymbol{X}-\boldsymbol{U}^{t+1}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}})\boldsymbol{D}(\boldsymbol{X}-\boldsymbol{U}^{t+1}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}})^{\mathrm{T}}\right)-\right.\\&\left.\mathrm{Tr}\left((\boldsymbol{X}-\boldsymbol{U}^{t}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}})\boldsymbol{D}(\boldsymbol{X}-\boldsymbol{U}^{t}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}})^{\mathrm{T}}\right)\right)\end{aligned}\tag{A1}$$

Proof Considering $\boldsymbol{D}_{ii}=\frac{1}{\|\boldsymbol{X}_i-(\boldsymbol{U}\boldsymbol{B}\boldsymbol{V})_i\|}$, the Left-Hand-Side (LHS) of Formula (A1) is

$$\begin{aligned}&\mathrm{LHS}=\sum_{i=1}^{n}\left(\left\|\boldsymbol{X}_i-(\boldsymbol{U}^{t+1}\boldsymbol{B}\boldsymbol{V})_i\right\|-\left\|\boldsymbol{X}_i-(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{V})_i\right\|\right)=\\&\sum_{i=1}^{n}\left(\left\|\boldsymbol{X}_i-(\boldsymbol{U}^{t+1}\boldsymbol{B}\boldsymbol{V})_i\right\|-\frac{1}{\boldsymbol{D}_{ii}}\right)\end{aligned}\tag{A2}$$

The Right-Hand-Side (RHS) of Formula (A1) is

$$\begin{aligned}&\mathrm{RHS}=\frac{1}{2}\sum_{i=1}^{n}\left(\left\|\boldsymbol{X}_i-(\boldsymbol{U}^{t+1}\boldsymbol{B}\boldsymbol{V})_i\right\|^2\boldsymbol{D}_{ii}-\left\|\boldsymbol{X}_i-(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{V})_i\right\|^2\boldsymbol{D}_{ii}\right)=\\&\frac{1}{2}\sum_{i=1}^{n}\left(\left\|\boldsymbol{X}_i-(\boldsymbol{U}^{t+1}\boldsymbol{B}\boldsymbol{V})_i\right\|^2\boldsymbol{D}_{ii}-\frac{1}{\boldsymbol{D}_{ii}}\right)\end{aligned}\tag{A3}$$

Hence, we derive the following equations:

$$\begin{aligned}&\mathrm{LHS}-\mathrm{RHS}=\\&\sum_{i=1}^{n}\left(\left\|\boldsymbol{X}_i-(\boldsymbol{U}^{t+1}\boldsymbol{B}\boldsymbol{V})_i\right\|-\frac{1}{2}\left\|\boldsymbol{X}_i-(\boldsymbol{U}^{t+1}\boldsymbol{B}\boldsymbol{V})_i\right\|^2\boldsymbol{D}_{ii}-\frac{1}{2\boldsymbol{D}_{ii}}\right)=\\&\sum_{i=1}^{n}\frac{\boldsymbol{D}_{ii}}{2}\left(\frac{2\left\|\boldsymbol{X}_i-(\boldsymbol{U}^{t+1}\boldsymbol{B}\boldsymbol{V})_i\right\|}{\boldsymbol{D}_{ii}}-\left\|\boldsymbol{X}_i-(\boldsymbol{U}^{t+1}\boldsymbol{B}\boldsymbol{V})_i\right\|^2-\frac{1}{\boldsymbol{D}_{ii}^2}\right)=\\&\sum_{i=1}^{n}\frac{-\boldsymbol{D}_{ii}}{2}\left(\left\|\boldsymbol{X}_i-(\boldsymbol{U}^{t+1}\boldsymbol{B}\boldsymbol{V})_i\right\|-\frac{1}{\boldsymbol{D}_{ii}}\right)^2\leqslant 0\end{aligned}\tag{A4}$$

The proof of Lemma 1 is completed.

Lemma 2

$$\begin{aligned}&\left\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}^{t+1}\boldsymbol{V}^{\mathrm{T}}\right\|_{2,1}-\left\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{V}^{\mathrm{T}}\right\|_{2,1}\leqslant\\&\frac{1}{2}\left(\mathrm{Tr}\left((\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}^{t+1}\boldsymbol{V}^{\mathrm{T}})\boldsymbol{D}(\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}^{t+1}\boldsymbol{V}^{\mathrm{T}})^{\mathrm{T}}\right)-\right.\\&\left.\mathrm{Tr}\left((\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}^{t}\boldsymbol{V}^{\mathrm{T}})\boldsymbol{D}(\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}^{t}\boldsymbol{V}^{\mathrm{T}})^{\mathrm{T}}\right)\right)\end{aligned}\tag{A5}$$

The proof of Lemma 2 is similar to the proof of Lemma 1, so it is omitted.

Lemma 3

$$\begin{aligned}&\left\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}(\boldsymbol{V}^{\mathrm{T}})^{t+1}\right\|_{2,1}-\left\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}(\boldsymbol{V}^{\mathrm{T}})^{t}\right\|_{2,1}\leqslant\\&\frac{1}{2}\left(\mathrm{Tr}\left(\left(\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}(\boldsymbol{V}^{\mathrm{T}})^{t+1}\right)\boldsymbol{D}\left(\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}(\boldsymbol{V}^{\mathrm{T}})^{t+1}\right)^{\mathrm{T}}\right)\right.\\&\left.-\mathrm{Tr}\left(\left(\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}(\boldsymbol{V}^{\mathrm{T}})^{t}\right)\boldsymbol{D}\left(\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}(\boldsymbol{V}^{\mathrm{T}})^{t}\right)^{\mathrm{T}}\right)\right)\end{aligned}\tag{A6}$$

The proof of Lemma 3 is similar to the proof of Lemma 1, so it is omitted.

