

Article

Sparse Feature-Weighted Double Laplacian Rank Constraint Non-Negative Matrix Factorization for Image Clustering

Hu Ma ¹, Ziping Ma ^{1,*} , Huirong Li ² and Jingyu Wang ¹¹ School of Mathematics and Information Science, North Minzu University, Yinchuan 750030, China² School of Mathematics and Computer Application, Shangluo University, Shangluo 726000, China

* Correspondence: 2006041@nmu.edu.cn or mazingping@tom.com

Abstract: As an extension of non-negative matrix factorization (NMF), graph-regularized non-negative matrix factorization (GNMF) has been widely applied in data mining and machine learning, particularly for tasks such as clustering and feature selection. Traditional GNMF methods typically rely on predefined graph structures to guide the decomposition process, using fixed data graphs and feature graphs to capture relationships between data points and features. However, these fixed graphs may limit the model's expressiveness. Additionally, many NMF variants face challenges when dealing with complex data distributions and are vulnerable to noise and outliers. To overcome these challenges, we propose a novel method called sparse feature-weighted double Laplacian rank constraint non-negative matrix factorization (SFLRNMF), along with its extended version, SFLRNMTF. These methods adaptively construct more accurate data similarity and feature similarity graphs, while imposing rank constraints on the Laplacian matrices of these graphs. This rank constraint ensures that the resulting matrix ranks reflect the true number of clusters, thereby improving clustering performance. Moreover, we introduce a feature weighting matrix into the original data matrix to reduce the influence of irrelevant features and apply an $L_{2,1/2}$ norm sparsity constraint in the basis matrix to encourage sparse representations. An orthogonal constraint is also enforced on the coefficient matrix to ensure interpretability of the dimensionality reduction results. In the extended model (SFLRNMTF), we introduce a double orthogonal constraint on the basis matrix and coefficient matrix to enhance the uniqueness and interpretability of the decomposition, thereby facilitating clearer clustering results for both rows and columns. However, enforcing double orthogonal constraints can reduce approximation accuracy, especially with low-rank matrices, as it restricts the model's flexibility. To address this limitation, we introduce an additional factor matrix R , which acts as an adaptive component that balances the trade-off between constraint enforcement and approximation accuracy. This adjustment allows the model to achieve greater representational flexibility, improving reconstruction accuracy while preserving the interpretability and clustering clarity provided by the double orthogonality constraints. Consequently, the SFLRNMTF approach becomes more robust in capturing data patterns and achieving high-quality clustering results in complex datasets. We also propose an efficient alternating iterative update algorithm to optimize the proposed model and provide a theoretical analysis of its performance. Clustering results on four benchmark datasets demonstrate that our method outperforms competing approaches.

Keywords: non-negative matrix factorization; local structure learning; feature weighting; sparse constraints; constrained Laplacian rank

MSC: 68Q99



Citation: Ma, H.; Ma, Z.; Li, H.; Wang, J. Sparse Feature-Weighted Double Laplacian Rank Constraint Non-Negative Matrix Factorization for Image Clustering. *Mathematics* **2024**, *12*, 3656. <https://doi.org/10.3390/math12233656>

Academic Editor: Sheng-Wei Chi

Received: 16 October 2024

Revised: 14 November 2024

Accepted: 18 November 2024

Published: 22 November 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the rapid development of information technology and artificial intelligence, the scale and dimensionality of data are also increasing significantly. This vast amount of

high-dimensional data pose great challenges to traditional machine learning and statistical analysis. In the fields of data mining and machine learning, data representation [1] is a fundamental and crucial task. It remains a fundamental and critical task, as it directly influences the performance and interpretability of subsequent models. Effective low-dimensional data representation can handle massive high-dimensional data, reduce redundant features in the original data, and reveal the latent structural information of the data [2]. The primary objective of data representation is to effectively characterize the original data using various techniques, thereby facilitating subsequent tasks like clustering and classification. As a result, data representation plays a crucial role in a range of applications, including information retrieval, classification, hyperspectral image processing [3], and information extraction [4–7].

Over the past few decades, to effectively handle high-dimensional data, researchers have developed a variety of data representation methods. These methods include Principal Component Analysis (PCA) [8], Manifold Learning [9–11], Linear Discriminant Analysis (LDA) [12], Concept Factorization (CF) [13], Sparse Coding (SC) [14], non-negative matrix factorization (NMF) [15,16] Deep Learning (DL) [17], and low-rank representation (LRR) [18]. Among them, non-negative matrix factorization (NMF) has become one of the most widely used data representation techniques [19] due to its excellent interpretability. However, the classic NMF algorithm only imposes non-negativity constraints, which is not sufficient to meet the diverse clustering needs. Therefore, to enhance its clustering performance, researchers have embedded various additional constraints into the NMF algorithm. For example, Gao et al. [20] proposed a sparse non-negative matrix factorization (SNMF) algorithm, which enhances the algorithm's learning capability by embedding sparsity constraints as a penalty term within the NMF framework. Similarly, Ding et al. [21] introduced an Orthogonal Non-negative Matrix Tri-factorization (ONMTF) algorithm, which imposes orthogonality constraints on both the basis matrix and coefficient matrix, aiming to produce clearer and more interpretable clustering results during NMF decomposition. Although these NMF variants improve clustering performance by imposing different constraints on the classical NMF algorithm, they fail to consider the manifold structure in the data, which is crucial for the manifold structure of clustering data. To explore the geometric structure of data manifolds and feature manifolds, researchers encode the geometric information of data and feature spaces by constructing similarity graphs and embed graph regularization into the original NMF to reveal the intrinsic geometric structure of the data.

Consequently, various graph-regularized NMF variants have been proposed. For example, Cai et al. [22] introduced graph-regularized NMF (GNMF), which learns the local manifold structure of the data space by constructing a similarity graph. However, GNMF only considers the similarity in the data space and does not address the feature space. For this purpose, Shang et al. [23] proposed dual regularized NMF (DNMF) by constructing two similarity graphs to explore the geometric structure information in both the data and feature spaces simultaneously. Inspired by Shang et al. [23], Sun et al. [24] introduced sparse dual graph-regularized NMF (SDGNMF), which not only incorporates label information into the graph regularization but also imposes sparsity constraints on the basis matrix. To prevent misalignment between the image and basis vectors and further enhance the algorithm's discriminative ability, Li et al. [25] proposed semi-supervised dual orthogonally constrained dual graph-regularized NMF (SDGNMF-BO). This method integrates dual orthogonality constraints and dual graph regularization into a semi-supervised NMF framework, further enhancing the learning capability in the subspace. To maximize the sparsity of the learned coefficient matrix, Li et al. [26] introduced semi-supervised graph and local coordinate regularized NMF. By embedding local coordinate constraints into a semi-supervised NMF framework with graph regularization, the sparsity of the coefficient matrix is enhanced. Inspired by Li et al. [26] and considering that matrix factorization may have multiple solutions, Wang et al. [27] proposed locally orthogonally constrained semi-supervised dual graph-regularized NMF (LOSDNMF). This method integrates dual graph regularization, local coordinate constraints, and dual orthogonality constraints into

a semi-supervised NMF framework, effectively enhancing the sparsity and discriminative power of data representation. However, although graph regularization methods are generally superior to many other approaches, these methods often rely on the K-nearest neighbors (KNN) approach to construct the graph structure. However, this construction can interfere with the matrix factorization process, potentially leading to suboptimal clustering results. To explore the nonlinear structure of data and construct a similarity graph with an optimal block-diagonal structure, Xu et al. [28] proposed an explicit data-driven kernel learning strategy. This strategy directly learns the kernel through the self-representation of the data, simultaneously enabling adaptive weighting. Based on this kernel, the local manifold structure of the data can be preserved in the nonlinear space via a kernel-based local manifold term, facilitating the construction of a graph structure with an optimal block-diagonal form. Recognizing that multi-view clustering can enhance clustering performance by effectively integrating complementary information from different views, Xu et al. [29] further proposed a novel multi-view clustering algorithm that adaptively constructs a kernel matrix without requiring a predefined kernel function. Inspired by the aforementioned research, constructing adaptive graph structures has also gained increasing attention in non-negative matrix factorization (NMF). For instance, the NMF with Adaptive Neighbors (NMFAN) method [30] introduces an adaptive graph mechanism to achieve simultaneous optimization of matrix factorization and similarity learning. This method balances the interactions between these two sub-tasks, allowing each sub-task to iteratively optimize based on the results of the other, thereby ensuring a more accurate construction of the similarity matrix in the data graph. Additionally, Shu et al. [31] proposed a new data representation method (RCNMF) by imposing a rank constraint on the Laplacian matrix of the learned graph to ensure that the connected components precisely match the sample categories. On the other hand, with the continuous development of non-negative matrix factorization (NMF), an increasing number of constraints have been introduced. However, excessive constraints may lead to unreliable solutions. Therefore, to address the issue of limited degrees of freedom caused by too many constraints, some researchers have introduced a third decomposition factor, denoted as R , within the NMF framework. This additional factor, serving as a scaling function, not only provides extra degrees of freedom for the factor matrices X , U , and V , but also enhances the flexibility of the decomposition process. For example, Tang et al. [32] proposed a new three-factor matrix decomposition model. By introducing dual graph regularization and dual orthogonality constraints into NMF, this model not only explores the geometric properties of data and feature manifolds but also ensures the orthogonality of the factor matrices.

Inspired by the aforementioned algorithms, we have not fixed the input feature map and input data graph related to the affinity matrix in our model. Instead, we learn new data similarity matrix D and feature similarity matrix A based on the initial data similarity matrix W and the initial feature similarity matrix S . These new optimal similarity graphs are more suitable for clustering tasks. Moreover, we have imposed rank constraints on the Laplacian graphs of these two new similarity matrices to ensure that the number of connected components is consistent with the number of sample categories. Furthermore, inspired by recent research on orthogonal constraints by Ding et al. [21] and sparse constraints by Luo et al. [33], we have introduced sparsity and double orthogonal constraints within the non-negative matrix factorization framework with dual Laplacian rank constraints. These double orthogonal constraints not only address the issues of slow optimization and high computational complexity in existing NMF models but also prevent mismatches between images and base vectors. Thus, this enhancement effectively improves the discriminative and exclusive nature of clustering.

Therefore, we propose a novel non-negative matrix factorization algorithm called sparse feature-weighted dual Laplacian rank-constrained non-negative matrix factorization (SFLRNMF), along with its extended version, sparse feature-weighted dual Laplacian rank-constrained non-negative tri-factor matrix factorization (SFLRNMTF). Our main contributions include the following:

- (1) A novel learning mechanism, dual Laplacian rank-constrained non-negative matrix factorization (SFLRNMF), has been proposed. This mechanism is capable of learning the optimal feature similarity graph and data similarity graph, and rank constraints are applied to the Laplacian matrices of both graphs to construct an optimal dual graph regularizer. Additionally, a weight matrix has been constructed to explore the attributes of the original data and the diversity of the samples. Furthermore, an $L_{2,1/2}$ -norm sparsity constraint has been imposed on the basis matrix to promote sparsity, simplify computation, and enhance the model's local learning capabilities and robustness.
- (2) Based on SFLRNMF, the dual Laplacian rank-constrained non-negative matrix tri-factorization (SFLRNMTF) model is proposed. Specially, orthogonal constraints are imposed on both the coefficient matrix and the basis matrix to ensure that each data point has a unique basis vector in the feature space, which allows one to enhance the discriminative ability of clustering. Additionally, excessive constraints may lead to too few degrees of freedom, which results in unreliable solutions during factor decomposition. Therefore, factor R has been introduced to ensure the accuracy of the factor decomposition.
- (3) Corresponding iterative update optimization schemes have been proposed, and convergence proofs for the two algorithms have been provided. Furthermore, the effectiveness of these two models has been validated by conducting experiments on benchmark datasets and comparing the results with several of the most advanced clustering methods.

The rest of the paper is structured as follows: In Section 2, we introduce the basic principles of standard NMF and its variants. In Section 3, the novel SFLRNMF algorithm is presented, and its convergence is provided theoretically. Section 4 introduces the extended version of SFLRNMF and provides the convergence of the optimization process. In Section 5, we conduct numerical experiments to demonstrate the efficiency of our two methods. In Section 6, we summarize the paper and discuss future work.

2. Related Works

2.1. NMF

NMF aims to approximate the original high-dimensional data matrix x by decomposing it into two lower-dimensional matrices u and v . Some of the notation used in NMF is shown in Table 1.

Given a non-negative dataset $X = [x_1, x_2, \dots, x_n] \in R^{m \times n}$, $U \in P^{m \times r}$, $V \in R^{n \times r}$, and $X \approx UV^T$. Here, $r \ll \min(m, n)$, where U represents the basis matrix in the data space, and V represents the coefficient matrix in the feature space. The objective function of this problem is as follows:

$$J_1 = \|X - UV^T\|_F^2 = \sum_{i=1}^p \sum_{j=1}^n \left(x_{ij} - \sum_{l=1}^r u_{il}v_{jl} \right)^2 \quad (1)$$

s.t. $U \geq 0, V \geq 0$

In the formula, $\|\cdot\|_F$ represents the Frobenius norm. The objective function J_1 in the above Formula (1) is non-convex with respect to the joint variables (U, V) . To make J_1 a convex function, it is necessary to fix one of the variables (U, V) . Therefore, the objective function can be solved iteratively using an alternating approach. To effectively solve the objective function of NMF, Lee et al. [15] proposed the well-known multiplicative update method, which allows the objective function to be solved using a simple multiplicative iterative update approach. The update rules for Formula (1) are given below:

$$u_{ij} \leftarrow u_{ij} \frac{(XV)_{ij}}{(UV^T)_{ij}} \quad (2)$$

$$v_{ij} \leftarrow v_{ij} \frac{(X^T U)_{ij}}{(V U^T U)_{ij}} \tag{3}$$

Here, $i \in \{1, \dots, m\}$, $l \in \{1, \dots, r\}$ and $j \in \{1, \dots, n\}$.

At the beginning of the iterative process, we randomly initialize the basis matrix U and the coefficient matrix V and then update them according to the iterative update rules in Formulas (2) and (3) until the final condition is satisfied.

Table 1. List of notation.

