

Article

Autoencoder-like Sparse Non-Negative Matrix Factorization with Structure Relationship Preservation

Ling Zhong¹ and Haiyan Gao^{1,2,*} 
¹ School of Statistics and Data Science, Lanzhou University of Finance and Economics, Lanzhou 730020, China; zhongling@lzufe.edu.cn

² Key Laboratory of Digital Economy and Social Computing Science of Gansu, Lanzhou 730020, China

* Correspondence: gaohy@lzufe.edu.cn

Abstract

Clustering algorithms based on non-negative matrix factorization (NMF) have garnered significant attention in data mining due to their strong interpretability and computational simplicity. However, traditional NMF often struggles to effectively capture and preserve topological structure information between data during low-dimensional representation. Therefore, this paper proposes an autoencoder-like sparse non-negative matrix factorization with structure relationship preservation (ASNMF-SRP). Firstly, drawing on the principle of autoencoders, a “decoder-encoder” co-optimization matrix factorization framework is constructed to enhance the factorization stability and representation capability of the coefficient matrix. Then, a preference-adjusted random walk strategy is introduced to capture higher-order neighborhood relationships between samples, encoding multi-order topological structure information of the data through an optimal graph regularization term. Simultaneously, to mitigate the impact of noise and outliers, the $l_{2,1}$ -norm is used to constrain the feature correlation between low-dimensional representations and the original data, preserving feature relationships between data, and a sparse constraint is imposed on the coefficient matrix via the inner product. Finally, clustering experiments conducted on 8 public datasets demonstrate that ASNMF-SRP consistently exhibits favorable clustering performance.

Keywords: structure relationship preservation; autoencoder-like; sparse constraint; non-negative matrix factorization; clustering



Academic Editor:
Friedhelm Schwenker

Received: 21 July 2025

Revised: 12 August 2025

Accepted: 17 August 2025

Published: 19 August 2025

Citation: Zhong, L.; Gao, H. Autoencoder-like Sparse Non-Negative Matrix Factorization with Structure Relationship Preservation. *Entropy* **2025**, *27*, 875. <https://doi.org/10.3390/e27080875>

Copyright: © 2025 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

Clustering aims to reveal intrinsic relationships between data through similarity measurement, and it is widely applied in various fields such as marketing [1], gene expression [2], and pattern recognition [3]. However, with the continuous increase in data scale, traditional clustering algorithms face challenges posed by the “curse of dimensionality”, including increased computational complexity, feature redundancy, and amplified noise. Consequently, researchers have proposed a collaborative optimization paradigm of “dimensionality reduction followed by clustering”. Matrix factorization, as a crucial dimensionality reduction method, factorizes the original data matrix into low-rank submatrices to achieve the goal of “parts representing the whole”. Common matrix factorization methods include Principal Component Analysis (PCA) [4], Singular Value Decomposition (SVD) [5], and non-negative matrix factorization (NMF) [6]. Due to advantages such as

non-negativity constraints, computational simplicity, and fast implementation, NMF is frequently used in clustering analysis.

In recent years, to further enhance the performance of NMF, researchers have proposed many improved NMF algorithms. For example, Hoyer et al. [7] proposed sparse non-negative matrix factorization (SNMF), utilizing l_1 -norm to improve the sparsity of NMF. Kong et al. [8] proposed robust non-negative matrix factorization (RNMF), replacing the Frobenius norm with the $l_{2,1}$ -norm to enhance the algorithm's robustness. Ding et al. [9] proposed orthogonal non-negative matrix factorization (ONMF), reducing interference from redundant information by imposing orthogonality constraints on the factorized sub-matrices. These NMF algorithms primarily focus on the reconstruction error after matrix factorization but overlook the spatial structural relationships within the original data. Studies have shown that introducing graph regularization techniques into NMF can not only optimize reconstruction error but also further capture local geometric structures between data [10]. Cai et al. [11] proposed the graph regularized non-negative matrix factorization algorithm (GNMF), encoding topological information between data into the objective function to effectively improve clustering performance. Wu et al. [12] proposed robust manifold non-negative matrix factorization (MNMF21), which considers manifold learning of data and employs the $l_{2,1}$ -norm to mitigate interference from noise and outliers. Li et al. [13] proposed graph regularized non-negative low-rank matrix factorization (GNLMF), enhancing clustering accuracy on image data by capturing low-rank structures. Liu et al. [14] proposed graph regularized discriminative non-negative matrix factorization (GDNMF), which incorporates local geometric structure information between samples and data label information to effectively improve clustering performance on image data. To enhance the quality of graph learning, Huang et al. [15] proposed non-negative matrix factorization with adaptive graph learning (NMFAN), dynamically adjusting graph structure learning to effectively improve algorithm performance. Ren et al. [16] proposed semi-supervised symmetric non-negative matrix factorization with graph quality improvement (S^3 NMFGC), improving clustering performance by dynamically generating and adaptively updating graph learning results. Mohammadi et al. [17] proposed a semi-supervised multi-view clustering method based on adaptive symmetric NMF (SSA-SNMF), which improves the performance of the algorithm by introducing a multi-constraint optimization strategy.