Proof of Theorem 1 Assume the proposed update rules are able to optimize the optimization Formula (11). Therefore, we can derive the following inequalities:

$$\begin{aligned}&\mathrm{Tr}\left((\boldsymbol{X}-\boldsymbol{U}^{t+1}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}})\boldsymbol{D}(\boldsymbol{X}-\boldsymbol{U}^{t+1}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}})^{\mathrm{T}}\right)\\&+2\mu\mathrm{Tr}\left((\boldsymbol{U}^{t+1})^{\mathrm{T}}\boldsymbol{L}_{\mathrm{hyper}}^{\boldsymbol{U}}\boldsymbol{U}^{t+1}\right)+2\nu\left\|(\boldsymbol{U}^{t+1})^{\mathrm{T}}\boldsymbol{U}^{t+1}-\boldsymbol{I}_K\right\|_{\mathrm{F}}^2\\&\leqslant\mathrm{Tr}\left((\boldsymbol{X}-\boldsymbol{U}^{t}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}})\boldsymbol{D}(\boldsymbol{X}-\boldsymbol{U}^{t}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}})^{\mathrm{T}}\right)\\&+2\mu\mathrm{Tr}\left((\boldsymbol{U}^{t})^{\mathrm{T}}\boldsymbol{L}_{\mathrm{hyper}}^{\boldsymbol{U}}\boldsymbol{U}\right)+2\nu\left\|(\boldsymbol{U}^{t})^{\mathrm{T}}\boldsymbol{U}^{t}-\boldsymbol{I}_K\right\|_{\mathrm{F}}^2\end{aligned}\tag{A7}$$

$$\begin{aligned}&\mathrm{Tr}\left((\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}^{t+1}\boldsymbol{V}^{\mathrm{T}})\boldsymbol{D}(\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}^{t+1}\boldsymbol{V}^{\mathrm{T}})^{\mathrm{T}}\right)\leqslant\\&\mathrm{Tr}\left((\boldsymbol{X}-\boldsymbol{U}^{t}\boldsymbol{B}^{t}\boldsymbol{V}^{\mathrm{T}})\boldsymbol{D}(\boldsymbol{X}-\boldsymbol{U}^{t}\boldsymbol{B}^{t}\boldsymbol{V}^{\mathrm{T}})^{\mathrm{T}}\right)\end{aligned}\tag{A8}$$

$$\begin{aligned}&\mathrm{Tr}\left(\left(\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}(\boldsymbol{V}^{t+1})^{\mathrm{T}}\right)\boldsymbol{D}\left(\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}(\boldsymbol{V}^{t+1})^{\mathrm{T}}\right)^{\mathrm{T}}\right)+\\&2\mu\mathrm{Tr}\left((\boldsymbol{V}^{t+1})^{\mathrm{T}}\boldsymbol{L}_{\mathrm{hyper}}^{\boldsymbol{V}}\boldsymbol{V}^{t+1}\right)+2\nu\left\|(\boldsymbol{V}^{t+1})^{\mathrm{T}}\boldsymbol{V}^{t+1}-\boldsymbol{I}_K\right\|_{\mathrm{F}}^2\leqslant\\&\mathrm{Tr}\left(\left(\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}(\boldsymbol{V}^{t})^{\mathrm{T}}\right)\boldsymbol{D}\left(\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}(\boldsymbol{V}^{t})^{\mathrm{T}}\right)^{\mathrm{T}}\right)+\\&2\mu\mathrm{Tr}\left((\boldsymbol{V}^{t})^{\mathrm{T}}\boldsymbol{L}_{\mathrm{hyper}}^{\boldsymbol{V}}\boldsymbol{V}^{t}\right)+2\nu\left\|(\boldsymbol{V}^{t})^{\mathrm{T}}\boldsymbol{V}^{t}-\boldsymbol{I}_K\right\|_{\mathrm{F}}^2\end{aligned}\tag{A9}$$

Substituting Formula (A7) into Lemma 1, Formula (A8) into Lemma 2, Formula (A9) into Lemma 3, we get

$$\begin{aligned}&\left\|\boldsymbol{X}-\boldsymbol{U}^{t+1}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}}\right\|_{2,1}+\mu\mathrm{Tr}\left((\boldsymbol{U}^{t+1})^{\mathrm{T}}\boldsymbol{L}_{\mathrm{hyper}}^{\boldsymbol{U}}\boldsymbol{U}^{t+1}\right)+\\&\nu\left\|(\boldsymbol{U}^{t+1})^{\mathrm{T}}\boldsymbol{U}^{t+1}-\boldsymbol{I}_K\right\|_{\mathrm{F}}^2\leqslant\\&\left\|\boldsymbol{X}-\boldsymbol{U}^{t}\boldsymbol{B}\boldsymbol{V}^{\mathrm{T}}\right\|_{2,1}+\mu\mathrm{Tr}\left((\boldsymbol{U}^{t})^{\mathrm{T}}\boldsymbol{L}_{\mathrm{hyper}}^{\boldsymbol{U}}\boldsymbol{U}^{t}\right)+\\&\nu\left\|(\boldsymbol{U}^{t})^{\mathrm{T}}\boldsymbol{U}^{t}-\boldsymbol{I}_K\right\|_{\mathrm{F}}^2\end{aligned}\tag{A10}$$

$$\left\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}^{t+1}\boldsymbol{V}^{\mathrm{T}}\right\|_{2,1}\leqslant\left\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{V}^{\mathrm{T}}\right\|_{2,1}\tag{A11}$$