Notation	Notation Description
$X \in R^{m \times n}$	Data matrix
$T \in R^{m \times m}$	Feature-weighted matrix
$R \in R^{c \times c}$	Diagonal scaling matrix
$A \in R^{n \times n}$	Initial data affinity matrix
$S \in R^{n \times n}$	Data affinity matrix
$D \in R^{m \times m}$	Initial feature affinity matrix
$W \in R^{m \times m}$	Feature affinity matrix
$U \in R^{m \times c}$	Basis matrix
$V \in R^{n \times c}$	Coefficient matrix
I	Unit matrix
1	Vector with all elements being 1
$L_S \in R^{n \times n}$	Data Laplacian matrix
$L_W \in R^{m \times m}$	Feature Laplacian matrix
$D_S \in R^{n \times n}$	Data graph degree matrix
$D_W \in R^{m \times m}$	Feature graph degree matrix
α	Dual graph parameter
β	Orthogonal parameter
θ	Sparse parameter
m	The number of data dimensions
n	The number of data points
c	The number of data clusters
k	The number of nearest data points

2.2. GNMF

Cai et al. [22]. developed a graph-regularized non-negative matrix factorization (GNMF) for data representation. This model constructs an affinity graph to describe the manifold structure embedded in a high-dimensional ambient space, aiming to optimize the problem. The objective function of GNMF is as follows:

$$O_{GNMF} = \|X - UV^T\|_F^2 + \lambda Tr(V^T L V) \tag{4}$$

s.t. $U \geq 0, V \geq 0$

Here, L is the Laplacian matrix in the data space. For the above Equation (4), the following update rules are provided:

$$u_{ij} \leftarrow u_{ij} \frac{(XV)_{ij}}{(UV^T V)_{ij}} \tag{5}$$

$$v_{ij} \leftarrow v_{ij} \frac{(X^T U + \lambda W V)_{ij}}{(V U^T U + \lambda D V)_{ij}} \tag{6}$$

2.3. NMFAN

Huang et al. [30] pay special attention to the local connectivity of data points in their adaptive graph-regularized non-negative matrix factorization (NMFAN) method. They strive to construct an ideal similarity matrix aimed at achieving effective neighbor

allocation. Based on the assumption that data points closer in distance are more likely to become neighbors, this method selects the best neighbors for each data point to build the similarity matrix. By imposing constraints on this matrix, the neighbor allocation process becomes adaptive. The objective function of NMFAN is as follows:

$$O_{NMFAN} = \|X - UV^T\|_F^2 + \lambda Tr(V^T L_S V) + \sum_{i,j=1}^n \left(\|x_i - x_j\|_2^2 S_{ij} + \gamma S_{ij}^2 \right) \tag{7}$$

$$s.t. s_i^T \mathbf{1} = 1, 0 \leq S_i \leq 1, U \geq 0, V \geq 0$$

Based on the above objective function, the iterative update rules for Formula (7) are as follows:

$$u_{ij} \leftarrow u_{ij} \frac{(XV)_{ij}}{(UV^T V)_{ij}} \tag{8}$$

$$v_{ij} \leftarrow v_{ij} \frac{(X^T U + \lambda W_S V)_{ij}}{(V U^T U + \lambda D_S V)_{ij}} \tag{9}$$

For the update of S in Equation (7), the following problem needs to be solved:

$$\min_{s_i^T \mathbf{1}=1, 0 \leq S_i \leq 1} \left\| s_i + \frac{1}{2\gamma} d^i \right\|_2^2 \tag{10}$$

Here, d^i is the j -th column element of d^{ij} , specifically as follows:

$$d^{ij} = \|x_i - x_j\|_2^2 + \frac{\lambda}{2\mu} \|v_i - v_j\|_2^2 \tag{11}$$

For the update of γ in Equation (7), the following problem needs to be solved:

$$\gamma = \frac{1}{n} \sum_{i=1}^n \left(\frac{k}{2} \|x_i - x_{k+1}\|_2^2 - \frac{1}{2} \sum_{j=1}^k \|x_i - x_j\|_2^2 \right) \tag{12}$$

2.4. DNMF

Considering that both the observed data and features are situated on a low-dimensional manifold, Shang et al. [23] proposed the dual graph-regularized non-negative matrix factorization (DNMF). This model constructs dual affinity graphs to simultaneously mine the geometric structural information contained in the data points and features, aiming to optimize the problem. The objective function of DNMF is as follows:

$$O_{DNMF} = \|X - UV^T\|_F^2 + \lambda Tr(V^T L_V V) + \mu Tr(U^T L_U U) \tag{13}$$

$$s.t. U \geq 0, V \geq 0$$

Based on the above objective function, the iterative update rules for Formula (13) are as follows:

$$u_{ij} \leftarrow u_{ij} \frac{(XV + \mu W^U U)_{ij}}{(UV^T V + \mu D^U U)_{ij}} \tag{14}$$

$$v_{ij} \leftarrow v_{ij} \frac{(X^T U + \lambda W^V V)_{ij}}{(V U^T U + \lambda D^V V)_{ij}} \tag{15}$$

2.5. SDGNMF-BO

Inspired by co-clustering algorithms, Li et al. [25] introduced dual graph regularization into the semi-supervised non-negative matrix factorization (NMF) framework and proposed the semi-supervised dual orthogonal constrained dual graph-regularized subspace clustering NMF method (SDGNMF-BO). This method aims to more effectively utilize the

latent structures and class information in the data. To enhance the model’s recognition capability, they imposed soft orthogonal constraints on the decomposed basis and coefficient matrices. The objective function of SDGNMF-BO is as follows:

$$O_{SDGNMF-BO} = \|X - HA^T C^T\|_F^2 + \alpha [Tr(H^T L^H H) + Tr(A^T C^T L^A C A)] + \beta [Tr(H^T H - I) + Tr(A^T C^T C A - I)] \tag{16}$$

s.t. $H^T H = I, A^T C^T C A = I, h_{ij} \geq 0, c_{ij} \geq 0$

Based on the above objective function, the iterative update rules for Formula (16) are as follows:

$$h_{ij} \leftarrow h_{ij} \frac{(XCA + \alpha W^H H)_{ij}}{(A^T C^T C A + \alpha D^H H + \beta H)_{ij}} \tag{17}$$

$$a_{ij} \leftarrow a_{ij} \frac{(C^T X^T H + \alpha C^T W^A C A)_{ij}}{(C^T C A H^T H + \alpha C^T D^A C A + \beta C^T C A)_{ij}} \tag{18}$$

$$c_{ij} \leftarrow c_{ij} \frac{(X^T H A^T + \alpha W^A C A)_{ij}}{(C A H^T H A^T + \alpha D^A C A A^T + \beta C A A^T)_{ij}} \tag{19}$$

3. The Presented Model

3.1. The Motivation of the Proposed Method

Inspired by Local Linear Embedding (LLE) [10] and Laplacian Eigenmaps (LE) [11], researchers have developed several graph-regularized NMF clustering algorithms. However, the clustering performance of these algorithms highly depends on the quality of the graph model. Therefore, we propose automatically adjusting the weights of a given similarity matrix to learn the optimal similarity matrix. Specifically, our goal is to learn the new data similarity matrix D and feature similarity matrix A based on the initial data similarity matrix W and the initial feature similarity matrix S, thus constructing an optimal graph more suited for clustering tasks. Figure 1 shows the process of constructing this optimal graph. We use traditional weighting methods, such as 0–1 weighting, heat kernel weighting, or probabilistic neighborhood methods, to construct nearest neighbor graphs related to the data similarity matrix W and feature similarity matrix S. Then, we impose rank constraints on the Laplacian matrix of the learned similarity matrices and iteratively update the similarity matrices A and D.

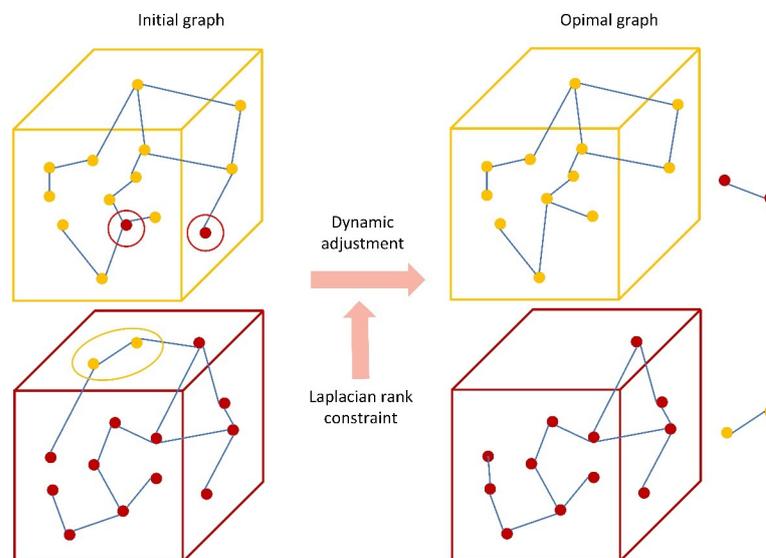


Figure 1. Construction of optimal graph.

Through the analysis above, in this section, we propose a new NMF model, namely sparse feature-weighted dual graph Laplacian rank-constrained non-negative matrix factorization (SFLRNMF), designed for graph-based clustering.

3.2. Sparse Constraints

Sparse constraints involve using appropriate sparse models to represent sparse data. Introducing sparse constraints into non-negative matrix factorization (NMF) combines the advantages of NMF and sparse representation, thus enhancing the effectiveness of NMF methods. In particular, imposing sparse constraints on the basis matrix U when decomposing the original matrix X has been proven to be a very successful and practical strategy. When each row of the basis matrix is sparse, fewer basis elements are needed to represent the original matrix, which greatly aids in data recovery. Therefore, sparse constraints have received widespread attention in recent years. Xu et al.'s [34] research shows that the sparse effect of the L_p -norm is better when $p = 1/2$, making the L_p -norm-based sparse constraint an increasingly favored condition among researchers. We first use the L_p -norm to impose sparse constraints on the basis matrix U with the specific method as follows:

$$\|U\|_{2,1/2} = \left(\sum_{i=1}^n \|U_i\|_2^{1/2} \right)^2 \quad (20)$$

Imposing L_p -norm sparse constraints on the basis matrix enhances the algorithm's robustness, local learning capability, and clustering performance, while making the basis matrix U sparser and simplifying the computational process.

3.3. Bi-Orthogonal Constraints

Ding et al. [21] proposed that in non-negative matrix factorization, satisfying $X \approx UV^T$, for a given solution (U, V) , there also exists another solution (UA, VB) where $AB^T = I$ and $UA \geq 0, VB \geq 0$. To avoid such erroneous solutions, orthogonal constraints should be applied to the basis matrix after decomposition, such that $UU^T = I$. Additionally, to differentiate various features and ensure that each feature vector points in distinct directions, which facilitates clearer and more distinguishable clustering in the sample space, orthogonal constraints should also be applied to the coefficient matrix, such that $VV^T = I$. Imposing orthogonal constraints in both the feature and sample spaces helps to distinguish different features and samples more prominently. This significantly enhances the performance of clustering algorithms, making different data groups more distinct and independent.

3.4. Feature Weighting

In this section, we integrate a feature weighting mechanism into non-negative matrix factorization (NMF) to better differentiate the importance of features in the original matrix, thereby improving the model's performance and interpretability. By introducing a feature weighting matrix T , the objective function can be summarized as follows:

$$\begin{aligned} O &= \left\| TX - UV^T \right\|_F^2 \\ \text{s.t. } &V \geq 0, U \geq 0, T_i = \text{diag}(t_i), t_i \geq 0, \sum_{i=1}^m t_i = 1 \end{aligned} \quad (21)$$

3.5. Models of Proposed Methods

Our strategy is to construct an optimal dual graph regularizer based on the initial data similarity matrix and the initial feature similarity matrix, both of which block diagonal matrices. To establish this objective, we begin with the following theorem.

Theorem 1 ([35]). *If the affinity matrix is non-negative, the Laplacian matrix $L_w = D_w - (W^T + W)/2$, where the i -th element is $\sum_j (w_{ij} + w_{ji})/2$. The degree matrix $D_w \in \mathbb{R}^{n \times n}$ is defined as a diagonal matrix with the following important properties [36,37], where W is the initial data similarity graph.*

Theorem 1 indicates that given an affinity matrix W , if the rank of the Laplacian matrix is equal to $n-k$, it is an ideal graph. Therefore, for a given initial data affinity matrix W and feature affinity matrix S , we can learn the corresponding data similarity matrix D and feature similarity matrix A . The Laplacian matrices corresponding to these two similarity matrices are $D_S - (S^T + S)/2$ and $L_W = D_W - (W^T + W)/2$, respectively.

By applying rank constraints to the two Laplacian matrices, namely $rank(L_S) = n - k$ and $rank(L_W) = m - k$, and using the Frobenius norm to measure the approximation error between the initial affinity matrices W and S and the learned similarity matrices D and A , respectively, the constrained Laplacian rank for graph-based clustering can be formulated as the following problem:

$$O = \|S - A\|_F^2 + \|W - D\|_F^2 \tag{22}$$

s.t. $\sum_j s_{ij} = 1, s_{ij} \geq 0, \sum_j w_{ij} = 1, w_{ij} \geq 0, rank(L_S) = n - k, rank(L_W) = m - k$

The above objective function, due to D_S depending on S and D_W depending on W , along with the constraints $rank(L_S) = n - k, rank(L_W) = m - k$, is evidently a complex nonlinear constrained problem. Next, we will reformulate this problem using Laplacian rank constraints.

Assume $\sigma_i(L_W)$ represents the i -th smallest eigenvalue of L_W , and $\sigma_i(L_S)$ represents the i -th smallest eigenvalue of L_S . Additionally, since L_W and L_S are both semi-definite matrices, their eigenvalues $\sigma_i(L_S)$ and $\sigma_i(L_W)$ are non-negative. It can be seen that for a sufficiently large value of λ , problem (22) is equivalent to the following problem:

$$\min_{S,W} J = \|S - A\|_F^2 + \|W - D\|_F^2 + 2\alpha \left(\sum_{i=1}^k \sigma_i(L_S) + \sum_{i=1}^k \sigma_i(L_W) \right) \tag{23}$$

s.t. $\sum_j s_{ij} = 1, s_{ij} \geq 0, \sum_j w_{ij} = 1, w_{ij} \geq 0$

Therefore, when λ is sufficiently large, the optimal solutions S and W to Equation (25) will make $\sum_{i=1}^k \sigma_i(L_S) + \sum_{i=1}^k \sigma_i(L_W)$ equal to zero, thereby satisfying the two rank constraints in problem (22). According to the Ky Fan theorem [38], the following equality can be obtained:

$$\sum_{i=1}^k \sigma_i(L_S) = \min_{F \in \mathbb{R}^{n \times k}} Tr(F^T L_S F), \sum_{i=1}^k \sigma_i(L_W) = \min_{F \in \mathbb{R}^{m \times k}} Tr(F^T L_W F) \tag{24}$$

Further, expression (23) can be rewritten in the following form:

$$\min_{S,W} J = \|S - A\|_F^2 + \|W - D\|_F^2 + \alpha (Tr(V^T L_S V) + Tr(U^T L_W U)) \tag{25}$$

$S1 = 1, W1 = 1, S \in \mathbb{R}^{n \times n}, W \in \mathbb{R}^{m \times m}$

It is worth noting that 1 is a vector with all elements equal to 1 . Next, we construct the optimal dual graph regularization term within the feature-weighted NMF framework. The objective function is written in the following form:

$$\min_{S,W} J = \|S - A\|_F^2 + \|W - D\|_F^2 + \alpha (Tr(V^T L_S V) + Tr(U^T L_W U)) + \|TX - UV^T\|_F^2 \tag{26}$$

s.t. $S1 = 1, W1 = 1, S \geq 0, W \geq 0, S \in \mathbb{R}^{n \times n}, W \in \mathbb{R}^{m \times m}, T = \text{diag}(t), t_i \geq 0,$
 $\sum_{i=1}^m t_i = 1, U \geq 0, V \geq 0$

Considering the sparsity constraints and orthogonality constraints in Equation (26), the objective function of the proposed SFLRNMF framework can be summarized as follows:

$$\begin{aligned}
 \min_{S,W} J &= \|S - A\|_F^2 + \|W - D\|_F^2 + \alpha(\text{Tr}(V^T L_S V) + \text{Tr}(U^T L_W U)) + \|TX - UV^T\|_F^2 + \theta \|U\|_{2,1/2}^{1/2} \\
 \text{s.t. } &V^T V = I, S1 = 1, W1 = 1, S \geq 0, W \geq 0, S \in \mathbb{R}^{n \times n}, W \in \mathbb{R}^{m \times m}, T_i = \text{diag}(t_i), t_i \geq 0, \\
 &\sum_{i=1}^m t_i = 1, U \geq 0, V \geq 0
 \end{aligned}
 \tag{27}$$

Here, α and θ are non-negative and balance the weight of the first reconstruction error term and the other terms, with θ being the sparsity parameter. T is a diagonal matrix, $T \in \mathbb{R}^{m \times m}$, which assigns a weight to each feature of the original matrix X .