In the process of graph learning, the quality of the learned graph structure directly affects the accuracy and robustness of clustering results. Currently, most graph learning methods are based on measuring Euclidean distances between samples. As the complexity of data internal structures increases, some samples may exhibit distant geometric distances, yet their local neighborhood topologies might show highly similar characteristics. Therefore, with the development of Graph Convolutional Neural Networks (GCNNs) [18], higher-order topological relationships between samples have gained widespread attention among researchers. Wang et al. [19] proposed a robust high-order graph learning algorithm for multi-view clustering (RHGL), which improves clustering accuracy by learning high-order graphs. Zhan et al. [20] proposed a multi-view clustering method with optimal high-order graph embedding (Co-MSE), which enhances the quality of graph learning by modeling high-order correlations. Additionally, Wang et al. [21] indicated that autoencoder-based methods can not only reduce data dimensionality but also further learn latent information between data points. Thus, from the perspective of “decoder-encoder” collaborative optimization, although the aforementioned improved NMF-based algorithms enhance clustering performance through different regularization terms, their objective functions only include the “decoder” component. They lack an explicit “encoder” structure with constraints on the coefficient matrix, thus preventing the coefficient matrix from fully uncovering latent information in the raw data.

Research indicates that introducing sparsity constraints into NMF can effectively enhance the clustering performance of algorithms [7,22]. Meng et al. [23] proposed sparse and orthogonally constrained dual-graph regularized non-negative matrix factorization (SODNMF), which incorporates dual-graph regularization, sparsity constraints, and orthogonality constraints into the objective function, significantly improving the algorithm's performance. Peng et al. [24] proposed logarithmic sparse non-negative matrix factorization (LS-NMF), designing a l_{\log} -(pseudo) norm to achieve sparse constraints. Xiong et al. [25] proposed dual-graph regularized sparse robust and adaptive non-negative matrix factorization (DRGSNMF), adopting the $L_{2,p}$ -norm to construct a sparse regularization term. Although the implementations of these algorithms differ, they all further enhance algorithm performance through sparse constraints.

Features often exhibit complex correlations or complementarity, such as in consumer behavior data. Therefore, in NMF, optimizing only the coefficient matrix may fail to effectively preserve structural relationships between features. To address this, Shang et al. [26] proposed dual-graph regularized non-negative matrix factorization (DNMF), which enhances the algorithm's generalization capability by incorporating learning of the feature graph. Gu et al. [27] proposed a dual-graph regularized co-optimization clustering algorithm (DRCC), simultaneously considering the learning of both sample and feature graphs. Furthermore, to ensure the effective preservation of feature relationships from the original data matrix during dimensionality reduction, Hedjam et al. [28] proposed feature relationship-preserving non-negative matrix factorization (FR-NMF), and Salahian et al. [29] proposed a deep autoencoder non-negative matrix factorization with contrastive regularization and feature relationship preservation (DANMF-CRFR).

Therefore, this paper proposes an autoencoder-like sparse non-negative matrix factorization with structure relationship preservation (ASNMF-SRP). This method incorporates autoencoder principles into NMF, extending the constraints on the coefficient matrix from implicit to explicit. Through these dual constraints, it not only enhances the decomposition stability of the coefficient matrix but also further uncovers latent information in the raw data. Simultaneously, it preserves topological information in the original data space using higher-order graph regularization terms for both samples and features and incorporates an optimization term for feature relationship preservation. Finally, sparse optimization is applied to the coefficient matrix. Thus, ASNMF-SRP integrates autoencoder-like NMF, higher-order graph regularization, feature relationship preservation, and sparse constraints into a unified optimization framework.

The main contributions of the ASNMF-SRP algorithm are as follows:

(1) Optimization of the coefficient matrix. Firstly, this paper introduces an autoencoder-like NMF, adding an explicit “encoder” structure to constrain the coefficient matrix. Then, in the local structure learning process, a higher-order graph regularization method is proposed, enabling the algorithm to progressively extend from first-order to higher-order graph learning. Finally, an inner-product-based sparse constraint is introduced, preserving high-order collaborative relationships between data at the element level.