$$\begin{aligned}&\left\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}(\boldsymbol{V}^{\mathrm{T}})^{t+1}\right\|_{2,1}+\mu\mathrm{Tr}\left((\boldsymbol{V}^{t+1})^{\mathrm{T}}\boldsymbol{L}_{\mathrm{hyper}}^{\boldsymbol{V}}\boldsymbol{V}^{t+1}\right)+\\&\nu\left\|(\boldsymbol{V}^{t+1})^{\mathrm{T}}\boldsymbol{V}^{t+1}-\boldsymbol{I}_K\right\|_{\mathrm{F}}^2\leqslant\\&\left\|\boldsymbol{X}-\boldsymbol{U}\boldsymbol{B}(\boldsymbol{V}^{\mathrm{T}})^{t+1}\right\|_{2,1}+\mu\mathrm{Tr}\left((\boldsymbol{V}^{t})^{\mathrm{T}}\boldsymbol{L}_{\mathrm{hyper}}^{\boldsymbol{V}}\boldsymbol{V}^{t}\right)+\\&\nu\left\|(\boldsymbol{V}^{t})^{\mathrm{T}}\boldsymbol{V}^{t}-\boldsymbol{I}_K\right\|_{\mathrm{F}}^2\end{aligned}\tag{A12}$$

Formulas (A10)–(A12) show that the update rules on optimization Formula (11) are able to decrease the optimization Formula (10). ■

B Proof of Theorem 2

To prove Theorem 2, we first introduce the definition of auxiliary function as follows:

Definiton $G(v,v')$ is an auxiliary function for $F(v)$ if it satisfys the following conditions[47]:

$$G(v,v') \geqslant F(v),\ G(v,v) = F(v) \tag{A13}$$

Lemma 4 If G is an auxiliary function for F, then F is non-increasing under the following updates steps from iteration t to t+1[47]:

$$v^{t+1} = \arg\min_{V} G(v,v') \tag{A14}$$

Proof

$$F(v^{t+1}) \leqslant G(v^{t+1},v^{\mathrm{T}}) \leqslant G(v^{\mathrm{T}},v^{\mathrm{T}}) = F(v^{\mathrm{T}}) \tag{A15}$$

From optimization Formula (12), we define O as follows:

$$\begin{aligned} O =& \mathrm{Tr}\left(\boldsymbol{XDX}^{\mathrm{T}}\right) - 2\mathrm{Tr}\left(\boldsymbol{UBV}^{\mathrm{T}}\boldsymbol{DX}^{\mathrm{T}}\right) + \\ & \mathrm{Tr}\left(\boldsymbol{UBV}^{\mathrm{T}}\boldsymbol{DVB}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\right) + \\ & \alpha\mathrm{Tr}\left(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{L}^{\boldsymbol{V}}_{\mathrm{hyper}}\boldsymbol{V}\right) + \alpha\mathrm{Tr}\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{L}^{\boldsymbol{U}}_{\mathrm{hyper}}\boldsymbol{U}\right) + \\ & \beta\mathrm{Tr}\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{UU}^{\mathrm{T}}\boldsymbol{U} - 2\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} + \boldsymbol{I}_k\right) + \\ & \beta\mathrm{Tr}\left(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{VV}^{\mathrm{T}}\boldsymbol{V} - 2\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V} + \boldsymbol{I}_k\right) \end{aligned} \tag{A16}$$

Next we will show the update rules which can minimize the auxiliary function G of the function O are exactly the update rules of Formulas (18)–(20).

We define $F_{jk}, F_{kk'}$, and $F_{ik'}$ to be the part of O which only involve the u_{jk} term, $b_{kk'}$ term, and v_{ik} term, respectively. The first and second partial derivatives of O with respect to $\boldsymbol{U},\boldsymbol{B}$, and $\boldsymbol{V}$ are defined as follows:

$$\begin{aligned} F'_{jk} = \left(\frac{\partial O}{\partial \boldsymbol{U}}\right)_{jk} =& \left(-2\boldsymbol{XDVB}^{\mathrm{T}} + 2\boldsymbol{UBV}^{\mathrm{T}}\boldsymbol{DVB}^{\mathrm{T}}\right)_{jk} + \\ & 2\alpha\left(\boldsymbol{L}^{\boldsymbol{U}}_{\mathrm{hyper}}\boldsymbol{U}\right)_{jk} + 4\beta\left(\boldsymbol{UU}^{\mathrm{T}}\boldsymbol{U} - \boldsymbol{U}\right)_{jk} \end{aligned} \tag{A17}$$

$$\begin{aligned} F''_{jk} =& \left(\frac{\partial^2 O}{\partial \boldsymbol{U}^2}\right)_{jk} = 2\left(\boldsymbol{BV}^{\mathrm{T}}\boldsymbol{DVB}^{\mathrm{T}}\right)_{kk} + 2\alpha\left(\boldsymbol{L}^{\boldsymbol{U}}_{\mathrm{hyper}}\right)_{jj} + \\ & 4\beta\left(\boldsymbol{J}^{jk}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U} + \boldsymbol{U}(\boldsymbol{J}^{jk})^{\mathrm{T}}\boldsymbol{U} + \boldsymbol{UU}^{\mathrm{T}}\boldsymbol{J}^{jk} - \boldsymbol{J}^{jk}\right)_{jk} \end{aligned} \tag{A18}$$