3.6. An Efficient Iterative Update Rule for Solving the Proposed Model

To address this non-convex problem, by alternately optimizing the following variables, we can transform the Lagrangian function of the objective function (27) into the following equations:

$$\begin{aligned}
 \min_{S,W} J &= \|S - A\|_F^2 + \|W - D\|_F^2 + \alpha(\text{Tr}(V^T L_S V) + \text{Tr}(U^T L_W U)) + \text{Tr}(TXX^T T^T) - 2\text{Tr}(X^T T^T UV^T) \\
 &+ \text{Tr}(VU^T UV^T) + \beta \text{Tr}(V^T V - I) + 4\theta \text{Tr}(U^T QU)
 \end{aligned}
 \tag{28}$$

Here, $Q \in \mathbb{R}^{m \times m}$, and $Q = [q_{ij}]$ is a diagonal matrix. We can compute the diagonal element of its i -th row as follows:

$$q_{ij} = \frac{1}{4\max(\|u_i\|_2^{3/2}, \epsilon)}
 \tag{29}$$

Here, ϵ is a sufficiently small constant to avoid overflow in the above equation. To iteratively update the basis matrix U , the coefficient matrix V , and the feature weighting matrix T , we should take the partial derivatives of J :

$$\frac{\partial J}{\partial U} = 2\alpha L_W U - 2TXV + 2UV^T V + 8\theta QU
 \tag{30}$$

$$\frac{\partial J}{\partial V} = 2\alpha L_S V - 2X^T T^T U + 2VU^T U + 2\beta V
 \tag{31}$$

$$\frac{\partial J}{\partial T} = 2TXX^T - 2UV^T X^T
 \tag{32}$$

According to the Karush–Kuhn–Tucker (KKT) conditions, the iterative updates for the basis matrix U , the coefficient matrix V , and the feature weighting matrix T are as follows:

$$u_{ij} \leftarrow u_{ij} \frac{(TXV + \alpha W^W U)_{ij}}{(\alpha D^W U + UV^T V + 4\theta QU)_{ij}}
 \tag{33}$$

$$v_{ij} \leftarrow v_{ij} \frac{(X^T T^T U + \alpha W^S V)_{ij}}{(VU^T U + \beta V + \alpha D^S V)_{ij}}
 \tag{34}$$

$$t_{ij} \leftarrow t_{ij} \frac{(UV^T X^T)_{ij}}{(TXX^T)_{ij}}
 \tag{35}$$

To obtain the updates for the new data graph S and the new feature graph W , we use alternating optimization. First, we fix U, V, T , and W , then update S :

$$\begin{aligned}
 L_1 &= \|S - A\|_F^2 + \alpha \text{Tr}(V^T L_S V) \\
 S1 &= 1, S \geq 0, S \in \mathbb{R}^{n \times n}
 \end{aligned}
 \tag{36}$$

Equation (36) is equivalent to optimizing the following problem:

$$\min_{\sum_j s_{ij}=1, s_{ij} \geq 0} \sum_{i,j} (s_{i,j} - a_{i,j})^2 + \alpha \sum_{i,j} \|v_i - v_j\|_2^2 s_{i,j} \tag{37}$$

Note that for different i , the above problem is independent, so we can solve the above problem by solving for each i separately.

$$\min_{\sum_j s_{ij}=1, s_{ij} \geq 0} \sum_i (s_{i,j} - a_{i,j})^2 + \alpha \sum_i \|v_i - v_j\|_2^2 s_{i,j} \tag{38}$$

where definition $f_{i,j} = \|v_i - v_j\|_2^2$, and the j -th column element of $f_{i,j}$ is denoted by f_i (similarly for s_i and a_i). Problem (35) can be written in vector form as follows:

$$\sum_{s_i^T=1, s_i \geq 0} \left\| s_i - \left(a_i - \frac{\lambda}{2} f_i \right) \right\|_2^2 \tag{39}$$

Equation (39) can be solved using the simplex sparse learning model proposed by Huang et al. [39].

Next, update matrix W ; similarly, we can update W by fixing matrices U, V, T , and S as follows:

$$\begin{aligned} L_2 &= \|W - D\|_F^2 + \alpha \text{Tr}(U^T L_S U) \\ \text{s.t. } &W1 = 1, W \geq 0, W \in \mathbb{R}^{m \times m} \end{aligned} \tag{40}$$

Similarly, optimizing (40) is equivalent to optimizing the following problem:

$$\min_{\sum_j w_{ij}=1, w_{ij} \geq 0} \sum_{i,j} (w_{i,j} - d_{i,j})^2 + \alpha \sum_{i,j} \|u_i - u_j\|_2^2 w_{i,j} \tag{41}$$

For each row of w_i and d_i , we have

$$\min_{\sum_j w_{ij}=1, w_{ij} \geq 0} \sum_i (w_{i,j} - d_{i,j})^2 + \alpha \sum_i \|u_i - u_j\|_2^2 w_{i,j} \tag{42}$$

Similarly, we define $g_{i,j} = \|u_i - u_j\|_2^2$, and the j -th column element of $g_{i,j}$ is denoted by g_i (similarly for w_i and d_i). Problem (39) can be written in vector form as follows:

$$\sum_{w_i^T=1, w_i \geq 0} \left\| w_i - \left(d_i - \frac{\lambda}{2} g_i \right) \right\|_2^2 \tag{43}$$

For solving Equation (43), the simplex sparse learning model proposed by Huang et al. [39] is used.

The optimization process of the SFLRNMF algorithm is shown in Algorithm 1.

Algorithm 1 The process of SFLRNMF algorithm

Input: Initial Data similarity matrix S and Characteristic similarity matrix W , parameter α, θ, λ ; Data matrix $X = [x_1, x_2, \dots, x_n]$. the neighbor number k , and the maximum iteration number $Niter$.

Output: fundamental matrix U , coefficient matrix V

Initialization: $S \leftarrow S^0, W \leftarrow W^0, L_S = \text{Laplace}(S), L_W = \text{Laplace}(W)$

Repeat: Fixing S and W , update U, V and T

Update U by Equation (33)

Update V by Equation (34)

Update T by Equation (35)

Fixing U, V and T , update S and W

Update S by Equation (39)

Update W by Equation (43)

Compute L_S by $L_S = D_S - (S^T + S)/2, L_W$ by $L_W = D_W - (W^T + W)/2$

Until: convergence

3.7. Convergence Analysis of the SFLRNMF Algorithm

In this section, we analyze the convergence of SFLRNMF and prove that the objective function in Equation (27) is monotonically decreasing under the iterative update rules (33) to (35).

Firstly, we analyze the convergence of the iterative update rule in Equation (34).

Definition 1. Provided the following conditions are met [40]: $G(x, x') \geq F(x), G(x, x) = F(x)$, where $G(x, x')$ is an auxiliary function of $F(x)$. Assuming that the $(t + 1)$ -th iteration update rule is as follows:

$$x^{t+1} = \underset{x}{\operatorname{argmin}} G(x, x^t) \tag{44}$$

Therefore, we can prove that $(x^{t+1}) \leq G(x^{t+1}, x^t) \leq G(x^t, x^t) = F(x^t)$, which implies that $F(x)$ converges.

Lemma 1.

$$G(v_{ij}, v_{ij}^t) = F_{ij}(v_{ij}^t) + F'_{ij}(v_{ij}^t)(v_{ij} - v_{ij}^t) + \frac{(\alpha D^s V + V U^T U + \beta V)_{ij}}{v_{ij}^t} (v_{ij} - v_{ij}^t)^2 \tag{45}$$

The above equation is an auxiliary function of $F_{ij}(v_{ij})$, where $F(v) = \|TX - UV^T\|_F^2 + \alpha \operatorname{Tr}(V^T L_S V) + \beta (V^T V - I)$.

Proof. Given that the first and second derivatives of $F_{ij}(v_{ij})$ are $F'_{ij}(v) = (2\alpha L_S V - 2X^T T^T U + 2V U^T U + 2\beta V)_{ij}$ and $F''_{ij}(v) = 2\alpha (L_S)_{ii} + 2(U^T U)_{jj} + 2\beta$, we can derive the Taylor expansion of $F_{ij}(v_{ij})$ as follows:

$$F(v_{ij}) = F_{ij}(v_{ij}^t) + F'_{ij}(v_{ij}^t)(v_{ij} - v_{ij}^t) + \left[(U^T U)_{jj} + \alpha (L_S)_{ii} + \beta \right] (v_{ij} - v_{ij}^t)^2 \tag{46}$$

Since we have

$$\begin{cases} (V U^T U)_{ij} = \sum_{r=1}^k v_{ir}^t (U^T U)_{rj} \geq v_{ij}^t (U^T U)_{jj} \\ (\alpha D^s V)_{ij} = \alpha \sum_{r=1}^n D_{ir}^s v_{rj}^t \geq \alpha (D^s - W^s)_{ii} v_{ij}^t = \alpha (L_S)_{ii} v_{ij}^t \end{cases} \tag{47}$$

By simultaneously solving Equations (44) and (45), we obtain v_{ij}^{t+1} as the local minimum of Equation (45), and $G(v_{ij}^{t+1}, v_{ij}^t)$ as the corresponding local minimum value.

Given the equation $v_{ij}^{t+1} = v_{ij}^t - \frac{v_{ij}^t F'_{ij}(v_{ij}^t)}{2(\alpha D^s V + V U^T U + \beta V)_{ij}} = v_{ij}^t \frac{(X^T T^T U + \alpha W^s V)_{ij}}{(V U^T U + \beta V + \alpha D^s V)_{ij}}$, we have $\frac{(\alpha D^s V + V U^T U + \beta V)_{ij}}{v_{ij}^t} \geq \alpha (L_s)_{ii} + (U^T U)_{jj} + \beta$. Therefore, we can derive that $G(v_{ij}, v_{ij}^t) \geq F(v_{ij})$, Since $G(v_{ij}, v_{ij}^t)$ is an auxiliary function of $F_{ij}(v_{ij})$, $F(v_{ij})$ is monotonically decreasing.

In the same method, we can prove that under the iterative update rule (35), $F_{ij}(t_{ij})$ is monotonically decreasing. \square

Lemma 2 ([41]).

$$\sum_{i=1}^m \left(\|g_i^{t+1}\|_2^{1/2} - \frac{\|g_i^{t+1}\|_2^2}{4\|g_i^t\|_2^{3/2}} \right) \leq \sum_{i=1}^m \left(\|g_i^t\|_2^{1/2} - \frac{\|g_i^t\|_2^2}{4\|g_i^t\|_2^{3/2}} \right) \tag{48}$$

Proof. According to Lemma 2, we have the following equation:

$$\sum_{i=1}^m \left(\|u_i^{t+1}\|_2^{1/2} - \frac{\|u_i^{t+1}\|_2^2}{4\|u_i^t\|_2^{3/2}} \right) \leq \sum_{i=1}^m \left(\|u_i^t\|_2^{1/2} - \frac{\|u_i^t\|_2^2}{4\|u_i^t\|_2^{3/2}} \right) \tag{49}$$

In the i -th iteration, we fix Q as Q^T to solve for U^{t+1} , V^{t+1} , and T^{t+1} . We define the following function:

$$L(U, V, T) = \alpha [\text{Tr}(V^T L_s V) + \text{Tr}(U^T L_w U)] + \text{Tr}(T X X^T T^T) - 2\text{Tr}(X^T T^T U V^T) + \text{Tr}(V U^T U V^T) + \beta \text{Tr}(V^T V - I) \tag{50}$$

Given that $\|u\|_{2,1/2}^{1/2} = \sum_{i=1}^m \|u_i\|_2^{1/2}$, we obtain the following inequality:

$$\begin{aligned} L T^{t+1} + \theta \sum_{i=1}^m \frac{\|u_i^{t+1}\|_2^2}{4\|u_i^t\|_2^{3/2}} &= L T^{t+1} + \theta \|u^{t+1}\|_{2,1/2}^{1/2} + \theta \sum_{i=1}^m \left(\frac{\|u_i^{t+1}\|_2^2}{4\|u_i^t\|_2^{3/2}} - \|u_i^{t+1}\|_2^{1/2} \right) \\ &\leq L T^t + \theta \sum_{i=1}^m \frac{\|u_i^t\|_2^2}{4\|u_i^t\|_2^{3/2}} = L T^t + \theta \|u^t\|_{2,1/2}^{1/2} + \theta \sum_{i=1}^m \left(\frac{\|u_i^t\|_2^2}{4\|u_i^t\|_2^{3/2}} - \|u_i^t\|_2^{1/2} \right) \end{aligned} \tag{51}$$

Combining inequalities (48) and (51), we obtain the following inequality: $L^{t+1} + \theta \|U^{t+1}\|_{2,1/2}^{1/2} \leq L^t + \theta \|U^t\|_{2,1/2}^{1/2}$. Thus, $F_{ij}(u_{ij})$ is monotonically decreasing under the updated Equation (33). Based on the above convergence analysis, we can conclude that the objective function (27) is monotonically decreasing under the iterative update rules (33)–(35), (39), and (43). \square

4. Sparse Feature-Weighted Double Laplace Rank-Constrained NMTF Model

In this section, we propose a novel extension of SFLRNMF, called sparse feature-weighted dual Laplacian rank-constrained non-negative matrix tri-factorization (SFLRN-MTF). In SFLRNMTF, we introduce an additional factor R and dual orthogonality constraints to SFLRNMF. This not only enhances the accuracy of the low-rank representation but also ensures a more robust model by incorporating different scales of X , U , and V .

4.1. SFLRNMF with Three-Factor (SFLRNMTF)

To apply the idea of dual orthogonality constraints to SFLRNMF, we incorporate an additional factor R and dual orthogonality constraints into Equation (27) for SFLRNMF, as shown below:

$$\begin{aligned} \min_{S,W} JT &= \|S - A\|_F^2 + \|W - D\|_F^2 + \alpha(\text{Tr}(V^T L_S V) + \text{Tr}(U^T L_W U)) + \|TX - URV^T\|_F^2 + \theta \|U\|_{2,1/2}^{1/2} \\ \text{s.t. } U^T U &= I, V^T V = I, S1 = 1, W1 = 1, S \geq 0, W \geq 0, \quad S \in \mathbb{R}^{n \times n}, W \in \mathbb{R}^{m \times m}, \\ T_i &= \text{diag}(t), t_i \geq 0, \sum_{i=1}^m t_i = 1, U \geq 0, V \geq 0 \end{aligned} \tag{52}$$

Here, $R \in \mathbb{R}^{c \times c}$ is a diagonal scaling matrix, and T has the same meaning as in Equation (27).

4.2. An Efficient Iterative Update Rule for Solving Model (SFLRNMTF)

We rewrite the objective function (51) of the SFLRNMTF model as a Lagrangian function, as shown below:

$$\begin{aligned} JT &= \|S - A\|_F^2 + \|W - D\|_F^2 + \alpha(\text{Tr}(V^T L_S V) + \text{Tr}(U^T L_W U)) + \text{Tr}(TXX^T T^T) - \\ &2\text{Tr}(X^T T^T URV^T) + \text{Tr}(VR^T U^T URV^T) + \beta[\text{Tr}(U^T U - I) + \text{Tr}(V^T V - I)] + 4\theta \text{Tr}(U^T QU) \end{aligned} \tag{53}$$

To update the matrices U, V, R, and T, first, we compute the partial derivatives of the Lagrangian function JT with respect to different variables, as shown below:

$$\frac{\partial JT}{\partial U} = 2\alpha L_w U - 2TXVVR^T + 2URV^T VR^T + 2\beta U + 8\theta QU \tag{54}$$

$$\frac{\partial JT}{\partial V} = 2\alpha L_s V - 2X^T T^T UR + 2VR^T U^T UR + 2\beta V \tag{55}$$

$$\frac{\partial JT}{\partial R} = -2U^T TXV + 2U^T URV^T V \tag{56}$$

$$\frac{\partial JT}{\partial T} = 2TXX^T - 2URV^T X^T \tag{57}$$

Based on the Karush–Kuhn–Tucker (KKT) conditions [39], the iterative updates for matrices U, V, R, and T are as follows:

$$u_{ij} \leftarrow u_{ij} \frac{(TXVR^T + \alpha W^w U)_{ij}}{(\alpha D^w V + URV^T VR^T + \beta U + 4\theta QU)_{ij}} \tag{58}$$

$$v_{ij} \leftarrow v_{ij} \frac{(X^T T^T UR + \alpha W^s V)_{ij}}{(\alpha D^s V + VR^T U^T UR + \beta V)_{ij}} \tag{59}$$

$$r_{ij} \leftarrow r_{ij} \frac{(U^T TXV)_{ij}}{(U^T URV^T V)_{ij}} \tag{60}$$

$$t_{ij} \leftarrow t_{ij} \frac{(URV^T X^T)_{ij}}{(TXX^T)_{ij}} \tag{61}$$

The optimization process for SFLRNMTF is shown in Algorithm 2.