(2) Optimization of the basis matrix. This paper proposes a robust feature relationship preservation method, thereby establishing a topology-preserving relationship between the low-dimensional embedding space and the original high-dimensional data feature structure.

(3) This paper demonstrates the feasibility of ASNMF-SRP through convergence analysis, time complexity analysis, and other aspects. Simultaneously, in the experimental section, comparative analysis of clustering results, visualization analysis of low-dimensional representation data, parameter sensitivity analysis, and other experiments are conducted.

Ultimately, the experimental results show that ASNMF-SRP exhibits excellent clustering performance.

The remaining work in this paper is organized as follows: Section 2 introduces some foundational work related to ASNMF-SRP. Section 3 primarily describes the derivation of the ASNMF-SRP algorithm, including the convergence and time complexity of submatrix update iterations. Section 4 presents a series of experiments conducted on eight public datasets. Section 5 provides a summary of the ASNMF-SRP algorithm.

2. Related Work

2.1. Non-Negative Matrix Factorization (NMF)

The objective of NMF is to factorize a given non-negative data matrix $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) \in \mathbb{R}_+^{m \times n}$ into the product of two non-negative and low-rank matrices: a basis matrix $\mathbf{U} \in \mathbb{R}_+^{m \times r}$ and a coefficient matrix $\mathbf{V} \in \mathbb{R}_+^{r \times n}$, i.e., $\mathbf{X} \approx \mathbf{UV}$, where $r \ll \min\{m, n\}$. During the factorization process, NMF optimizes the approximate representation by minimizing the reconstruction error. Therefore, it employs the Frobenius norm to define the objective function as follows:

$$\min \|\mathbf{X} - \mathbf{UV}\|_F^2, \quad \text{s.t. } \mathbf{U} \geq 0, \mathbf{V} \geq 0 \quad (1)$$

Here, $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, which measures the reconstruction error after matrix factorization. For the optimization problem in Equation (1), Lee et al. [30] proposed an iterative algorithm based on multiplicative update rules, with the update formulas given by the following:

$$\mathbf{u}_{ij} \leftarrow \mathbf{u}_{ij} \frac{(\mathbf{XV}^T)_{ij}}{(\mathbf{UVV}^T)_{ij}}, \quad \mathbf{v}_{ij} \leftarrow \mathbf{v}_{ij} \frac{(\mathbf{U}^T \mathbf{X})_{ij}}{(\mathbf{U}^T \mathbf{UV})_{ij}} \quad (2)$$

2.2. Robust Non-Negative Matrix Factorization (RNMF)

NMF measures reconstruction error based on Euclidean distance, making it highly susceptible to noise and outliers. To further enhance the robustness of the matrix factorization process, Kong et al. [8] proposed robust non-negative matrix factorization (RNMF), which employs the $l_{2,1}$ -norm to assess the quality of data reconstruction. The objective function of RNMF is as follows:

$$\min \|\mathbf{X} - \mathbf{UV}\|_{2,1}, \quad \text{s.t. } \mathbf{U} \geq 0, \mathbf{V} \geq 0 \quad (3)$$

The update formulas for the basis matrix \mathbf{U} and coefficient matrix \mathbf{V} in Equation (3) are as follows:

$$\mathbf{u}_{ij} \leftarrow \mathbf{u}_{ij} \frac{(\mathbf{XG}_1 \mathbf{V}^T)_{ij}}{(\mathbf{UVG}_1 \mathbf{V}^T)_{ij}}, \quad \mathbf{v}_{ij} \leftarrow \mathbf{v}_{ij} \frac{(\mathbf{U}^T \mathbf{XG}_1)_{ij}}{(\mathbf{U}^T \mathbf{UVG}_1)_{ij}} \quad (4)$$

Here, \mathbf{G}_1 is a diagonal matrix with diagonal elements given by the following:

$$(\mathbf{G}_1)_{jj} = 1 / \|\mathbf{X} - \mathbf{UV}\|_2 \quad (5)$$

2.3. Graph Regularized Non-Negative Matrix Factorization (GNMF)

The standard NMF achieves the goal of “representing the whole by parts” by minimizing the reconstruction error, but this approach overlooks the latent structural information in the data. Therefore, Cai et al. [11] proposed graph regularized non-negative matrix factorization (GNMF). GNMF improves upon NMF by encoding the local geometric structure among samples into the objective function for optimization, thereby enhancing the

representation quality of samples in the low-dimensional space. The objective function of GNMF is as follows:

$$\min \|X - UV\|_F^2 + \lambda \text{Tr}(VLV^T), \quad \text{s.t. } U \geq 0, V \geq 0 \quad (6)$$