$$F'_{kk'} = \left(\frac{\partial O}{\partial \boldsymbol{B}}\right)_{kk'} = \left(2\boldsymbol{U}^{\mathrm{T}}\boldsymbol{UBV}^{\mathrm{T}}\boldsymbol{DV}\right)_{kk'} - \left(2\boldsymbol{U}^{\mathrm{T}}\boldsymbol{XDV}\right)_{kk'} \tag{A19}$$

$$F''_{kk'} = \left(\frac{\partial^2 O}{\partial \boldsymbol{B}^2}\right)_{kk'} = \left(2\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\right)_{kk}\left(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{DV}\right)_{k'k'} \tag{A20}$$

$$\begin{aligned} F'_{ik} = \left(\frac{\partial O}{\partial \boldsymbol{V}}\right)_{ik} =& \left(-2\boldsymbol{DX}^{\mathrm{T}}\boldsymbol{UB} + 2\boldsymbol{DVB}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{UB}\right)_{ik} + \\ & 2\alpha\left(\boldsymbol{L}^{\boldsymbol{V}}_{\mathrm{hyper}}\boldsymbol{V}\right)_{ik} + 4\beta\left(\boldsymbol{VV}^{\mathrm{T}}\boldsymbol{V} - \boldsymbol{V}\right)_{ik} \end{aligned} \tag{A21}$$

$$\begin{aligned} F''_{ik} =& \left(\frac{\partial^2 O}{\partial \boldsymbol{V}^2}\right)_{ik} = \\ & 2\boldsymbol{D}_{ii}\left(\boldsymbol{B}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{UB}\right)_{kk} + 2\alpha\left(\boldsymbol{L}^{\boldsymbol{V}}_{\mathrm{hyper}}\right)_{ii} + \\ & 4\beta\left(\boldsymbol{J}^{ik}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{V} + \boldsymbol{V}(\boldsymbol{J}^{ik})^{\mathrm{T}}\boldsymbol{V} + \boldsymbol{VV}^{\mathrm{T}}\boldsymbol{J}^{ik} - \boldsymbol{J}^{ik}\right)_{ik} \end{aligned} \tag{A22}$$

where $\boldsymbol{J}^{jk}$ denotes an $m\times k$ single-element matrix, whose element is 1 at coordinate (j, k), and 0 elsewhere, so are $\boldsymbol{J}^{kk'}$ and $\boldsymbol{J}^{ik}$.

Apparently, we are required to prove that F_{jk}, and $F_{kk'}$, and $F_{i,k}$ are monotonically decreasing under the update rules of Formulas (18)–(20).

Lemma 5 The functions constructed below are auxiliary functions for $F_{ik}(u), F_{kk'}(b)$, and $F_{jk}(v)$,

$$\begin{aligned} & G(u,u^{\mathrm{T}}_{jk}) = \\ & F_{jk}(u^{\mathrm{T}}_{jk}) + F'_{jk}(u^{\mathrm{T}}_{jk})(u-u^{\mathrm{T}}_{jk}) + \\ & \frac{\left(\boldsymbol{UBV}^{\mathrm{T}}\boldsymbol{DVB}^{\mathrm{T}} + \alpha\boldsymbol{D}^{\boldsymbol{U}}_{\mathrm{hyper}}\boldsymbol{U} + 2\beta\boldsymbol{UU}^{\mathrm{T}}\boldsymbol{U}\right)_{jk}}{u^{\mathrm{T}}_{jk}}(u-u^{\mathrm{T}}_{jk})^2 \end{aligned} \tag{A23}$$

$$\begin{aligned} G(b,b^{\mathrm{T}}_{kk'}) =& F_{kk'}(b^{\mathrm{T}}_{kk'}) + F'_{kk'}(b^{t}_{kk'})(b-b^{\mathrm{T}}_{kk'}) + \\ & \frac{\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{UBV}^{\mathrm{T}}\boldsymbol{DV}\right)_{kk'}}{b^{\mathrm{T}}}(b-b^{\mathrm{T}}_{kk'})^2 \end{aligned} \tag{A24}$$

$$\begin{aligned} & G(v,v^{\mathrm{T}}_{ik}) = F_{ik}(v^{\mathrm{T}}_{ik}) + F'_{ik}(v^{\mathrm{T}}_{ik})(v-v^{\mathrm{T}}_{ik}) + \\ & \frac{\left(\boldsymbol{DVB}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{UB} + \alpha\boldsymbol{D}^{\boldsymbol{V}}_{\mathrm{hyper}}\boldsymbol{V} + 2\beta\boldsymbol{VV}^{\mathrm{T}}\boldsymbol{V}\right)_{ik}}{v^{\mathrm{T}}_{ik}}(v-v^{2}_{ik}) \end{aligned} \tag{A25}$$

Proof We perform Taylor expansion on $F_{jk}(u)$, $F_{kk'}(b)$, and $F_{ik}(v)$, and get

$$F_{jk}(u) = F_{jk}(u^{\mathrm{T}}_{jk}) + F'_{jk}(u^{\mathrm{T}}_{jk})(u-u^{\mathrm{T}}_{jk}) + \frac{1}{2}F''_{jk}(u-u^{\mathrm{T}}_{jk})^2 \tag{A26}$$

$$F_{kk'}(b) = F_{kk'}(b^{\mathrm{T}}_{kk'}) + F'_{kk'}(b^{\mathrm{T}}_{kk'})(b-b^{\mathrm{T}}_{kk'}) + \frac{1}{2}F''_{kk'}(b-b^{\mathrm{T}}_{kk'})^2 \tag{A27}$$

$$F_{ik}(v) = F_{ik}(v^{\mathrm{T}}_{ik}) + F'_{ik}(v^{\mathrm{T}}_{ik})(v-v^{\mathrm{T}}_{ik}) + \frac{1}{2}F''_{ik}(v-v^{\mathrm{T}}_{ik})^2 \tag{A28}$$