Algorithm 2 The process of SFLRNMTF algorithm

Input: Initial Data similarity matrix S and Characteristic similarity matrix W , parameter α, θ, λ ; Data matrix $X = [x_1, x_2, \dots, x_n]$. the neighbor number k , and the maximum iteration number Niter.

Output: fundamental matrix U , coefficient matrix V

Initialization: $S \leftarrow S^0, W \leftarrow W^0, L_S = \text{Laplace}(S), L_W = \text{Laplace}(W)$

Repeat: Fixing S and W , update U, V and T

Update U by Equation (58)

Update V by Equation (59)

Update R by Equation (60)

Update T by Equation (61)

Fixing U, V and T , update S and W

Update S by Equation (39)

Update W by Equation (43)

Compute L_S by $L_S = D_S - (S^T + S)/2, L_W$ by $L_W = D_W - (W^T + W)/2$

Until: convergence

4.3. Convergence Analysis of the SFLRNMTF Algorithm

In this section, we theoretically prove the convergence of SFLRNMTF and demonstrate that the objective function in Equation (52) is monotonically decreasing under the iterative update rules (58)–(61).

Definition 2. If the following conditions proposed by Shang et al. [40] are satisfied, that is, $GT(x, x') \geq FT(x), GT(x, x) = FT(x)$, and $GT(x, x')$ is an auxiliary function of $FT(x)$, then for iteration $(t + 1)$, the update rule is as follows:

$$x^{t+1} = \underset{x}{\operatorname{argmin}} GT(x, x^t) \tag{62}$$

Thus, it can be proven that $FT(x^{t+1}) \leq GT(x^{t+1}, x^t) \leq GT(x^t, x^t) = FT(x^t)$ converges.

Lemma 3.

$$T(v_{ij}, v_{ij}^t) = FT_{ij}(v_{ij}^t) + FT'_{ij}(v_{ij}^t)(v_{ij} - v_{ij}^t) + \frac{(\alpha D^S V + VR^T U^T UR + \beta V)}{v_{ij}^t} (v_{ij} - v_{ij}^t)^2 \tag{63}$$

The above equation is an auxiliary function for $FT_{ij}(v_{ij})$, where $FT(V) = \|TX - URV^T\|_F^2 + \alpha \operatorname{Tr}(V^T L_S V) + \beta(V^T V - I)$.

Proof. First, the first-order derivative of $FT_{ij}(v_{ij})$ is $FT'_{ij}(V) = (2\alpha L_S V - 2X^T T^T UR + 2VR^T U^T UR + 2\beta V)_{ij}$, and the second-order derivative of $FT_{ij}(v_{ij})$ is $FT''_{ij}(V) = 2\alpha(L_S)_{ii} + 2(R^T U^T UR)_{jj} + 2\beta$.

Thus, we can rewrite $FT_{ij}(v_{ij})$ in the following Taylor series form:

$$FT(v_{ij}) = FT_{ij}(v_{ij}^t) + FT'_{ij}(v_{ij}^t)(v_{ij} - v_{ij}^t) + \left[\alpha(L_S)_{ii} + (R^T U^T UR)_{jj} + \beta \right] (v_{ij} - v_{ij}^t)^2 \tag{64}$$

Given

$$\begin{cases} (VR^T U^T UR)_{ij} = \sum_{r=1}^k v_{ir}^t (R^T U^T UR)_{rj} \geq v_{ir}^t (R^T U^T UR)_{ii} \\ \beta = \beta \\ \alpha(D^S v)_{ij} = \alpha \sum_{r=1}^n D_{ir}^S v_{rj}^t \geq \alpha(D_{ii}^S - w_{ii}^S) v_{ij}^t = \alpha(L_S)_{ii} v_{ij}^t \end{cases} \tag{65}$$

By solving Equations (60) and (61) simultaneously, we find that $GT_{ij}(v_{ij}^{t+1}, v_{ij}^t)$ is a local minimum of Equation (61), and v_{ij}^{t+1} is the local minimum point corresponding to this local minimum.

$$v_{ij}^{t+1} = v_{ij}^t - \frac{v_{ij}^t FT_{ij}'(v_{ij}^t)}{2(D^s V + VR^T U^T UR + \beta V)} = v_{ij}^t \frac{(X^T T^T UR + \alpha W^s V)_{ij}}{(\alpha D^s V + VR^T U^T UR + \beta V)} \tag{66}$$

we can derive that $GT(v_{ij}, v_{ij}^t) \geq FT(v_{ij})$. Since $GT(v_{ij}, v_{ij}^t)$ is an auxiliary function of $FT_{ij}(v_{ij})$, $FT_{ij}(v_{ij})$ is monotonically decreasing.

In the same method, we can prove that under the iterative update rule (35), $FT_{ij}(t_{ij})$, and $FT_{ij}(r_{ij})$ are monotonically decreasing. □

Lemma 4 ([41]).

$$\sum_{i=1}^m \left(\|g_i^{t+1}\|_2^{1/2} - \frac{\|g_i^{t+1}\|_2^2}{4\|g_i^t\|_2^{3/2}} \right) \leq \sum_{i=1}^m \left(\|g_i^t\|_2^{1/2} - \frac{\|g_i^t\|_2^2}{4\|g_i^t\|_2^{3/2}} \right) \tag{67}$$

According to Lemma 4, we have

$$\sum_{i=1}^m \left(\|u_i^{t+1}\|_2^{1/2} - \frac{\|u_i^{t+1}\|_2^2}{4\|u_i^t\|_2^{3/2}} \right) \leq \sum_{i=1}^m \left(\|u_i^t\|_2^{1/2} - \frac{\|u_i^t\|_2^2}{4\|u_i^t\|_2^{3/2}} \right) \tag{68}$$

To solve for $U^{t+1}, V^{t+1}, T^{t+1}$, and R^{t+1} , we set Q to Q^t in the i -th generation. We define the following function:

$$LT(U, V, T, R) = \lambda [\text{Tr}(V^T L_s V) + \text{Tr}(U^T L_w U)] + \text{Tr}(T X X^T T^T) - 2\text{Tr}(X^T T^T U R V^T) + \text{Tr}(V R^T U^T U R V^T) + \beta [\text{Tr}(U^T U - I) - \text{Tr}(V^T V - I)] \tag{69}$$

Since $\|U\|_{2,1/2} = \sum_{i=1}^m \|u_i\|_2^{1/2}$, we obtain the following inequality:

$$\begin{aligned} LT^{t+1} + \theta \sum_{i=1}^m \frac{\|u_i^{t+1}\|_2^2}{4\|u_i^t\|_2^{3/2}} &= LT^{t+1} + \theta \|U^{t+1}\|_{2,1/2}^{1/2} + \theta \sum_{i=1}^m \left(\frac{\|u_i^{t+1}\|_2^2}{4\|u_i^t\|_2^{3/2}} - \|u_i^{t+1}\|_2^{1/2} \right) \leq LT^t \\ + \theta \sum_{i=1}^m \frac{\|u_i^t\|_2^2}{4\|u_i^t\|_2^{3/2}} &= LT^t + \theta \|U^t\|_{2,1/2}^{1/2} + \theta \sum_{i=1}^m \left(\frac{\|u_i^t\|_2^2}{4\|u_i^t\|_2^{3/2}} - \|u_i^t\|_2^{1/2} \right) \end{aligned} \tag{70}$$

Combining inequalities (67) and (70), we obtain the following inequality:

$$LT^{t+1} + \theta \|U^{t+1}\|_{2,1/2}^{1/2} \leq LT^t + \theta \|U^t\|_{2,1/2}^{1/2} \tag{71}$$

Thus, $FT_{ij}(u_{ij})$ is monotonically decreasing. Based on the above analysis, we can conclude that the objective function (52) of SFLRNMTF is monotonically decreasing under the update rules in Equations (39), (43), and (58)–(61).

5. Experiments and Analysis

In this subsection, to verify the efficiency of the proposed methods (SFLRNMF and SFLRNMTF), we conducted numerical experiments on four datasets (COIL20, JAFFE, UMIST, and YaleB32) to evaluate their robustness, sensitivity, and convergence. We compared the clustering performance with eight algorithms, including PCA [8], NMF [15,16],

GNMF [22], DNMF [23], DSNMF-LDC [42], NMFAN [30], LOSDNMF [27], EWNMF [43], SGLNMF [26], and SDGNMF-BO [25].

All numerical experiments were conducted on a PC with Windows 10 operating system, a CPU of 3.40 GHz, and 8GB of memory, using the Matlab 2021a platform. The specific characteristics of these datasets are shown in Table 2.

Table 2. Benchmark dataset.

Datasets	Samples	Dimensions	Classes	Type
COIL-20	1440	1024	20	Object
JAFFE	213	1024	10	Face
UMIST	575	1024	20	Face
YaleB	2414	1024	38	Face

5.1. Datasets

- (1) COIL-20 Dataset: The COIL-20 dataset was developed by Columbia University and contains images of 20 different real objects. The dataset comprises a total of 1,440 images.
- (2) JAFFE Dataset: This dataset involves Japanese female facial expressions from ten female volunteers, with each category containing seven facial expressions. Each facial emotion of each volunteer was photographed two to four times, resulting in a total of 213 images.
- (3) UMIST Dataset: This dataset was constructed by the University of Manchester Institute of Science and Technology (UMIST) and includes facial images of 20 individuals with varying poses, races, genders, and appearances. The dataset contains a total of 575 images.
- (4) YaleB Dataset: This dataset contains 2414 frontal face images taken under controlled lighting conditions in a laboratory, representing 38 subjects. Each image was manually cropped to a 32×32 pixel grayscale image and then converted into a 1024-dimensional vector.

5.2. Parameter Setting

In this section, to ensure fairness, we employ k-means as the clustering method to compare the clustering performance of SFLRNMF, SFLRNMTF, and eight other algorithms across four datasets. For our SFLRNMF and SFLRNMTF models, the primary parameters involve the dual Laplacian graph rank constraint parameter α , the sparsity parameter θ , and the orthogonality constraint parameter β . The range for all regularization parameters is set within $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10^1, 10^2, 10^3, 10^4, 10^5, 10^6\}$. The maximum number of iterations is set to 200, and the best average result from 10 experiments is considered the final clustering result. For all graph-based methods, the number of nearest neighbors k is fixed at five. For all semi-supervised NMF methods, in each dataset, 20% of the data points from each category are randomly selected as labeled samples, and these points are used to construct the label constraint matrix A . Meanwhile, to verify the performance of our proposed algorithms on different classification metrics, we randomly select the number of clusters c within the range of 2 to 10 for clustering experiments on the Jaffe and YaleB datasets. For clustering experiments on the COIL20 and UMIST datasets, the range for the number of clusters c is set from 10 to 20.

Furthermore, due to the multitude of algorithms compared, to enhance the clarity of reading for the readers, in the clustering experiments, the names of all NMF variant algorithms have been abbreviated as follows: GNMF is abbreviated as G, DSNMF-LDC as LDC, NMFAN as AN, DNMF as D, LOSDNMF as LOSD, EWNMF as EW, SGLNMF as SGL, SDGNMF-BO as SDG, SFLRNMF as SFLR, and SFLRNMTF as SFLRT. This treatment helps in making the understanding and comparison of each algorithm clearer to the readers.

5.3. Clustering Performance

Tables 3–10 show the clustering results in terms of ACC and NMI values for the four datasets. The results in bold represent the optimal results. The corresponding clustering results are shown in Figures 2–9.

Table 3. ACC results for clustering of SFLRNMF, SFLRNMTF and ten other algorithms on the JAFFE dataset.

c	PCA	NMF	G	LDC	AN	D	LOSD	EW	SGL	SDG	SFLR	SFLRT
2	99.10	98.58	99.02	100	99.01	100	100	100	100	99.15	100	100
3	97.78	93.32	98.15	100	98.19	98.02	98.29	97.18	98.89	98.96	99.04	99.65
4	95.73	93.25	96.32	99.03	95.93	97.31	98.72	95.55	98.60	97.86	98.29	98.36
5	93.66	92.18	96.14	96.85	97.17	96.39	96.22	95.38	97.20	97.93	97.08	98.04
6	93.24	91.26	92.45	94.49	94.73	93.42	95.63	93.34	94.63	96.82	97.14	98.15
7	92.31	89.32	95.22	96.78	95.93	93.19	95.58	92.47	94.35	95.46	96.58	97.72
8	90.42	85.15	90.54	92.23	92.18	91.58	94.59	89.54	93.36	94.00	96.55	97.31
9	88.32	84.58	90.27	91.97	93.02	91.45	93.87	88.92	94.17	94.79	95.52	97.26
10	84.93	83.02	90.19	90.05	90.52	90.55	92.05	88.50	93.05	92.16	95.31	96.97
Avg.	92.83	90.07	94.26	95.71	95.19	94.66	96.11	93.43	96.03	96.35	97.28	98.16

Table 4. NMI results for clustering of SFLRNMF, SFLRNMTF and ten other algorithms on the JAFFE dataset.

c	PCA	NMF	G	LDC	AN	D	LOSD	EW	SGL	SDG	SFLR	SFLRT
2	95.62	92.07	90.18	100	95.79	100	100	100	100	96.21	100	100
3	93.02	89.14	94.21	97.25	95.12	94.56	95.02	91.89	96.82	97.85	97.01	99.06
4	91.51	87.56	91.36	98.13	88.65	93.75	96.68	90.10	96.46	94.71	96.14	96.11
5	89.79	86.28	93.44	95.79	93.63	91.45	92.29	90.01	94.23	95.44	95.09	96.03
6	89.13	84.19	86.56	92.15	94.46	89.25	92.23	90.90	90.34	94.16	94.55	96.09
7	89.49	83.37	91.00	95.10	95.44	91.46	92.15	88.46	90.51	92.12	95.41	96.05
8	87.89	80.46	89.01	94.02	93.68	90.35	93.03	86.12	91.00	93.10	95.33	96.05
9	86.09	81.55	90.06	91.53	95.11	92.89	91.69	86.51	92.12	93.06	94.59	96.02
10	83.83	80.31	88.09	92.23	93.75	90.71	90.69	85.48	92.45	91.78	94.92	95.82
Avg.	89.60	85.00	90.43	95.13	93.96	92.71	93.75	89.94	93.77	94.27	95.89	96.80

Table 5. ACC results for clustering of SFLRNMF, SFLRNMTF and ten other algorithms on the COIL20 dataset.