Here, λ is a non-negative regularization parameter that adjusts the strength of graph learning. L is the Laplacian matrix, which satisfies $L = D - W$. In this context, W is the similarity matrix reflecting the relationships between samples, and the degree matrix $D = \text{diag}(D_{11}, D_{22}, \dots, D_{nn})$ is a diagonal matrix formed by the row sums of W , i.e., $D_{ii} = \sum_j W_{ij}$. The update rules for GNMF are as follows:

$$U_{ij} \leftarrow U_{ij} \frac{(XV^T)_{ij}}{(UVV^T)_{ij}}, \quad V_{ij} \leftarrow V_{ij} \frac{(U^T X + \lambda WV)_{ij}}{(U^T UV + \lambda DV)_{ij}} \quad (7)$$

3. Methodology

3.1. Autoencoder-like Non-Negative Matrix Factorization

In clustering analysis, the coefficient matrix V of NMF serves as the clustering indicator matrix. Therefore, the coefficient matrix V needs to effectively encode the structural features of the original data to ensure the accuracy of subsequent clustering. Inspired by the non-negative symmetric encoder-decoder method proposed by Sun et al. [31] in community detection and the principle of autoencoders, this paper imposes an explicit constraint on the coefficient matrix V , i.e., $V \approx f(X) = U^T X$. This approach not only improves the factorization stability of the coefficient matrix but also further uncovers the latent representations of the original data. The specific implementation process is as follows:

Linear Decoder:

$$g(v) = Uv \rightarrow X \approx g(V) = UV \rightarrow \|X - UV\|_F^2 \quad (8)$$

Linear Encoder:

$$f(x) = U^T x \rightarrow V \approx f(X) = U^T X \rightarrow \|V - U^T X\|_F^2 \quad (9)$$

In the “decoder-encoder” architecture, the coefficient matrix V acts as the encoder matrix. From Equations (8) and (9), it can be seen that the “decoder” part corresponds to standard NMF, aiming to minimize the reconstruction error between data, i.e., maximizing the approximation $X \approx UV$. The “encoder” part, on the other hand, transforms the data matrix X into a distributed representation using the basis matrix $U \in \mathbb{R}_+^{m \times r}$, thereby imposing an explicit constraint on the coefficient matrix, i.e., $V \approx U^T X$.

By combining Equations (8) and (9), an autoencoder-like non-negative matrix factorization framework is obtained, with the following objective function:

$$\min \underbrace{\|X - UV\|_F^2}_{\text{decoder}} + \underbrace{\|V - U^T X\|_F^2}_{\text{encoder}}, \quad \text{s.t. } U \geq 0, V \geq 0 \quad (10)$$

Here, the decoder and encoder parts share the same basis matrix U , thereby forming a symmetry constraint. Figure 1 is a schematic diagram of Equation (10).

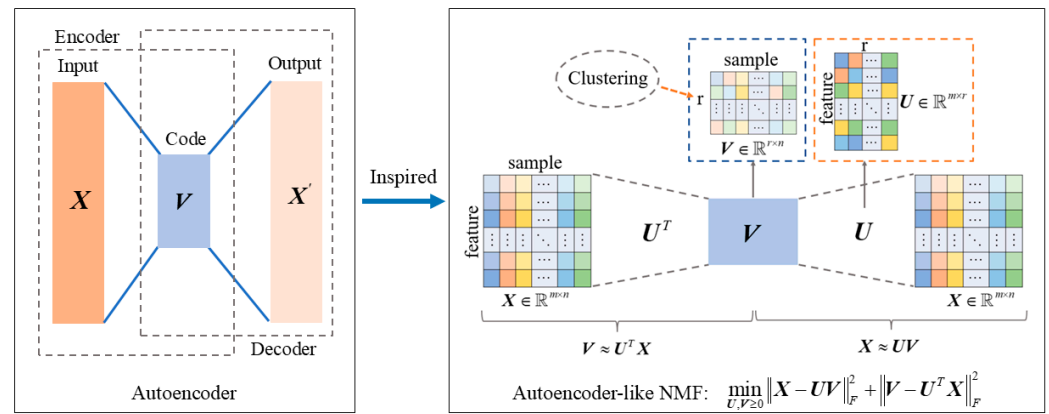


Figure 1. Schematic diagram of autoencoder-like non-negative matrix factorization.