It can be easily obtained that $G(u,u) = F_{jk}(u)$, $G(b,b) = F_{kk'}(b)$, and $G(v,v) = F_{ik}(v)$. Next we need to prove $G(u,u^{\mathrm{T}}_{jk}) \geqslant F_{jk}(u)$, $G(b,b^{\mathrm{T}}_{kk'}) \geqslant F_{kk'}(b)$, and $G(v,v^{\mathrm{T}}_{ik}) \geqslant F_{ik}(v)$. To guarantee $G(u,u^{\mathrm{T}}_{jk}) \geqslant F_{jk}(u)$, we combine Eqs. (A23) and (A26) and learn that proving the following inequality is needed:

$$\frac{\left(\boldsymbol{UBV}^{\mathrm{T}}\boldsymbol{DVB}^{\mathrm{T}}+\alpha\boldsymbol{D}^{\boldsymbol{U}}_{\mathrm{hyper}}\boldsymbol{U}+2\beta\boldsymbol{UU}^{\mathrm{T}}\boldsymbol{U}\right)_{jk}}{u^{\mathrm{T}}_{jk}}\geqslant$$
$$\left(\boldsymbol{BV}^{\mathrm{T}}\boldsymbol{DVB}^{\mathrm{T}}\right)_{kk}+\alpha\left(\boldsymbol{L}^{\boldsymbol{U}}_{\mathrm{hyper}}\right)_{jj}+$$
$$2\beta\left(\boldsymbol{J}^{jk}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}+\boldsymbol{U}(\boldsymbol{J}^{jk})^{\mathrm{T}}\boldsymbol{U}+\boldsymbol{UU}^{\mathrm{T}}\boldsymbol{J}^{jk}-\boldsymbol{J}^{jk}\right)_{jk} \quad \text{(A29)}$$

Since $\alpha>0$ and $\beta>0$, we have

$$\left(\boldsymbol{UBV}^{\mathrm{T}}\boldsymbol{DVB}^{\mathrm{T}}\right)_{jk}=\sum_{x=1}^{k}u^{\mathrm{T}}_{jx}\left(\boldsymbol{BV}^{\mathrm{T}}\boldsymbol{DVB}^{\mathrm{T}}\right)_{xk}\geqslant u^{\mathrm{T}}_{jk}\left(\boldsymbol{BV}^{\mathrm{T}}\boldsymbol{DVB}^{\mathrm{T}}\right)_{kk} \quad \text{(A30)}$$

$$\left(\boldsymbol{D}^{\boldsymbol{U}}_{\mathrm{hyper}}\boldsymbol{U}\right)_{jk}=\sum_{x=1}^{M}\left(\boldsymbol{D}^{\boldsymbol{U}}_{\mathrm{hyper}}\right)_{jx}u^{\mathrm{T}}_{xk}\geqslant\left(\boldsymbol{D}^{\boldsymbol{U}}_{\mathrm{hyper}}\right)_{jj}u_{jk}\geqslant\left(\boldsymbol{D}^{\boldsymbol{U}}_{\mathrm{hyper}}-\boldsymbol{S}^{\boldsymbol{U}}_{\mathrm{hyper}}\right)_{jj}u_{jk} \quad \text{(A31)}$$

$$\left(\boldsymbol{UU}^{\mathrm{T}}\boldsymbol{U}\right)_{jk}=\sum_{x=1}^{M}\sum_{y=1}^{K}u_{jy}u_{yx}u_{xk}\geqslant$$
$$\left(\boldsymbol{J}^{jk}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}+\boldsymbol{U}(\boldsymbol{J}^{jk})^{\mathrm{T}}\boldsymbol{U}+\boldsymbol{UU}^{\mathrm{T}}\boldsymbol{J}^{jk}\right)_{jk}u^{t}_{jk}\geqslant$$
$$\left(\boldsymbol{J}^{jk}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}+\boldsymbol{U}(\boldsymbol{J}^{jk})^{\mathrm{T}}\boldsymbol{U}+\boldsymbol{UU}^{\mathrm{T}}\boldsymbol{J}^{jk}-\boldsymbol{J}^{jk}\right)_{jk}u^{t}_{jk} \quad \text{(A32)}$$

Formulas (A30)–(A32) prove $G(u,u^{\mathrm{T}}_{jk})\geqslant F_{jk}(u)$ holds. Similarly, to guarantee $G(b,b^{\mathrm{T}}_{kk'})\geqslant F_{kk'}(b)$, we combine (A24) and (A27) and prove the following inequality:

$$\frac{\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{UBV}^{\mathrm{T}}\boldsymbol{DV}\right)_{kk'}}{b^{\mathrm{T}}_{kk'}}\geqslant\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\right)_{kk}\left(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{DV}\right)_{k'k'} \quad \text{(A33)}$$

We have

$$\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{UBV}^{\mathrm{T}}\boldsymbol{DV}\right)_{kk'}=\sum_{x=1}^{K}\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\right)_{kk}b^{\mathrm{T}}_{kx}\left(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{DV}\right)_{xk'}\geqslant\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}\right)_{kk}b^{\mathrm{T}}_{kk'}\left(\boldsymbol{V}^{\mathrm{T}}\boldsymbol{DV}\right)_{k'k'} \quad \text{(A34)}$$

Hence, $G(b,b^{\mathrm{T}}_{kk'})\geqslant F_{kk'}(b)$ holds. The proof for $G(v,v^{\mathrm{T}}_{ik})\geqslant F_{ik}(v)$ is similar to the proof for $G(u,u^{\mathrm{T}}_{jk})\geqslant F_{jk}(u)$, so we omit it here. ■