c	PCA	NMF	G	LDC	AN	D	LOSD	EW	SGL	SDG	SFLR	SFLRT
10	77.14	71.24	86.27	91.93	73.96	87.85	89.56	73.46	85.65	88.76	96.04	94.57
11	74.68	68.13	83.80	88.06	68.89	87.88	86.62	71.07	85.43	85.77	95.29	95.51
12	74.07	64.89	83.51	88.91	69.13	86.23	84.25	70.85	85.13	85.93	93.19	94.39
13	74.07	71.95	81.09	87.99	68.09	82.50	83.79	69.65	83.44	87.04	92.71	95.09
14	73.82	69.23	82.45	83.54	70.15	82.41	81.49	69.41	82.89	83.62	94.02	94.15
15	72.06	66.69	82.15	83.08	67.45	83.39	81.43	69.28	80.90	82.92	93.76	90.34
16	72.93	65.55	79.53	85.59	65.76	83.72	81.05	67.04	79.90	81.88	89.08	88.15
17	71.16	65.15	78.91	82.12	64.39	80.57	80.90	65.95	78.99	81.32	88.24	88.82
18	71.13	63.11	82.20	82.09	66.93	81.06	79.47	65.52	81.62	80.76	86.38	88.34
19	70.91	64.76	80.13	77.01	65.21	79.50	79.33	64.95	78.44	79.35	84.67	88.11
20	68.78	64.32	79.69	77.98	65.35	81.83	78.61	64.87	79.09	79.51	86.04	87.75
Avg.	72.80	66.82	81.79	84.39	67.76	83.36	82.41	68.37	81.95	83.35	90.86	91.38

Table 6. NMI results for clustering of SFLRNMF, SFLRNMTF and ten other algorithms on the COIL20 dataset.

c	PCA	NMF	G	LDC	AN	D	LOSD	EW	SGL	SDG	SFLR	SFLRT
10	80.38	72.15	88.09	92.79	75.89	92.61	94.12	75.71	89.58	92.27	97.89	96.14
11	78.72	71.36	83.95	92.31	72.25	93.74	89.98	74.50	86.96	87.62	97.71	97.52
12	78.01	68.45	83.34	91.79	73.16	91.55	88.88	73.24	87.99	90.40	96.25	98.28
13	78.87	75.38	81.29	92.55	74.36	87.95	90.76	74.19	89.12	91.64	95.93	98.39
14	79.67	75.06	82.16	88.74	77.42	87.90	88.05	74.18	89.82	89.36	95.05	97.28
15	78.81	73.58	82.59	89.40	75.15	89.67	89.61	74.54	88.17	88.69	95.12	94.43
16	78.65	73.42	88.43	87.42	73.27	89.91	90.12	74.71	87.45	88.61	93.23	94.29
17	78.67	73.19	87.32	89.89	73.19	88.91	89.69	72.56	87.26	88.67	95.36	95.71
18	78.03	73.31	89.95	89.09	74.32	89.30	88.40	73.59	87.43	88.24	95.15	95.08
19	78.71	73.42	88.46	86.85	74.25	88.82	88.89	72.88	87.98	88.65	93.23	94.21
20	68.78	74.55	89.79	86.98	75.65	89.76	89.09	72.95	88.10	88.55	95.42	94.15
Avg.	77.94	73.08	85.94	89.80	74.45	90.01	89.78	73.91	88.17	89.34	95.49	95.95

Table 7. ACC results for clustering of SFLRNMF, SFLRNMTF and ten other algorithms on the UMIST dataset.

c	PCA	NMF	G	LDC	AN	D	LOSD	EW	SGL	SDG	SFLR	SFLRT
10	53.49	48.95	65.55	75.19	50.91	66.09	69.25	50.36	67.97	68.40	77.12	78.06
11	52.37	48.46	64.53	72.25	50.85	65.67	68.30	46.97	66.49	65.90	77.12	78.19
12	54.10	47.31	63.71	71.34	47.99	65.78	64.88	45.89	62.19	63.76	75.28	77.21
13	50.11	47.98	60.43	70.16	47.03	60.96	63.56	46.07	62.15	63.49	74.19	77.16
14	49.13	45.79	60.55	69.01	45.53	61.52	62.86	44.93	61.36	64.15	75.27	75.63
15	48.26	44.01	59.29	68.36	44.57	59.91	61.25	45.14	61.96	61.93	74.29	75.27
16	46.74	42.25	59.46	69.16	44.48	60.22	61.07	44.16	58.09	62.00	76.06	75.19
17	45.84	41.46	60.39	69.09	43.19	60.40	60.01	42.79	59.60	62.13	73.01	74.46
18	46.12	41.29	59.34	66.14	43.24	59.52	59.23	42.46	57.69	63.24	72.53	74.25
19	44.90	41.35	59.48	62.23	43.35	59.78	59.09	41.09	56.67	59.17	72.11	74.21
20	44.21	41.59	58.98	59.38	42.77	59.10	59.70	41.23	57.41	59.65	71.14	69.09
Avg.	48.66	44.59	61.16	68.39	45.81	61.72	62.56	44.64	61.05	63.07	74.37	75.34

Table 8. NMI results for clustering of SFLRNMF, SFLRNMTF and ten other algorithms on the UMIST dataset.

c	PCA	NMF	G	LDC	AN	D	LOSD	EW	SGL	SDG	SFLR	SFLRT
10	65.89	56.79	76.41	83.01	60.91	76.26	78.88	59.52	78.34	77.25	86.01	84.09
11	66.12	59.12	76.47	84.78	60.96	77.35	78.48	57.79	77.69	76.63	85.23	87.27
12	67.29	58.18	75.23	82.46	57.89	78.70	75.43	57.56	76.21	75.96	85.04	85.18
13	65.49	57.16	74.88	82.54	60.71	75.11	76.23	60.03	73.78	76.27	84.09	83.41
14	66.33	58.59	75.96	81.78	59.08	74.29	76.37	58.69	76.82	76.33	83.18	84.05
15	66.14	58.78	75.89	82.13	59.14	74.90	76.13	59.00	76.26	75.95	86.33	85.19
16	66.05	58.93	74.65	84.06	61.78	76.57	76.07	58.74	73.80	75.95	84.52	86.79
17	65.48	58.95	76.43	83.25	59.16	76.72	76.02	58.60	75.04	75.63	84.45	85.14
18	66.02	58.46	75.29	82.03	61.25	74.53	75.96	59.29	74.32	77.24	85.04	83.21
19	65.40	58.29	75.13	80.79	60.29	75.12	75.61	58.19	73.02	75.93	85.13	83.13
20	65.50	58.01	75.59	79.15	60.37	75.07	75.91	59.23	74.46	76.08	84.18	81.16
Avg.	65.97	58.30	75.63	82.36	60.14	75.88	76.46	58.79	75.43	76.29	84.84	84.42

Table 9. ACC results for clustering of SFLRNMF, SFLRNMTF and ten other algorithms on the YaleB dataset.

c	PCA	NMF	G	LDC	AN	D	LOSD	EW	SGL	SDG	SFLR	SFLRT
2	52.52	51.65	79.13	79.53	58.50	80.14	86.36	52.87	81.72	73.82	91.64	88.39
3	37.51	36.71	56.82	75.27	43.12	65.48	60.53	36.99	67.57	59.06	78.56	78.66
4	29.54	27.80	54.08	57.70	34.76	54.42	58.44	29.78	61.05	57.36	69.56	72.00
5	26.26	24.04	47.69	55.06	30.42	51.87	51.68	26.51	53.50	57.67	61.04	67.88
6	22.73	20.33	46.65	47.44	26.86	45.87	44.62	25.10	51.74	48.45	59.47	64.78
7	21.27	18.62	42.93	44.67	24.38	37.30	41.74	25.23	46.15	45.52	59.42	48.95
8	20.98	18.49	41.70	43.63	22.53	36.00	39.54	23.43	47.37	42.38	56.20	45.87
9	18.49	18.15	39.62	40.52	21.65	34.62	35.16	22.15	44.81	42.26	54.09	55.48
10	17.23	17.53	39.84	21.74	19.98	31.25	33.66	22.05	42.54	41.66	55.24	49.43
Avg.	27.39	25.92	49.83	51.73	31.36	48.55	50.19	29.35	55.16	52.02	65.02	63.49

Table 10. NMI results for clustering of SFLRNMF, SFLRNMTF and ten other algorithms on the YaleB dataset.

c	PCA	NMF	G	LDC	AN	D	LOSD	EW	SGL	SDG	SFLR	SFLRT
2	0.28	0.11	45.74	38.44	3.65	44.04	63.50	0.28	45.55	41.19	81.56	81.01
3	0.84	0.47	27.12	44.85	3.73	37.83	32.84	0.91	41.32	34.37	64.41	67.11
4	1.50	0.67	36.88	37.59	4.38	23.81	32.05	1.15	44.39	45.47	58.47	58.95
5	4.11	2.30	31.64	43.57	5.04	40.55	40.12	3.76	43.81	48.22	50.24	57.48
6	3.80	1.15	39.13	27.40	5.79	39.72	36.77	7.66	42.18	39.26	55.08	59.01
7	6.27	2.60	36.46	34.78	6.31	29.28	36.71	11.80	42.73	41.85	58.38	45.99
8	7.38	3.93	36.03	41.86	6.33	34.72	36.54	11.86	45.79	39.47	53.95	40.83
9	7.62	6.93	39.83	33.01	7.18	34.19	35.67	15.81	46.26	40.03	54.08	54.91
10	7.10	8.49	41.06	15.59	7.78	31.72	35.50	17.59	42.89	41.14	53.45	48.54
Avg.	4.32	2.96	37.10	35.23	5.58	35.10	38.86	7.87	43.88	41.22	58.85	57.09

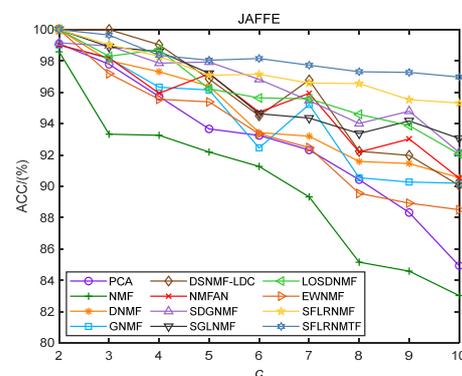


Figure 2. Clustering ACC on the dataset JAFFE.

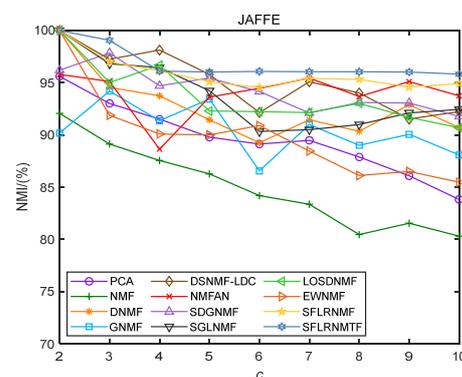


Figure 3. Clustering NMI on the dataset JAFFE.

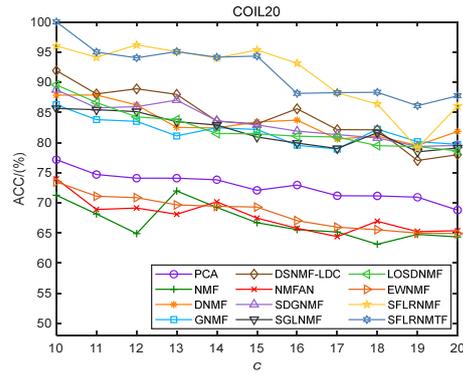


Figure 4. Clustering ACC on the dataset COIL20.

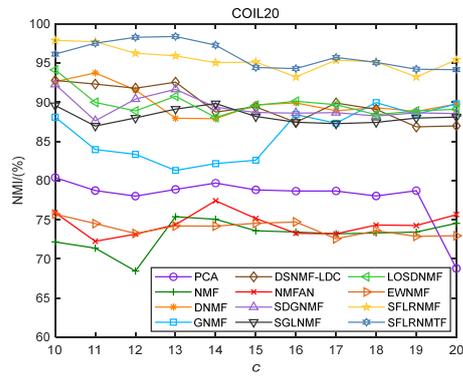


Figure 5. Clustering NMI on the dataset COIL20.

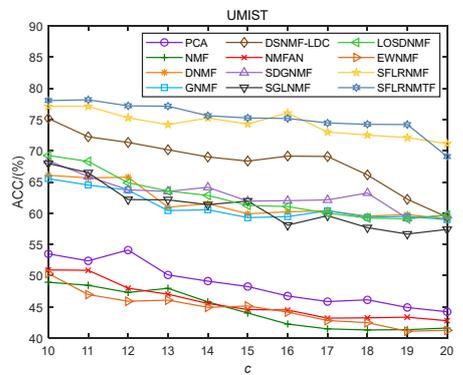


Figure 6. Clustering ACC on the dataset UMIST.

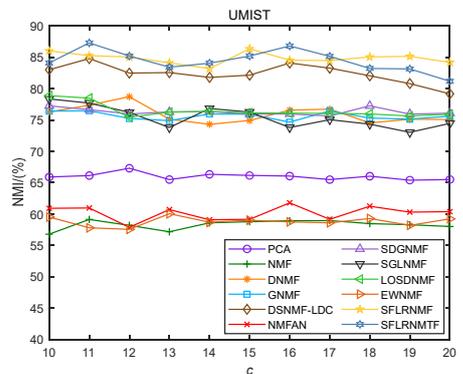


Figure 7. Clustering NMI on the dataset UMIST.

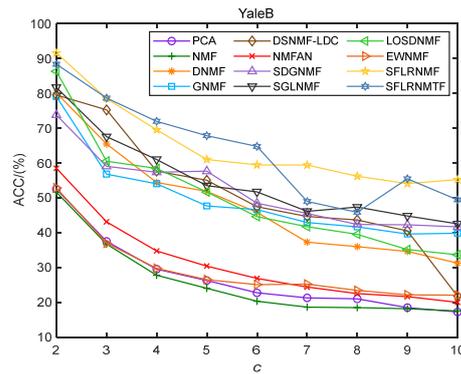


Figure 8. Clustering ACC on the dataset YaleB.

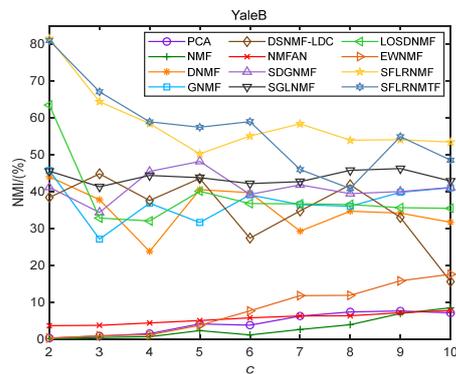


Figure 9. Clustering NMI on the dataset YaleB.

Tables 2 and 3, along with Figures 2 and 3, display the clustering results on the JAFFE dataset. As shown, some algorithms achieved 100% in both ACC and NMI when the number of clusters was set to two. However, as the number of clusters increased, our algorithms, SFLRNMF and SFLRNMTF, consistently performed well across most clustering settings, achieving average accuracy rates (ACC) of 97.28% and 95.89% and average Normalized Mutual Information (NMI) of 98.16% and 96.80%, respectively. Compared to the lowest-performing NMF algorithm, SFLR and SFLRT showed an average increase in accuracy of 7.21% and 8.09% and in NMI of 10.89% and 11.8%, respectively. These results demonstrate that our methods have a significant advantage in processing facial recognition images.

Tables 4 and 5, along with Figures 4 and 5, present the clustering results on the COIL20 dataset. These data clearly show that, compared to ten other benchmark algorithms, our algorithms achieved satisfactory results. Specifically, in terms of average ACC scores, DSNMF-LDC’s scores are second only to ours, and in terms of average NMI scores, DNMF’s scores also rank just below ours. More precisely, our SFLRNMF and SFLRNMTF algorithms exceed the average ACC of DSNMF-LDC by 6.47% and 6.99%, respectively, and surpass DNMF’s average NMI by 5.48% and 5.94%, respectively. Moreover, as observed from Figures 4 and 5, regardless of the number of clusters, the highest ACC and NMI scores consistently appear in our two algorithms. These results amply demonstrate the superiority and stability of our algorithms.