3.2. High-Order Graph Regularization

To further improve the quality of the data representation in the low-dimensional space, this paper introduces graph regularization techniques based on Equation (10) to learn the geometric structural information of the original high-dimensional space. Therefore, this paper captures the neighborhood relationships between samples by constructing a nearest neighbor graph. The first-order similarity matrix W_1 is defined as follows:

$$(W_1)_{ij} = \begin{cases} 1, & x_i \in N_k(x_j) \text{ or } x_j \in N_k(x_i) \\ 0, & \text{otherwise} \end{cases} \quad (11)$$

Here, $N_k(x_i)$ denotes the set of k -nearest data points to sample x_i . x_i and x_j are the i -th and j -th column samples of the original data matrix $X = (x_1, x_2, \dots, x_n) \in \mathbb{R}_+^{m \times n}$, respectively.

The coefficient matrix $V = (v_1, v_2, \dots, v_n) \in \mathbb{R}_+^{r \times n}$ is the low-dimensional representation to be optimized. According to manifold learning and spectral graph theory, the smoothness of this low-dimensional representation is given by the following:

$$\begin{aligned} \mathcal{Z} &= \frac{1}{2} \sum_{i,j=1}^n \|v_i - v_j\|^2 (W_1)_{ij} \\ &= \sum_{i=1}^n v_i v_i^T (D_1)_{ii} - \sum_{i,j=1}^n v_i v_j^T (W_1)_{ij} \\ &= \text{Tr}(VD_1V^T) - \text{Tr}(VW_1V^T) \\ &= \text{Tr}(VL_1V^T) \end{aligned} \quad (12)$$

Here, D_1 is a diagonal matrix with $(D_1)_{ii} = \sum_j W_{ij}$. L_1 is defined as the first-order Laplacian matrix, which satisfies the equation $L_1 = D_1 - W_1$.

In real-world scenarios, samples often exhibit complex structural relationships. For instance, in online shopping, consumers from different regions may have no direct connection. However, by analyzing behavioral data, it is possible to identify neighborhood relationships where consumers share highly similar purchasing patterns. Such neighborhood relationships provide a scientific basis for businesses to segment target customer groups and implement precision marketing strategies. As shown in Figure 2, samples x_i and x_l are both connected to samples 1, 2, and 3, indicating that x_i and x_l share the same neighborhood structure. However, x_i and x_l are far apart and not within each other's neighborhood sets. From Equation (11), it follows that $W_{il} = 0$. Therefore, the first-order similarity matrix can only capture the neighborhood relationships between samples but fails to consider the similarity of their neighborhood structures. To further capture higher-order similarities between different neighborhood systems, this paper introduces a preference-adjusted random

walk strategy. The random walk strategy [32,33] constructs sequences of sample points and encodes multi-step reachability into a higher-order similarity matrix, thereby reflecting higher-order structural information among samples. During the random walk process, the transition probability matrix P_{ij} represents the probability of transitioning from sample x_i to sample x_j , defined as follows:

$$P_{ij} = \frac{a_{ij}}{k_i} \quad (13)$$

Here, a_{ij} indicates the connection state between samples x_i and x_j : if x_i and x_j are connected, then $a_{ij} = 1$; otherwise, $a_{ij} = 0$. k_i represents the degree of sample x_i . Since a_{ij} takes values of 0 or 1, k_i here denotes the number of samples connected to x_i .

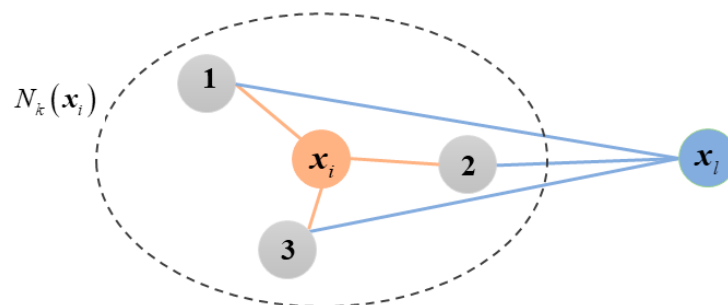


Figure 2. Example of associations between samples.

The transition probability matrix in Equation (13) is based on an equal-probability assumption, which often fails to accurately reflect the topological heterogeneity among samples when processing high-dimensional complex data. Therefore, this paper incorporates the Preferential Attachment (PA) model into Equation (13). By assigning connection preference weights between sample points, the transition probability matrix is further optimized to more realistically reflect higher-order neighborhood relationships among samples. Thus, the new expression is as follows:

$$P_{ij} = \frac{\text{PA}_{\text{simScore}}}{\text{Max}(\text{PA}_{\text{simScore}})} \times \frac{a_{ij}}{k_i} \quad (14)$$

Therefore, the d -th order similarity between samples x_i and x_j is as follows:

$$S_{ij}^d = P_{ik} \times P_{kl} \times \cdots \times P_{zj} \quad (15)$$

where $d = 1, 2, \dots, n$. In practical applications, the value of d should not be too large, as an excessively large d may introduce noise interference, thereby affecting the quality of graph learning. The average of the d -th order adjacency matrices is as follows:

$$W_2 = (S^1 + S^2 + \cdots + S^d) / d \quad (16)$$

In Equation (16), W_2 is referred to as the second-order similarity matrix, which aims to capture higher-order neighborhood relationships between samples. The second-order Laplacian matrix is defined as $L_2 = D_2 - W_2$, where D_2 is a diagonal matrix with entries $(D_2)_{ii} = \sum_j W_{ij}$.