Proof of Theorem 2 Considering Lemma 4, we obtain the update rules for auxiliary function $G(u,u^{\mathrm{T}}_{jk})$, $G(b,b^{\mathrm{T}}_{kk'})$, and $G(v,v^{\mathrm{T}}_{ik})$,

$$u^{t+1}_{jk}-u^{t}_{jk}=-u^{\mathrm{T}}_{jk}\frac{F'_{jk}(u^{\mathrm{T}}_{jk})}{2\left(\boldsymbol{UBV}^{\mathrm{T}}\boldsymbol{DVB}^{\mathrm{T}}+\alpha\boldsymbol{D}^{\boldsymbol{U}}_{\mathrm{hyper}}\boldsymbol{U}+2\beta\boldsymbol{UU}^{\mathrm{T}}\boldsymbol{U}\right)_{jk}},$$
$$u^{t+1}_{jk}=\frac{\left(\boldsymbol{XDVB}^{\mathrm{T}}+\alpha\boldsymbol{S}^{\boldsymbol{U}}_{\mathrm{hyper}}\boldsymbol{U}+2\beta\boldsymbol{U}\right)_{jk}}{\left(\boldsymbol{UBV}^{\mathrm{T}}\boldsymbol{DVB}^{\mathrm{T}}+\alpha\boldsymbol{D}^{\boldsymbol{U}}_{\mathrm{hyper}}\boldsymbol{U}+2\beta\boldsymbol{UU}^{\mathrm{T}}\boldsymbol{U}\right)_{jk}} \quad \text{(A35)}$$

$$b^{t+1}_{kk'}-b^{t}_{kk'}=-b^{\mathrm{T}}_{kk'}\frac{F'_{kk'}(b^{\mathrm{T}}_{kk'})}{2\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{UBV}^{\mathrm{T}}\boldsymbol{DV}\right)_{kk'}},$$
$$b^{t+1}_{kk'}=b_{kk'}\frac{\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{XDV}\right)_{kk'}}{\left(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{UBV}^{\mathrm{T}}\boldsymbol{DV}\right)_{kk'}} \quad \text{(A36)}$$

$$v^{t+1}_{ik}-v^{t}_{ik}=-v^{\mathrm{T}}_{ik}\frac{F'_{ik}(v^{\mathrm{T}}_{ik})}{2\left(\boldsymbol{DVB}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{UB}+\alpha\boldsymbol{D}^{\boldsymbol{V}}_{\mathrm{hyper}}\boldsymbol{V}+2\beta\boldsymbol{VV}^{\mathrm{T}}\boldsymbol{V}\right)_{ik}},$$
$$v^{t+1}_{ik}=\frac{\left(\boldsymbol{DX}^{\mathrm{T}}\boldsymbol{UB}+\alpha\boldsymbol{S}^{\boldsymbol{V}}_{\mathrm{hyper}}\boldsymbol{V}+2\beta\boldsymbol{V}\right)_{ik}}{\left(\boldsymbol{DVB}^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{UB}+\alpha\boldsymbol{D}^{\boldsymbol{V}}_{\mathrm{hyper}}\boldsymbol{V}+2\beta\boldsymbol{VV}^{\mathrm{T}}\boldsymbol{V}\right)_{ik}} \quad \text{(A37)}$$

It can been seen that the update rules of Formulas (18)–(20) derived from the KKT condition are exactly the update rules for minimizing auxiliary function G. According to Lemma 4, O is monotonically decreasing under the update rules of Formulas (18)–(20). ■

References

[1] D. Seung and L. Lee, Algorithms for non-negative matrix factorization, *Advances in Neural Information Processing Systems*, vol. 13, pp. 556–562, 2001.

[2] P. Padilla, M. Lopez, J. M. Gorriz, J. Ramirez, D. Salas-Gonzalez, and I. Alvarez, NMF-SVM based CAD tool applied to functional brain images for the diagnosis of Alzheimer's disease, *IEEE Trans. Med. Imag.*, vol. 31, no. 2, pp. 207–216, 2012.

[3] J. J.-Y. Wang, X. Wang, and X. Gao, Non-negative matrix factorization by maximizing correntropy for cancer clustering, *BMC Bioinform.*, vol. 14, no. 1, p. 107, 2013.

[4] M. Nagayama, T. Aritake, H. Hino, T. Kanda, T. Miyazaki, M. Yanagisawa, S. Akaho, and N. Murata, Detecting cell assemblies by NMF-based clustering from calcium imaging data, *Neural Networks*, vol. 149, p. 29–39, 2022.

[5] A. Saha and V. Sindhwani, Learning evolving and emerging topics in social media: A dynamic NMF approach with temporal regularization, in *Proc. fifth ACM Int. Conf. Web Search and Data Mining*, Seattle, WA, USA, 2012, p. 693–702.

[6] A. Alfajri, D. Richasdy, and M. A. Bijaksana, Topic modelling using non-negative matrix factorization (NMF) for telkom university entry selection from instagram comments, *J. Comput. Syst. Inform. Josyc*, vol. 3, no. 4, pp. 485–492, 2022.

[7] N. Yu, Y.-L. Gao, J.-X. Liu, J. Wang, and J. Shang, Robust hypergraph regularized non-negative matrix factorization for sample clustering and feature selection in multi-view gene expression data, *Hum. Genom.*, vol. 13, no. 1, p. 46, 2019.

[8] F. Wang, T. Li, X. Wang, S. Zhu, and C. Ding, Community discovery using nonnegative matrix factorization, *Data Min. Knowl. Discov.*, vol. 22, no. 3, pp.