The clustering results on the UMIST dataset are shown in Tables 6 and 7, as well as Figures 4 and 5. According to Tables 6 and 7, it can be observed that in terms of ACC, the SFLRNMTF algorithm generally performs the best among the majority of the clustering settings, followed by the SFLRNMF algorithm. In terms of NMI, the SFLRNMTF algorithm also exhibits high performance, with an average score of 84.42%, while the average score for the SFLRNMF algorithm is 84.84%. Compared to other algorithms, such as the poorest performing NMF and the best performing DSNMF-LDC, our SFLRNMTF algorithm’s average ACC is higher by 30.75% and 6.95%, respectively. In terms of NMI,

the average NMI of the SFLRNMF algorithm is higher than that of NMF and DSNMF-LDC by 26.54% and 2.48%, respectively. It is noteworthy that although the average ACC score of SFLRNMTF is higher than that of SFLRNMF, its average NMI score is lower than that of SFLRNMF. The reason for this phenomenon is that NMI is actually a measure of the joint probability distribution across different classes, which means it calculates by comparing the mutual information and entropy between generated labels and true labels. Therefore, the more points that are misclassified into the same class, the higher the NMI score at the same ACC value. This explains why SFLRNMTF has a higher average ACC score than SFLRNMF, yet a slightly lower average NMI score. Furthermore, as depicted in Figures 4 and 5, regardless of the number of clusters, the ACC and NMI curves of SFLRNMF and SFLRNMTF consistently remain above those of the other algorithms compared. These results indicate that the SFLRNMF and SFLRNMTF algorithms proposed by us not only produce significant effects but also demonstrate strong stability when processing facial recognition images in the UMIST dataset.

Similarly, the clustering performance on the YaleB dataset is presented in Tables 8 and 9, as well as in Figures 8 and 9. In terms of accuracy, the SFLRNMF algorithm was found to perform best in most clustering setups, especially when the number of clusters was set to 2, achieving a high score of 91.64%, significantly surpassing the other algorithms. On average, an ACC score of 65.02% was obtained by SFLRNMF, with SFLRNMTF closely following at an average score of 63.49%. In contrast, the worst performing algorithm was NMF, which only achieved an average ACC score of 27.39%. In terms of Normalized Mutual Information (NMI), high scores were also demonstrated by SFLRNMTF, particularly when the number of clusters was 2, reaching an NMI score of 81.56%. Overall, the average NMI score for SFLRNMTF was 58.85%, slightly higher than that of SFLRNMF, which was 57.09%. Compared to LOSDNMF, which exhibited poorer clustering performance, both SFLRNMF and SFLRNMTF achieved higher scores in ACC and NMI across various numbers of clusters. Furthermore, as observed in Figures 8 and 9, although the ACC and NMI curves for all algorithms showed a downward trend with an increasing number of clusters, the curves for SFLRNMF and SFLRNMTF consistently remained above those of the other comparative algorithms. These results indicate that the proposed SFLRNMF and SFLRNMTF algorithms not only produce significant effects in handling facial recognition images from the YaleB dataset but also exhibit stability across various cluster number settings. Particularly, the SFLRNMF algorithm showed the most stable performance among all compared algorithms.

Similarly, the clustering performance on the YaleB dataset is demonstrated in Tables 8 and 9, as well as Figures 8 and 9. In terms of accuracy, the SFLRNMTF algorithm performs best under most clustering counts. For accuracy (ACC), the average ACC score of SFLRNMF is 65.02%, followed closely by SFLRNMTF with an average score of 63.49%. In contrast, the worst performing algorithm is NMF, with an average ACC score of only 27.39%. In terms of Normalized Mutual Information (NMI), SFLRNMF also shows higher scores, particularly when the cluster count is 2, reaching an NMI score of 81.56%. Overall, the average NMI score of SFLRNMF is 58.85%, slightly higher than that of SFLRNMTF at 57.09%. Compared to LOSDNMF, which has poorer clustering performance, both SFLRNMF and SFLRNMTF outperform in both ACC and NMI scores across various numbers of clusters. Furthermore, as shown in Figures 8 and 9, although the ACC and NMI curves for all algorithms tend to decline as the number of clusters increases, the curves for SFLRNMF and SFLRNMTF consistently remain above those of the other comparative algorithms. These results indicate that the proposed SFLRNMF and SFLRNMTF algorithms not only produce significant effects in handling facial recognition images from the YaleB dataset but also exhibit stability across different cluster settings. Notably, the SFLRNMF algorithm shows the most stable performance among all compared algorithms.

5.4. Visualization Comparison

Based on the clustering comparison experiments mentioned above, we plan to delve deeper into the superiority of our two algorithms through their subspace learning capa-

bilities. For a more intuitive and effective comparison, we selected 10 categories from four test datasets as the input data for the experiments (since too many data categories might create visual clutter that could hinder analysis, while too few categories might not adequately highlight the subspace learning performance of different algorithms). The visual comparison results are displayed in Figures 10 and 11. Note that the subgraphs containing category labels do not obscure any comparative information.

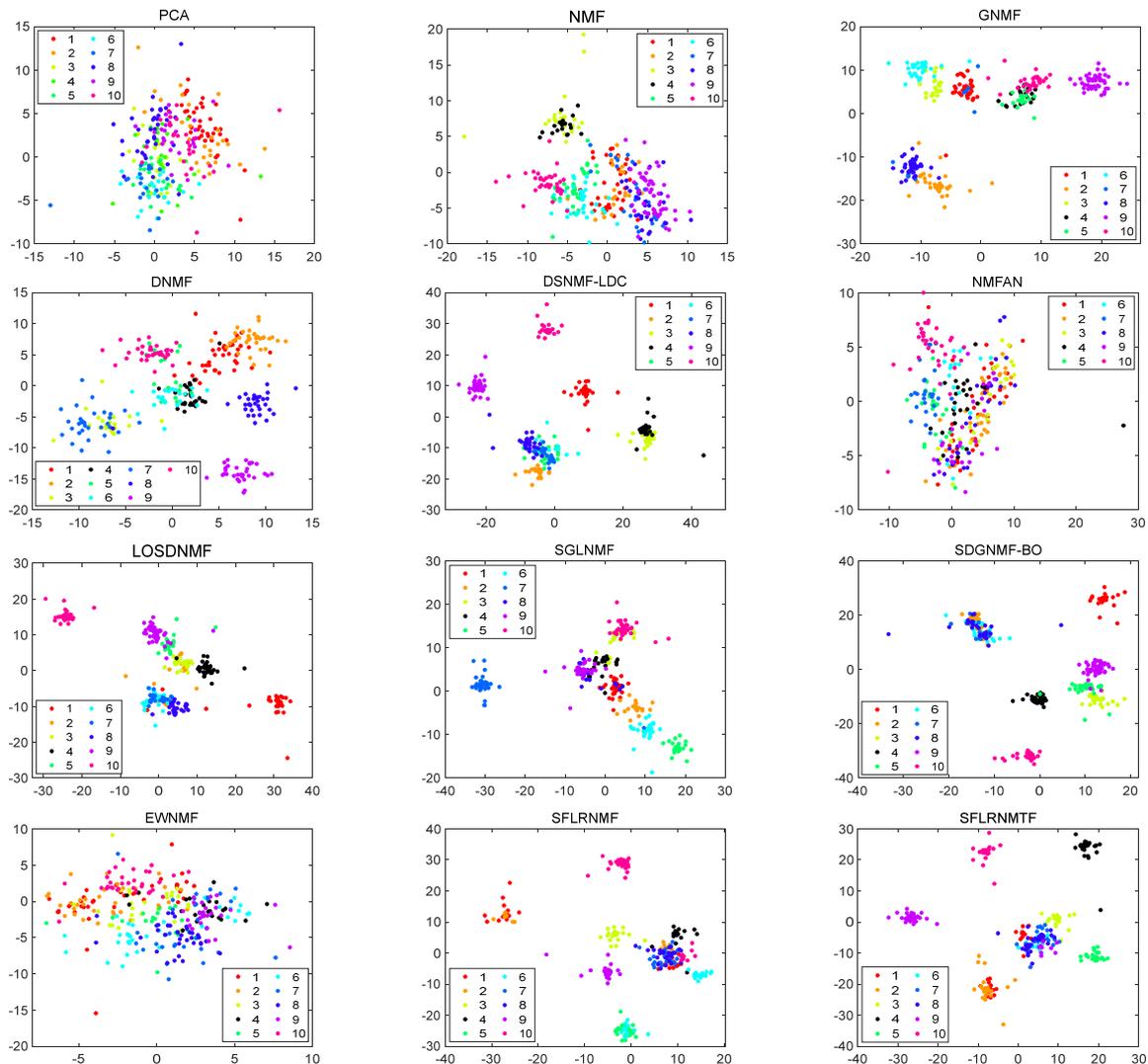


Figure 10. Two-dimensional representations of UMIST dataset using t-SNE on the results of different methods.

Figures 10 and 11 demonstrate that, on the COIL-20 and UMIST datasets, our SFRLNMF and SFRLNMTF algorithms perform excellently, being capable of distinguishing data samples more clearly. Particularly in the visual comparison on the COIL-20 dataset, both methods proposed by us are demonstrated to successfully and clearly separate the samples from other categories. This indicates that our methods can efficiently learn low-dimensional subspace representations. Not only does this experiment validate the reliability of the aforementioned clustering experiments, but it also confirms the contributions of our two methods.

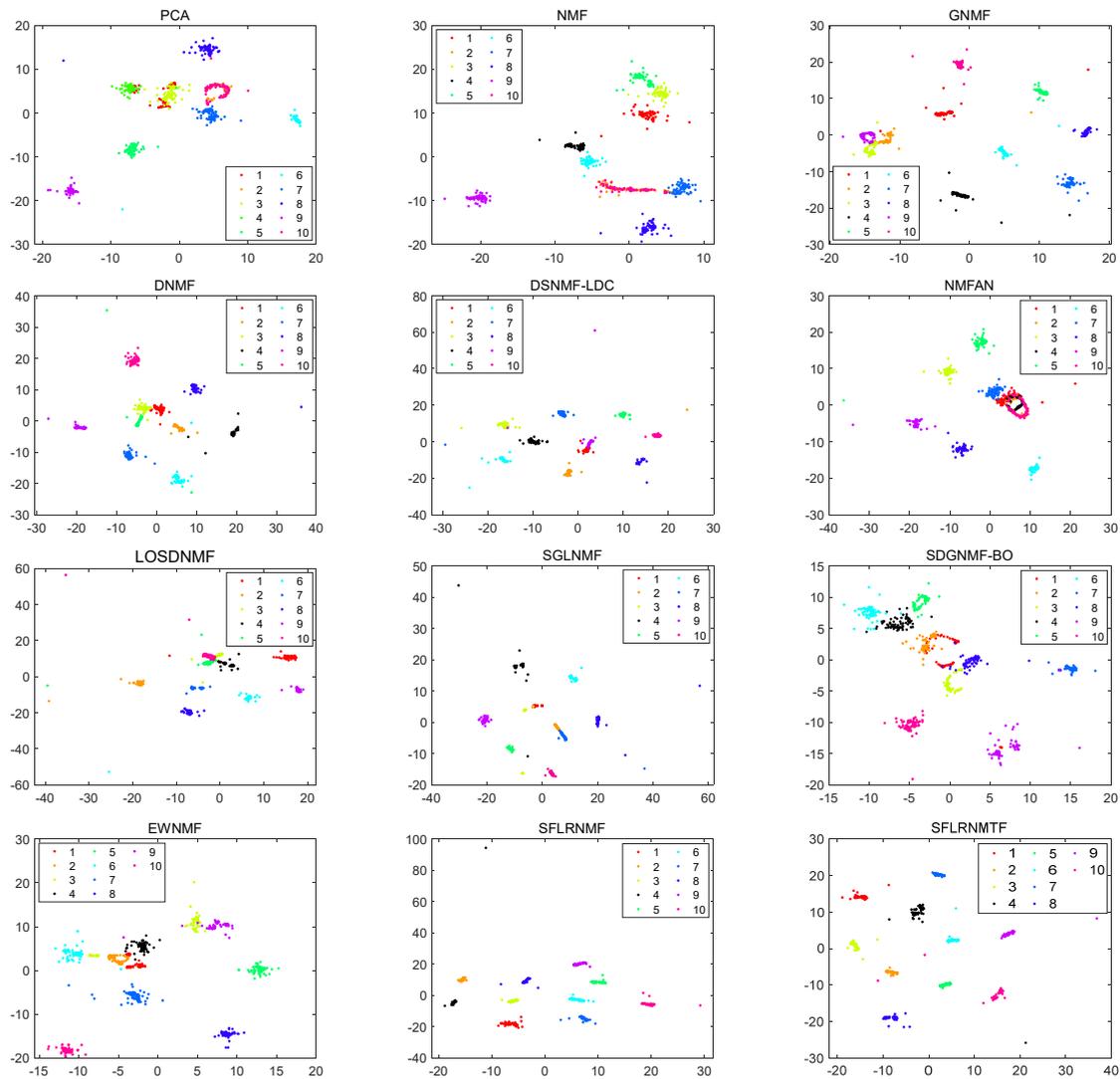
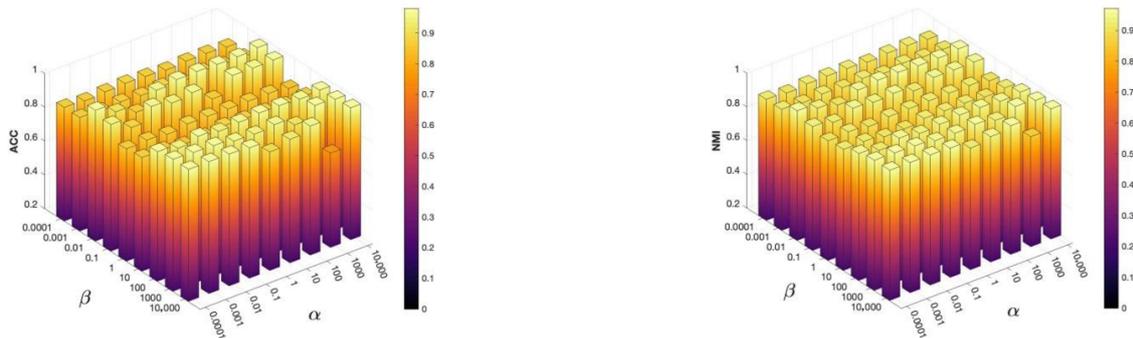


Figure 11. Two-dimensional representations of COIL20 dataset using t-SNE on the results of different methods.