By considering both the first-order and second-order Laplacian matrices, the optimal Laplacian matrix is now defined as follows:

$$L = \omega_1 L_1 + \omega_2 L_2 \quad (17)$$

where w_1 and w_2 are balancing parameters. Consequently, the optimal graph regularization term is given by the following:

$$\min_{V \geq 0} \text{Tr}(VLV^T) \quad (18)$$

3.3. Feature Relationship Preservation

As data structures increase in complexity, feature information also plays a crucial role in clustering analysis. For example, when enterprises identify target customer groups, consumer characteristics, such as purchase frequency (x_1^*), interests (x_2^*), spending power (x_3^*), shopping preferences (x_4^*), and brand loyalty (x_5^*), are key dimensions. Therefore, when high-dimensional data are transformed into low-dimensional representations, preserving the intrinsic relationships between features becomes essential for effectively uncovering the latent structure of customer groups. This is illustrated in Figure 3.

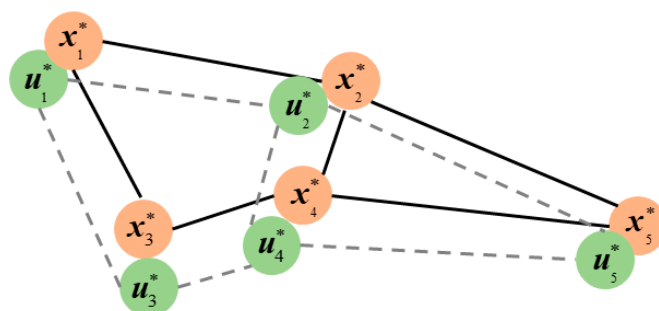


Figure 3. Example of feature relationship preservation in high-dimensional and low-dimensional spaces.

In Figure 3, x_i^* represents the i -th row vector of matrix X , corresponding to the i -th feature information in the original high-dimensional data, where $i = 1, 2, 3, 4, 5$. u_i^* represents the i -th row vector of the basis matrix U . As shown in Figure 3, the feature relationship between x_i^* and u_i^* remains consistent, which provides a theoretical basis for enterprises to identify target customer groups. Therefore, incorporating the feature relationships between data into the objective function for optimization can further preserve the feature relationships among the data.

$(XX^T)_{ij} = \sum_{l=1}^n x_{il}x_{jl}$ represents the inner product between the i -th and j -th features of X . $(UU^T)_{ij} = \sum_{l=1}^n u_{il}u_{jl}$ represents the inner product between the i -th and j -th rows of the basis matrix U . To maintain consistency between feature relationships in the low-dimensional space and original data, we construct the feature preservation constraint term:

$$\min_{U \geq 0} \|XX^T - \beta UU^T\|_F^2 \quad (19)$$

where β is a non-negative balance parameter.

Equation (19) defines the reconstruction error based on the Frobenius norm. Therefore, when seeking to minimize the objective function value, Equation (19) is susceptible to interference from noise and outliers. To address this, this paper employs the $l_{2,1}$ -norm to further optimize Equation (19), thereby enhancing its robustness. The specific expression is as follows:

$$\min_{U \geq 0} \|XX^T - \beta UU^T\|_{2,1} \quad (20)$$

3.4. Sparsity of Coefficient Matrix

In the clustering process, the coefficient matrix V plays the role of a cluster indicator matrix. Therefore, the state of the coefficient matrix V affects the accuracy of clustering

results. Imposing sparse constraints on the coefficient matrix V can make the representation of each sample in the low-dimensional space significantly correlated with only a few basis vectors, thereby weakening the interference of irrelevant information on clustering results. Common sparsity methods include: l_0 -norm, $l_{1/2}$ -norm, l_1 -norm, $l_{2,1}$ -norm, l_{\log} -(pseudo) norm, etc. A detailed introduction is shown in Table 1, where v_{ij} represents the element in the i -th row and j -th column of the coefficient matrix $V \in \mathbb{R}_+^{r \times n}$.