493–521, 2011.

[9] C. He, X. Fei, Q. Cheng, H. Li, Z. Hu, and Y. Tang, A survey of community detection in complex networks using nonnegative matrix factorization, *IEEE Trans. Comput. Soc. Syst.*, vol. 9, no. 2, pp. 440–457, 2022.

[10] I. Psorakis, S. Roberts, and B. Sheldon, Efficient Bayesian community detection using non-negative matrix factorisation, arXiv preprint, arXiv: 1009.2646, 2010.

[11] J. Cao, W. Xu, D. Jin, X. Zhang, A. Miller, L. Liu, and W. Ding, A network embedding-enhanced NMF method for finding communities in attributed networks, *IEEE Access*, vol. 10, pp. 118141–118155, 2022.

[12] W. Wu, S. Kwong, Y. Zhou, Y. Jia, and W. Gao, Nonnegative matrix factorization with mixed hypergraph regularization for community detection, *Inf. Sci.*, vol. 435, pp. 263–281, 2018.

[13] N. Yu, M.-J. Wu, J.-X. Liu, C.-H. Zheng, and Y. Xu, Correntropy-based hypergraph regularized NMF for clustering and feature selection on multi-cancer integrated data, *IEEE Trans. Cybern.*, vol. 51, no. 8, pp. 3952–3963, 2021.

[14] N. Yu, Y.-L. Gao, J.-X. Liu, J. Wang, and J. Shang, Hypergraph regularized NMF by L2, 1-norm for clustering and com-abnormal expression genes selection, in *Proc. IEEE Int. Conf. Bioinformatics and Biomedicine*, Madrid, Spain, 2018, pp. 578–582.

[15] M. Venkatasubramanian, K. Chetal, D. J. Schnell, G. Atluri, and N. Salomonis, Resolving single-cell heterogeneity from hundreds of thousands of cells through sequential hybrid clustering and NMF, *Bioinformatics*, vol. 36, no. 12, pp. 3773–3780, 2020.

[16] Y.-J. Hao, Y.-L. Gao, M.-X. Hou, L.-Y. Dai, and J.-X. Liu, Hypergraph regularized discriminative nonnegative matrix factorization on sample classification and co-differentially expressed gene selection, *Complexity*, vol. 2019, p. 12, 2019.

[17] M. H. Aghdam and M. D. Zanjani, A novel regularized asymmetric non-negative matrix factorization for text clustering, *Inf. Process. Manag.*, vol. 58, no. 6, p. 102694, 2021.

[18] M. Bansal and D. Sharma, A novel multi-view clustering approach *via* proximity-based factorization targeting structural maintenance and sparsity challenges for text and image categorization, *Inf. Process. Manag.*, vol. 58, no. 4, p. 102546, 2021.

[19] H. Che and J. Wang, A nonnegative matrix factorization algorithm based on a discrete-time projection neural network, *Neural Netw.*, vol. 103, pp. 63–71, 2018.

[20] H. Che, J. Wang, and A. Cichocki, Bicriteria sparse nonnegative matrix factorization *via* two-timescale duplex neurodynamic optimization, *IEEE Trans. Neural Netw. Learning Syst.*, vol. 34, no. 8, pp. 4881–4891, 2023.

[21] D. Kong, C. Ding, and H. Huang, Robust nonnegative matrix factorization using L21-norm, in *Proc. 20th ACM Int. Conf. Information and knowledge management*, Glasgow Scotland, UK, 2011, pp. 673–682.

[22] H. Gao, F. Nie, W. Cai, and H. Huang, Robust capped norm nonnegative matrix factorization: Capped norm NMF, in *Proc. 24th ACM Int. on Conf. Information and Knowledge Management*, Melbourne, Australia, 2015, pp. 871–880.

[23] N. Guan, T. Liu, Y. Zhang, D. Tao, and L. S. Davis, Truncated cauchy non-negative matrix factorization, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 1, pp. 246–259, 2019.

[24] N. Guan, D. Tao, Z. Luo, and J. Shawe-Taylor, MahNMF: Manhattan non-negative matrix factorization, arXiv preprint, arXiv:1207.3438, 2012.

[25] Z. Lin, C. Xu, and H. Zha, Robust matrix factorization by majorization minimization, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 208–220, 2018.

[26] B. Yang, X. Zhang, B. Chen, F. Nie, Z. Lin, and Z. Nan, Efficient correntropy-based multi-view clustering with anchor graph embedding, *Neural Networks*, vol. 146, pp. 290–302, 2022.

[27] L. Li, J. Yang, K. Zhao, Y. Xu, H. Zhang, and Z. Fan, Graph regularized non-negative matrix factorization by maximizing correntropy, *J. Comput.*, vol. 9, pp. 2570–2579, 2014.

[28] X. Yang, H. Che, M.-F. Leung, and C. Liu, Adaptive graph nonnegative matrix factorization with the self-paced regularization, *Appl. Intell.*, vol. 53, no. 12, pp. 15818–15835, 2023.

[29] D. Cai, X. He, J. Han, and T. S. Huang, Graph regularized nonnegative matrix factorization for data representation, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, 2011.

[30] F. Shang, L. C. Jiao, and F. Wang, Graph dual regularization non-negative matrix factorization for co-clustering, *Pattern Recognit.*, vol. 45, no. 6, pp. 2237–2250, 2012.

[31] D. Tolić, N. Antulov-Fantulin, and I. Kopriva, A nonlinear orthogonal non-negative matrix factorization approach to subspace clustering, *Pattern Recognit.*, vol. 82, pp. 40–55, 2018.