5.5. Parameter Sensitivity Analysis

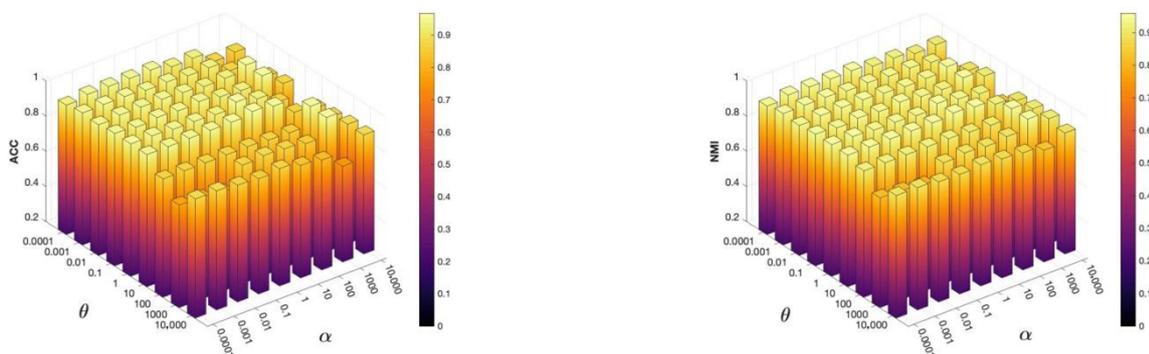
In this section, we explored the sensitivity of parameters for our method on the JAFFE dataset. As seen from the SFRLNMF objective function in Equation (27), three fundamental parameters are involved: the orthogonality constraint parameter β , the sparsity parameter θ , and the graph regularization parameter α . These parameters were selected from the range $[10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3, 10^4]$. The comparison results for ACC and NMI are displayed in three-dimensional bar charts in Figure 12a,b, Figure 13a,b and Figure 14a,b.

From Figure 12a,b, it can be observed that with the variation in the values of parameters α and β , the values of ACC and NMI decrease, indicating that the SFLRNMF is sensitive to the parameters α and β on the JAFFE dataset, particularly evident when θ equals 1000. Specifically, when α is selected from $[10^{-2}, 10^{-1}, 10^2, 10^3, 10^4]$, the scores of these two metrics are slightly higher.



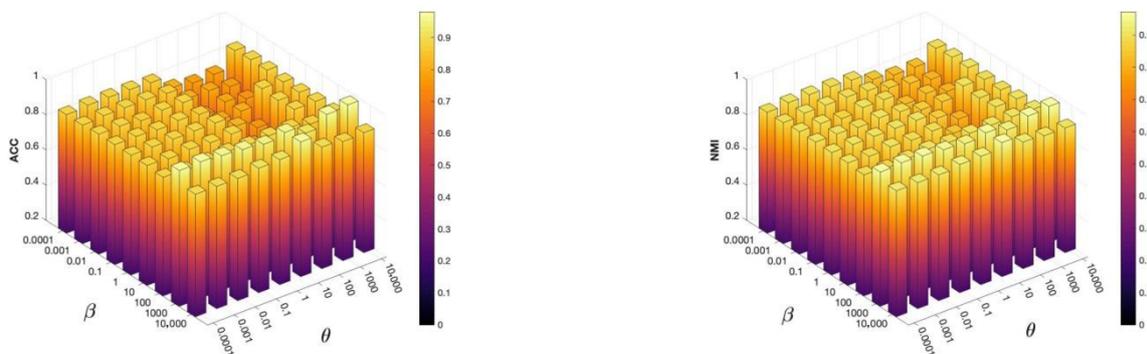
(a) The ACC of SFLRNMF with different α and β on JAFFE (b) The NMI of SFLRNMF with different α and β on JAFFE

Figure 12. The ACC and NMI of SFLRNMF with different α and β on JAFFE.



(a) The ACC of SFLRNMF with different α and θ on JAFFE. (b) The ACC and NMI of SFLRNMF with different α and θ on JAFFE.

Figure 13. The ACC and NMI of SFLRNMF with different α and θ on JAFFE.



(a) The ACC of SFLRNMF with different θ and β on JAFFE. (b) The NMI of SFLRNMF with different θ and β on JAFFE.

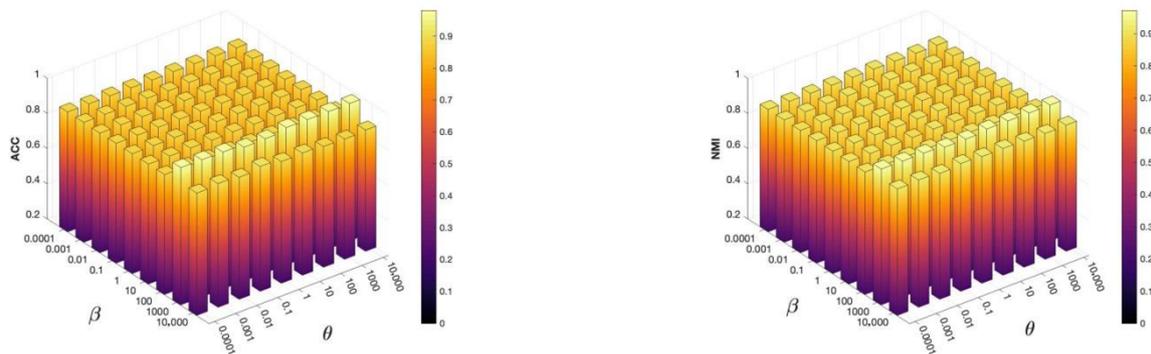
Figure 14. The ACC and NMI of SFLRNMF with different θ and β on JAFFE.

Similarly, we fixed β at 1000 and varied θ and α to compute the scores for ACC and NMI. The corresponding 3D histograms for ACC and NMI are shown in Figure 13a,b. From Figure 13a,b, it is evident that the overall scores for ACC and NMI on the JAFFE dataset are satisfactory, particularly when θ is chosen from the range $[10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1]$, where the scores are highest. Additionally, when θ is within the range $[10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3, 10^4]$, regardless of how the value of α changes within the given range, the scores for ACC and NMI remain high, indicating that the parameter α has a minimal impact on the clustering results. In this scenario, it can be considered that the SFLRNMF algorithm demonstrates greater robustness when β is fixed.

To explore the sensitivity of other parameters in the SFLRNMF algorithm, a similar analysis of parameter sensitivity was conducted with the α parameter fixed, as illustrated in Figure 14.

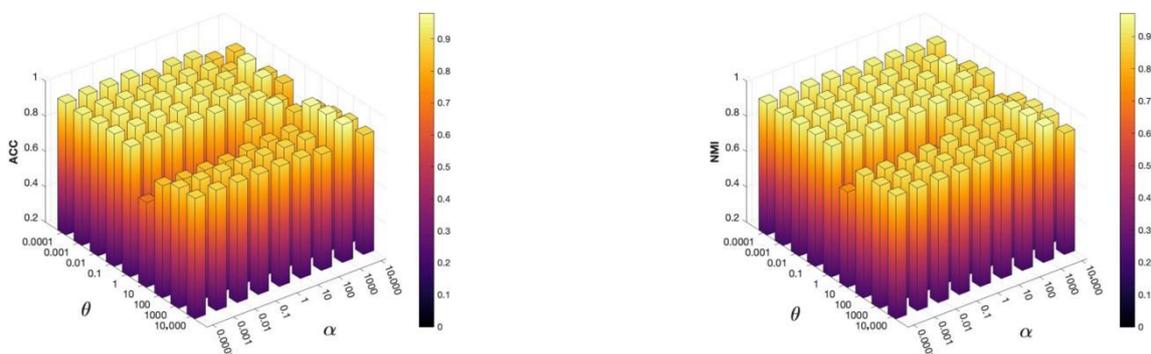
According to Figure 14a,b, overall, the scores of the ACC and NMI metrics vary with changes in these two parameters. Particularly, the best results are obtained when the values of parameters β and θ range from $[10^3, 10^4]$. Therefore, in this case, the SFLRNMF is more sensitive to parameters β and θ .

Next, we also conducted a parameter sensitivity analysis for SFLRNMTF with respect to the parameters α , β , and θ , since SFLRNMTF is an improved version of SFLRNMF and thus they share the same parameters. When α , β , and θ were fixed at 1000, respectively, the corresponding 3D histograms for the ACC and NMI scores on the JAFFE dataset are shown in Figures 15–17, while the ranges for the other two parameters remained set at $10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3, 10^4$.



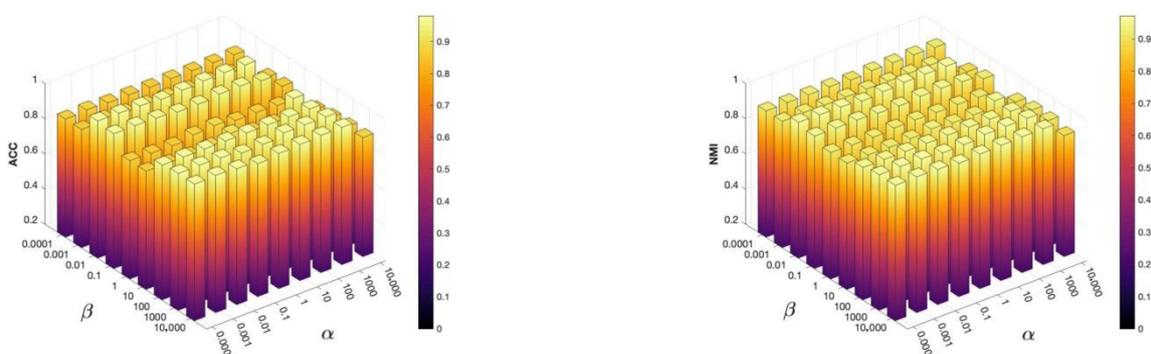
(a) The ACC of SFLRNMTF with different θ and β on JAFFE. (b) The NMI of SFLRNMTF with different θ and β on JAFFE.

Figure 15. The ACC and NMI of SFLRNMTF with different θ and β on JAFFE.



(a) The ACC of SFLRNMTF with different θ and α on JAFFE. (b) The NMI of SFLRNMTF with different θ and α on JAFFE.

Figure 16. The ACC and NMI of SFLRNMTF with different θ and α on JAFFE.



(a) The ACC of SFLRNMTF with different β and α on JAFFE. (b) The NMI of SFLRNMTF with different β and α on JAFFE.

Figure 17. The ACC and NMI of SFLRNMTF with different β and α on JAFFE.

From these figures, it can be concluded that compared to parameter θ , parameter β shows higher sensitivity to the performance of SFLRNMFTF. When parameter α is set to 1000, the appropriate range for β is $[10^3, 10^4]$. When β is 1000 and θ is within the range $[10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0]$, better performance is achieved by the algorithm. Additionally, regardless of the values of θ and β , the overall variation in ACC and NMI remains almost constant, indicating that when parameter α is fixed at 1000, both θ and β exhibit robustness in the performance of SFLRNMFTF. However, in Figures 16 and 17, it can be seen that the variations in ACC and NMI are uneven and unstable, indicating that parameters θ and β exhibit higher sensitivity to the performance of SFLRNMFTF. When parameter α is set to 1000, the clustering performance of the SFLRNMFTF algorithm is optimal and remains stable under other conditions. Additionally, it is evident that under these circumstances, the optimal ranges for θ and β are $[10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0]$ and $[10^{-2}, 10^{-1}]$, respectively.

5.6. Convergence Analysis

Based on the mathematical derivations of the convergence of two algorithms discussed in Sections 3.7 and 4.3 of the paper, it was verified that these two algorithms are highly efficient in addressing the problem of local optima. To further assess the convergence of the SFLRNMF and SFLRNMFTF algorithms, experiments were conducted on the four datasets discussed in this paper. In the experiments, the number of categories was set to the maximum number of categories for each dataset, and the convergence curves of our model on these four datasets were plotted, as shown in Figures 18 and 19. In these figures, the y -axis represents the value of the objective function, while the x -axis represents the number of iterations. The experimental results demonstrated the superior performance of our algorithms on these datasets.

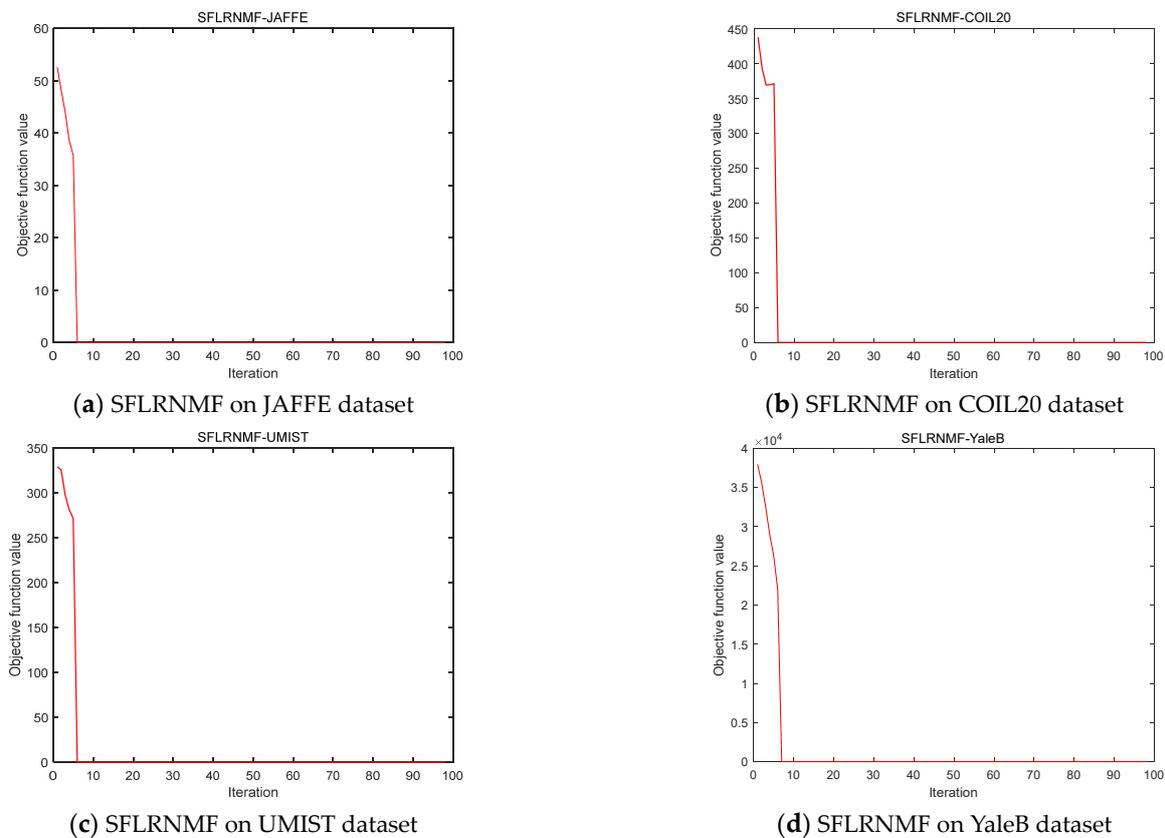


Figure 18. Convergence curves of the SFLRNMF algorithm on four datasets.