Table 1. Introduction to five common sparsity methods.

No.	Method	Introduction to Sparsity Method
1	l_0 -norm [34]	$\ V\ _0$ denotes the number of non-zero elements in V .
2	$l_{1/2}$ -norm [35]	$\ V\ _{1/2} = \sum_{i=1}^r \sum_{j=1}^n v_{ij} ^{1/2}$
3	l_1 -norm [22]	$\ V\ _1 = \sum_{i=1}^r \sum_{j=1}^n v_{ij} $
4	$l_{2,1}$ -norm [36]	$\ V\ _{2,1} = \sum_{j=1}^n \sqrt{\sum_{i=1}^r v_{ij}^2}$
5	l_{\log} -(pseudo) norm [24]	$\ V\ _{\log} = \sum_{i=1}^r \sum_{j=1}^n \log(1 + v_{ij})$

The five common sparsity methods in Table 1 primarily impose element-wise regularization on the coefficient matrix V but overlook the correlations within the internal structure of V . This element-level separate optimization struggles to capture potential high-order collaborative relationships in the sample space. Therefore, this paper proposes a sparse constraint based on inner product penalty.

For the coefficient matrix $V = (v_1, v_2, \dots, v_n) \in \mathbb{R}_+^{r \times n}$, we have

$$V^T V = \begin{bmatrix} \langle v_1, v_1 \rangle & \langle v_1, v_2 \rangle & \cdots & \langle v_1, v_n \rangle \\ \langle v_2, v_1 \rangle & \langle v_2, v_2 \rangle & \cdots & \langle v_2, v_n \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle v_n, v_1 \rangle & \langle v_n, v_2 \rangle & \cdots & \langle v_n, v_n \rangle \end{bmatrix} \quad (21)$$

In Equation (21), the diagonal elements of $V^T V$ reflect autocorrelation, while the off-diagonal elements reflect correlations between different vectors. Here, to avoid interference from autocorrelation, we remove the diagonal elements based on Equation (21) and only retain the off-diagonal elements for sparsification. Thus, we can further obtain the following:

$$\sum_{i=1}^n \sum_{j=1}^n \langle v_i, v_j \rangle - \sum_{i=1}^n \langle v_i, v_i \rangle = \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n \langle v_i, v_j \rangle = \text{Tr}(V^T V \mathbf{1}_n) - \text{Tr}(V^T V) \quad (22)$$

where $\mathbf{1}_n$ denotes an $n \times n$ matrix with all-one element.

Therefore, the sparse constraint term for the coefficient matrix V is as follows:

$$\min_{V \geq 0} \left(\text{Tr}(V^T V \mathbf{1}_n) - \text{Tr}(V^T V) \right) \quad (23)$$

3.5. Objective Function

From Equations (10), (18), (20) and (23), the objective function of ASNMF-SRP is derived as follows:

$$\min_{\mathbf{U}, \mathbf{V} \geq 0} \|\mathbf{X} - \mathbf{UV}\|_F^2 + \|\mathbf{V} - \mathbf{U}^T \mathbf{X}\|_F^2 + \alpha \text{Tr}(\mathbf{V} \mathbf{L} \mathbf{V}^T) + \|\mathbf{X} \mathbf{X}^T - \beta \mathbf{U} \mathbf{U}^T\|_{2,1} + \lambda (\text{Tr}(\mathbf{V}^T \mathbf{V} \mathbf{1}_n) - \text{Tr}(\mathbf{V}^T \mathbf{V})) \quad (24)$$

where α , β , and λ are all non-negative regularization parameters. Figure 4 is a schematic diagram of the ASNMF-SRP algorithm.