[32] J. Yu, D. Tao, and M. Wang, Adaptive hypergraph learning and its application in image classification, *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3262–3272, 2012.

[33] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, Hypergraph neural networks, *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, pp. 3558–3565, 2019.

[34] W. Wang, Y. Qian, and Y. Y. Tang, Hypergraph-regularized sparse NMF for hyperspectral unmixing, *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.*, vol. 9, no. 2, pp. 681–694, 2016.

[35] K. Zeng, J. Yu, C. Li, J. You, and T. Jin, Image clustering by hyper-graph regularized non-negative matrix factorization, *Neurocomputing*, vol. 138, pp. 209–217, 2014.

[36] A. Salah, M. Ailem, and M. Nadif, Word co-occurrence regularized non-negative matrix tri-factorization for text data co-clustering, in *Proc. AAAI Conf. Artif. Intell.*, https://doi.org/10.1609/aaai.v32i1.11659, 2018.

[37] T. Hwang, G. Atluri, M. Xie, S. Dey, C. Hong, V. Kumar, and R. Kuang, Co-clustering phenome-genome for phenotype classification and disease gene discovery,

Nucleic Acids Res., vol. 40, no. 19, p. e146, 2012.
[38] P. Deng, T. Li, H. Wang, S.-J. Horng, Z. Yu, and X. Wang, Tri-regularized nonnegative matrix tri-factorization for co-clustering, *Knowl. Based Syst.*, vol. 226, p. 107101, 2021.
[39] C. Ding, T. Li, W. Peng, and H. Park, Orthogonal nonnegative matrix t-factorizations for clustering, in *Proc. 12th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, Philadelphia, PA, USA, 2006, pp. 126–135.
[40] S. Wang and A. Huang, Penalized nonnegative matrix tri-factorization for co-clustering, *Expert Syst. Appl.*, vol. 78, pp. 64–73, 2017.
[41] S. Peng, W. Ser, B. Chen, and Z. Lin, Robust orthogonal nonnegative matrix tri-factorization for data representation, *Knowl. Based Syst.*, vol. 201, p. 106054, 2020.
[42] Y. Xu, L. Lu, Q. Liu, and Z. Chen, Hypergraph-regularized Lp smooth nonnegative matrix factorization for data representation, *Mathematics*, vol. 11, no. 13, p. 2821, 2023.
[43] C.-N. Jiao, J.-X. Liu, Y.-L. Gao, X.-Z. Kong, C.-H. Zheng, and X. Yu, Sparse hyper-graph non-negative matrix factorization by maximizing correntropy, in *Proc. IEEE Int. Conf. Bioinformatics and Biomedicine*, Houston, TX, USA, 2021, pp. 418–423.
[44] S. Li, W. Li, J. Hu, and Y. Li, Semi-supervised bi-orthogonal constraints dual-graph regularized NMF for subspace clustering, *Appl. Intell.*, vol. 52, no. 3, pp. 3227–3248, 2022.
[45] G. Lu, C. Leng, B. Li, L. Jiao, and A. Basu, Robust dual-graph discriminative NMF for data classification, *Knowl. Based Syst.*, vol. 268, p. 110465, 2023.
[46] L. Lovász and M. D. Plummer, *Matching Theory*. Rhode Island, USA: AMS Chelsea Publishing, 2009.
[47] D. Lee and H. S. Seung, Algorithms for non-negative matrix factorization, in *Proc. Advances in Neural Information Processing Systems*, Denver, CO, USA, 2000, pp. 535–541.

Jiyang Yu is currently an undergraduate student at School of Electronic Information, Southwest University, China. His main research interest is machine learning.

Man-Fai Leung received the PhD degree in computer science from The City University of Hong Kong, Hong Kong, China in 2019. He is currently working at School of Computing and Information Science, Faculty of Science and Engineering, Anglia Ruskin University, UK. His research interests include portfolio optimization, multiobjective optimization, and computational intelligence.

Zheng Yan received the BEng degree in automation and computer-aided engineering in 2010, and the PhD degree in mechanical and automation engineering in 2014, both from The Chinese University of Hong Kong, Hong Kong, China. He is currently a research fellow at Australian Artificial Intelligence Institute, University of Technology Sydney, Australia. Prior to this position, he took several leadership roles in tech companies, including Alibaba (China), contributing to the research and development of machine-learning systems in cloud computing, video analytics, and logistics. His research interests include neural networks, neuromorphic computing, and machine learning. He has been an area editor of *International Journal of Computational Intelligence Systems* since 2018. He served on the organizing committees of several international conferences, including ICONIP and IJCNN. He is a chartered engineer in Australia.

Hangjun Che received the BEng degree in electronic and information engineering from Chongqing University of Posts and Telecommunications, China in 2013, and the MEng degree in signal and information processing from Southwest University, China in 2016, and the PhD degree in computer science from The City University of Hong Kong, Hong Kong, China in 2019. He is currently an associate professor at College of Electronic and Information Engineering, Southwest University, China. His research interests include neurodynamic optimization, sparse coding, and nonnegative matrix factorization.

Cheng Liu received the PhD degree in computer science from The City University of Hong Kong, Hong Kong, China in 2018. He is currently an assistant professor at Department of Computer Science, Shantou University, China. His research interests include machine learning and pattern recognition.

Wenhui Wu received the BEng and MEng degrees from Xidian University, China in 2012 and 2015, respectively, and the PhD degree in computer science from The City University of Hong Kong, Hong Kong, China in 2019. She is currently an associate professor at College of Electronics and Information Engineering, Shenzhen University, China. Her research interests include machine learning, image enhancement, and community detection.