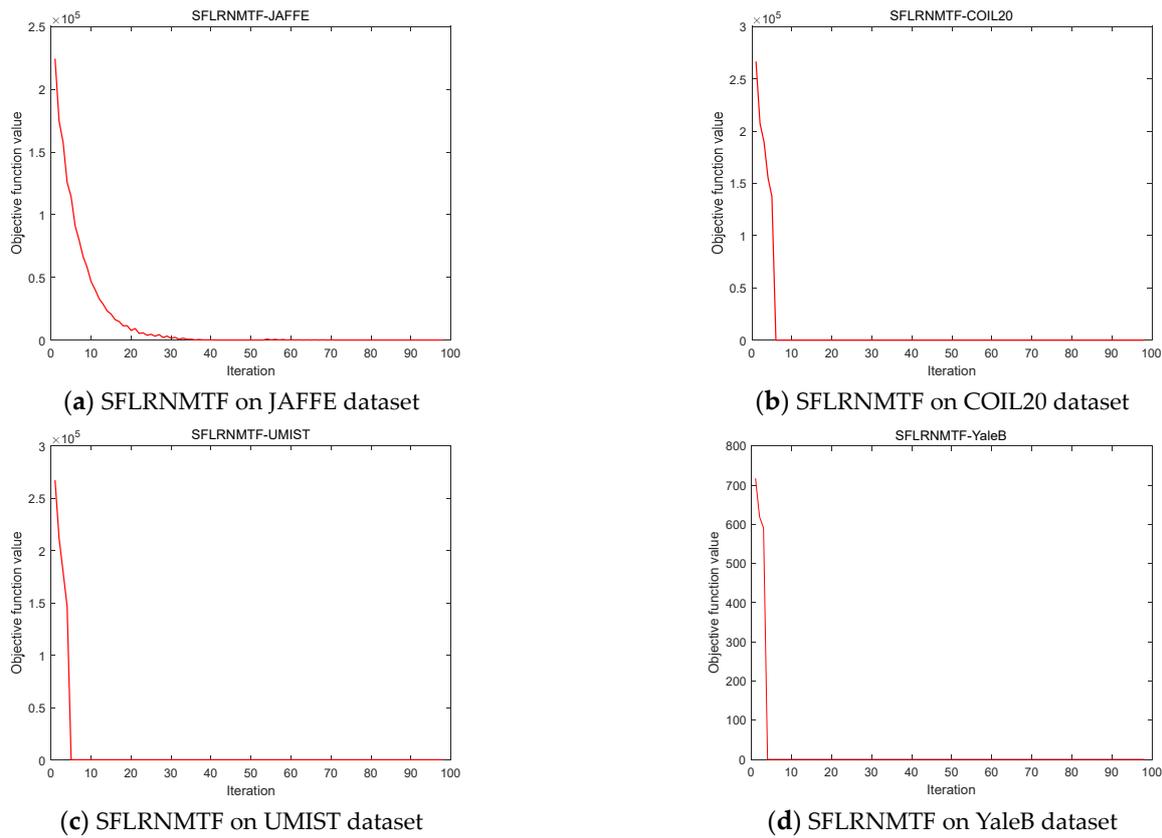


Figure 19. Convergence curves of the SFLRNMTF algorithm on four datasets.

From Figure 18a–d, it can be seen that the SFLRNMF algorithm demonstrates extremely fast convergence on four datasets (JAFFE, COIL20, UMIST, and YaleB). The objective function value drops rapidly and stabilizes within less than 10 iterations. This consistent, fast convergence indicates that SFLRNMF can efficiently find a local optimal solution across various datasets, highlighting the robustness and adaptability of the algorithm. In contrast, Figure 19 shows the convergence curves of the SFLRNMTF algorithm. Observing Figure 19b–d, it can be seen that the SFLRNMTF algorithm also exhibits a relatively fast convergence rate on the COIL20, UMIST, and YaleB datasets, with the objective function value stabilizing within 10 iterations, indicating high optimization efficiency on these datasets. However, in Figure 19a, the convergence rate of SFLRNMTF is relatively slower on the JAFFE dataset. Although the objective function value decreases significantly in the first 10 iterations, it takes approximately 40 iterations to fully converge.

In summary, Figures 18 and 19 indicate that both algorithms exhibit favorable convergence performance across different datasets. Overall, SFLRNMF and SFLRNMTF both converge within a relatively small number of iterations, confirming their stability and efficiency on various datasets. Notably, the SFLRNMF algorithm demonstrates a consistently rapid convergence across all datasets.

5.7. Calculation Time Analysis

In this section, we analyze the computation time by comparing the duration required for clustering experiments conducted on four datasets, in order to more clearly understand the specific impacts of different data dimensions and dataset sizes on computation time. These datasets include JAFFE, COIL20, UMIST, and YaleB. During the experiments, the number of clusters was set to the maximum number of categories each dataset contains. To ensure the stability and reliability of the results, ten independent experiments were conducted on each dataset, and the average execution time of these experiments was

calculated. The results are summarized in Table 11, which details the average computation time for each dataset under various clustering settings.

Table 11. Comparison of computation times of different algorithms on four datasets (s).

Datasets	PCA	NMF	G	LDC	AN	D	LOSD	EW	SGL	SDG	SFLR	SFLRT
JAFFE	0.02	0.11	0.12	0.45	0.92	2.86	1.71	0.84	2.58	3.72	0.87	4.59
COIL20	0.23	0.99	1.05	4.51	23.90	88.71	93.62	6.12	71.73	60.05	3.88	3.61
UMIST	0.08	0.40	0.42	1.12	4.41	16.82	7.61	2.39	13.90	12.84	2.66	3.40
YaleB	0.37	2.45	2.82	12.59	70.89	178.45	305.21	11.66	213.96	187.89	8.79	20.34

From the data analysis in Table 11, it is observed that, firstly, datasets with larger dimensions significantly increase runtime, demonstrating that the model's runtime is notably influenced by the dimensions and size of the data. Secondly, algorithms based on graph structures have longer computation times compared to non-graph-based algorithms. This indicates that graph-based algorithms consume a substantial amount of time in constructing graphs, especially in high-dimensional datasets, where this difference in time is more pronounced. In terms of algorithm types, algorithms utilizing dual graph structures take longer to execute compared to those using a single graph structure. Furthermore, semi-supervised algorithms (such as LOSDNMF, SGLNMF, SDGNMF-BO) have longer runtimes compared to unsupervised algorithms. For unsupervised dual graph algorithms like DNMF, as well as SFLRNMF and SFLRNMTF, the latter two demonstrate superior performance on high-dimensional datasets. This is primarily due to these algorithms having faster convergence rates and fewer iterations, significantly enhancing efficiency when dealing with large-scale or high-dimensional data.

6. Conclusions

In this study, two innovative unsupervised algorithms, SFLRNMF and SFLRNMTF, are proposed, with the latter being a further improvement of the former. The SFLRNMF algorithm effectively addresses the issue of inaccurate low-rank matrix approximation by integrating orthogonality, sparsity, and dual Laplacian rank constraints in non-negative matrix factorization, significantly enhancing the model's capability in low-dimensional representation. Building on this, SFLRNMTF introduces an additional factor R , which is capable of handling scale differences among the input matrices X , U , and V , further improving the accuracy of factor decomposition. Moreover, both models are designed with a learning mechanism to optimize the graph model, enabling the automatic adjustment of the affinity matrix's weights to more precisely capture the intrinsic structure of the data. According to the numerical experiment results, SFLRNMF and SFLRNMTF outperform several existing mainstream methods on multiple standard datasets, demonstrating a clear competitive advantage.

Author Contributions: H.M., software, data curation, and writing—original draft preparation; Z.M., conceptualization, methodology, writing—review and editing, and validation; H.L., visualization and investigation; J.W., supervision and writing—review and editing. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data and code that support the findings of this study are available from the corresponding author [Z. Ma] upon reasonable request.

Conflicts of Interest: The authors declare that they have no competing interests.

References

1. Gu, B.; Sun, X. Structural Minimax Probability Machine. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *1*, 1646–1656. [[CrossRef](#)] [[PubMed](#)]
2. Babaei, M. Discriminative Nonnegative Matrix Factorization for Dimensionality Reduction. *Neurocomputing* **2015**, *173*, 212–223. [[CrossRef](#)]
3. Deborah, H.; Richard, N.; Hardeberg, J.Y. A Comprehensive Evaluation of Spectral Distance Functions and Metrics for Hyperspectral Image Processing. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 3224–3234. [[CrossRef](#)]
4. Bengio, Y.; Courville, A.; Vincent, P. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 1798–1828. [[CrossRef](#)] [[PubMed](#)]
5. Shu, Z.; Wu, X.; Fan, H.; Huang, P.; Wu, D.; Hu, C.; Ye, F. Parameter-Less Auto-Weighted Multiple Graph Regularized Nonnegative Matrix Factorization for Data Representation. *Knowl. Based Syst.* **2017**, *131*, 105–112. [[CrossRef](#)]
6. Wang, Y.-X.; Zhang, Y.-J. Nonnegative Matrix Factorization: A Comprehensive Review. *IEEE Trans. Knowl. Data Eng.* **2013**, *25*, 1336–1353. [[CrossRef](#)]
7. Shu, Z.; Zhao, C.; Huang, P. Local Regularization Concept Factorization and Its Semi-Supervised Extension for Image Representation. *Neurocomputing* **2015**, *158*, 1–12. [[CrossRef](#)]
8. Wold, S.; Esbensen, K.; Geladi, P. Principal Component Analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52. [[CrossRef](#)]
9. Tenenbaum, J.B.; Silva, V.D.; Langford, J.C. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* **2000**, *290*, 2319–2323. [[CrossRef](#)]
10. Roweis, S.T.; Saul, L.K. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* **2000**, *290*, 2323–2326. [[CrossRef](#)]
11. Belkin, M.; Niyogi, P. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Comput.* **2003**, *15*, 1373–1396. [[CrossRef](#)]
12. Belhumeur, P.N.; Hespanha, J.P.; Kriegman, D.J. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 711–720. [[CrossRef](#)]
13. Xu, W.; Gong, Y. Document Clustering by Concept Factorization. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, 25–29 July 2004; pp. 202–209.
14. Wright, J.; Yang, A.Y.; Ganesh, A.; Sastry, S.S.; Yi, M. Robust Face Recognition via Sparse Representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 210–227. [[CrossRef](#)] [[PubMed](#)]
15. Lee, D.D.; Seung, H.S. Learning the Parts of Objects by Non-Negative Matrix Factorization. *Nature* **1999**, *401*, 788–791. [[CrossRef](#)] [[PubMed](#)]
16. Lee, D.D.; Seung, H.S. Algorithms for Non-Negative Matrix Factorization. *Adv. Neural Inf. Process. Syst.* **2000**, *13*, 1969–1976.
17. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)]
18. Liu, G.; Lin, Z.; Yan, S.; Sun, J.; Yu, Y.; Ma, Y. Robust Recovery of Subspace Structures by Low-Rank Representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 171–184. [[CrossRef](#)]
19. Salehani, Y.E.; Arabnejad, E.; Rahiche, A.; Bakhta, A.; Cheriet, M. MSdB-NMF: MultiSpectral Document Image Binarization Framework via Non-Negative Matrix Factorization Approach. *IEEE Trans. Image Process.* **2020**, *29*, 9099–9112. [[CrossRef](#)]
20. Gao, Y.; Church, G. Improving Molecular Cancer Class Discovery through Sparse Non-Negative Matrix Factorization. *Bioinformatics* **2005**, *21*, 3970–3975. [[CrossRef](#)]
21. Ding, C.; Li, T.; Peng, W.; Park, H. Orthogonal Nonnegative Matrix T-Factorizations for Clustering. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, 20–23 August 2006; ACM: Philadelphia, PA, USA, 2006; pp. 126–135.
22. Cai, D.; He, X.; Han, J.; Huang, T.S. Graph Regularized Nonnegative Matrix Factorization for Data Representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1548–1560. [[CrossRef](#)]
23. Shang, F.; Jiao, L.C.; Wang, F. Graph Dual Regularization Non-Negative Matrix Factorization for Co-Clustering. *Pattern Recognit.* **2012**, *45*, 2237–2250. [[CrossRef](#)]
24. Sun, J.; Wang, Z.; Sun, F.; Li, H. Sparse Dual Graph-Regularized NMF for Image Co-Clustering. *Neurocomputing* **2018**, *316*, 156–165. [[CrossRef](#)]
25. Li, S.; Li, W.; Hu, J.; Li, Y. Semi-Supervised Bi-Orthogonal Constraints Dual-Graph Regularized NMF for Subspace Clustering. *Appl. Intell.* **2022**, *52*, 3227–3248. [[CrossRef](#)]
26. Li, H.; Gao, Y.; Liu, J.; Zhang, J.; Li, C. Semi-Supervised Graph Regularized Nonnegative Matrix Factorization with Local Coordinate for Image Representation. *Signal Process. Image Commun.* **2022**, *102*, 116589. [[CrossRef](#)]
27. Wang, J.; Ma, Z.; Li, H.; Feng, D. Semi-Supervised Dual-Graph Regularization Non-Negative Matrix Factorization with Local Coordinate and Orthogonal Constraints for Image Clustering. *J. Electron. Imaging* **2022**, *31*, 053009. [[CrossRef](#)]
28. Xu, K.; Chen, L.; Wang, S. Data-Driven Kernel Subspace Clustering with Local Manifold Preservation. In Proceedings of the 2022 IEEE International Conference on Data Mining Workshops (ICDMW), Orlando, FL, USA, 8 November–1 December 2022; pp. 876–884.
29. Xu, K.; Chen, L.; Wang, S. A Multi-View Kernel Clustering Framework for Categorical Sequences. *Expert Syst. Appl.* **2022**, *197*, 116637. [[CrossRef](#)]

30. Huang, S.; Xu, Z.; Wang, F. Nonnegative Matrix Factorization with Adaptive Neighbors. In Proceedings of the 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017; pp. 486–493.
31. Shu, Z.; Wu, X.-J.; You, C.; Liu, Z.; Li, P.; Fan, H.; Ye, F. Rank-Constrained Nonnegative Matrix Factorization for Data Representation. *Inf. Sci.* **2020**, *528*, 133–146. [[CrossRef](#)]
32. Tang, J.; Wan, Z. Orthogonal Dual Graph-Regularized Nonnegative Matrix Factorization for Co-Clustering. *J. Sci. Comput.* **2021**, *87*, 66. [[CrossRef](#)]
33. Luo, M.; Zhang, K. A Hybrid Approach Combining Extreme Learning Machine and Sparse Representation for Image Classification. *Eng. Appl. Artif. Intell.* **2014**, *27*, 228–235. [[CrossRef](#)]
34. Xu, Z.; Chang, X.; Xu, F.; Zhang, H. $L_{1/2}$ Regularization: A Thresholding Representation Theory and a Fast Solver. *IEEE Trans. Neural Netw. Learn. Syst.* **2012**, *23*, 1013–1027. [[CrossRef](#)]
35. Das, K.C. The Laplacian Spectrum of a Graph. *Comput. Math. Appl.* **2004**, *48*, 715–724. [[CrossRef](#)]
36. Biggs, N. SPECTRAL GRAPH THEORY (CBMS Regional Conference Series in Mathematics 92). *Bull. Lond. Math. Soc.* **1998**, *30*, 197–199. [[CrossRef](#)]
37. Nie, F.; Wang, X.; Jordan, M.; Huang, H. The Constrained Laplacian Rank Algorithm for Graph-Based Clustering. *Proc. AAAI Conf. Artif. Intell.* **2016**, *30*. [[CrossRef](#)]
38. Fan, K. On a Theorem of Weyl Concerning Eigenvalues of Linear Transformations I. *Proc. Natl. Acad. Sci. USA* **1949**, *35*, 652–655. [[CrossRef](#)]
39. Huang, J.; Nie, F.; Huang, H. *A New Simplex Sparse Learning Model to Measure Data Similarity for Clustering*; AAAI Press: Washington, DC, USA, 2015; pp. 3569–3575.
40. Shang, R.; Zhang, Z.; Jiao, L.; Liu, C.; Li, Y. Self-Representation Based Dual-Graph Regularized Feature Selection Clustering. *Neurocomputing* **2016**, *171*, 1242–1253. [[CrossRef](#)]
41. Shi, C.; Ruan, Q.; An, G.; Zhao, R. Hessian Semi-Supervised Sparse Feature Selection Based on $\{L_{2,1/2}\}$ -Matrix Norm. *IEEE Trans. Multimed.* **2015**, *17*, 16–28. [[CrossRef](#)]
42. Meng, Y.; Shang, R.; Jiao, L.; Zhang, W.; Yuan, Y.; Yang, S. Feature Selection Based Dual-Graph Sparse Non-Negative Matrix Factorization for Local Discriminative Clustering. *Neurocomputing* **2018**, *290*, 87–99. [[CrossRef](#)]
43. Wei, J.; Tong, C.; Wu, B.; He, Q.; Qi, S.; Yao, Y.; Teng, Y. An Entropy Weighted Nonnegative Matrix Factorization Algorithm for Feature Representation. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 5381–5391. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.