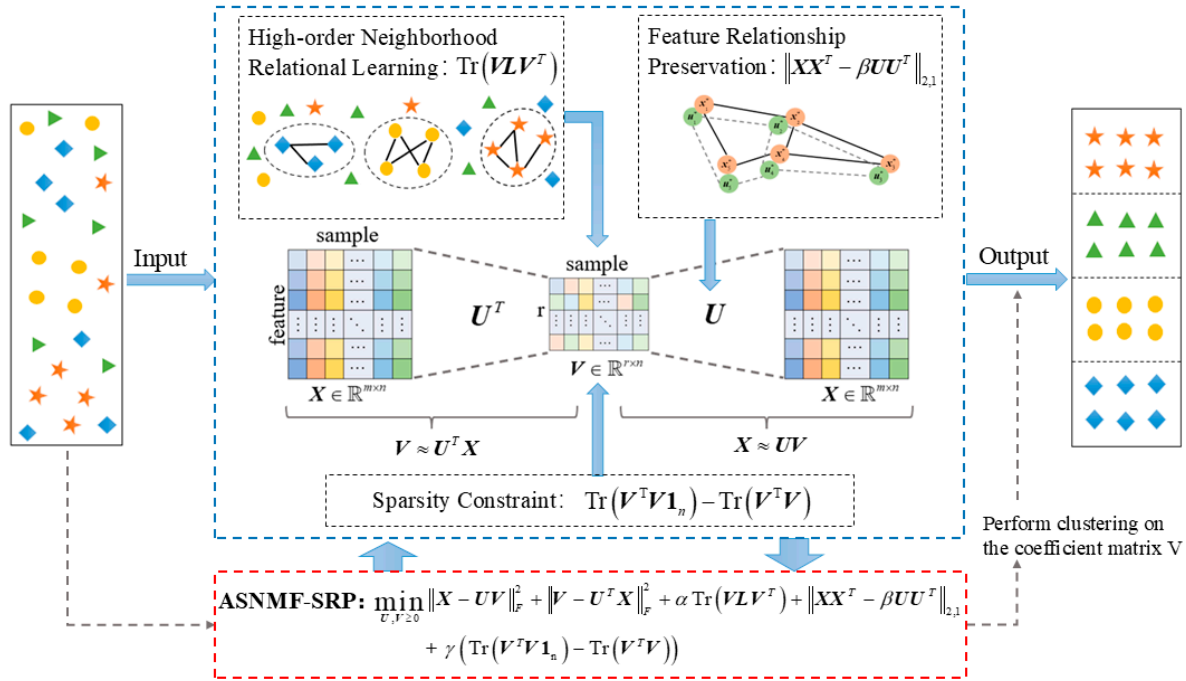


Figure 4. Schematic diagram of the ASNMF-SRP algorithm.

In the objective function Equation (24), each term plays an important role. The first term ($\|\mathbf{X} - \mathbf{UV}\|_F^2$) aims to focus on the reconstruction error of matrix factorization. The second term ($\|\mathbf{V} - \mathbf{U}^T \mathbf{X}\|_F^2$) represents an explicit constraint on the coefficient matrix, thereby improving the factorization stability of the coefficient matrix. The third term ($\text{Tr}(\mathbf{V} \mathbf{L} \mathbf{V}^T)$) encodes higher-order topological information between samples by constructing the optimal Laplacian matrix. The fourth term ($\|\mathbf{X} \mathbf{X}^T - \beta \mathbf{U} \mathbf{U}^T\|_{2,1}$) ensures that the feature relationship structure of the original high-dimensional data can be preserved in the low-dimensional space after dimensionality reduction. The fifth term ($\text{Tr}(\mathbf{V}^T \mathbf{V} \mathbf{1}_n) - \text{Tr}(\mathbf{V}^T \mathbf{V})$) imposes a structured sparse constraint on the coefficient matrix by introducing a sparse regularization term in the form of inner product.

3.6. Optimization Algorithm

ASNMF-SRP employs multiplicative update rules to iteratively optimize the basis matrix \mathbf{U} and coefficient matrix \mathbf{V} . The augmented Lagrangian function of the objective function Equation (24) is as follows:

$$\mathcal{P}(\mathbf{U}, \mathbf{V}) = \|\mathbf{X} - \mathbf{UV}\|_F^2 + \|\mathbf{V} - \mathbf{U}^T \mathbf{X}\|_F^2 + \alpha \text{Tr}(\mathbf{V} \mathbf{L} \mathbf{V}^T) + \|\mathbf{X} \mathbf{X}^T - \beta \mathbf{U} \mathbf{U}^T\|_{2,1} + \lambda (\text{Tr}(\mathbf{V}^T \mathbf{V} \mathbf{1}_n) - \text{Tr}(\mathbf{V}^T \mathbf{V})) - \text{Tr}(\mathbf{\Lambda}_1 \mathbf{U}^T) - \text{Tr}(\mathbf{\Lambda}_2 \mathbf{V}^T) \quad (25)$$

where $\mathbf{\Lambda}_1 \in \mathbb{R}_+^{m \times r}$ and $\mathbf{\Lambda}_2 \in \mathbb{R}_+^{r \times n}$ denote the Lagrange multipliers for \mathbf{U} and \mathbf{V} , respectively.

Expanding Equation (25) and omitting terms independent of \mathbf{U} and \mathbf{V} , we obtain the following:

$$\begin{aligned}\mathcal{L}(\mathbf{U}, \mathbf{V}) = & \text{Tr}(\mathbf{V}^T \mathbf{U}^T \mathbf{U} \mathbf{V} - 2\mathbf{V}^T \mathbf{U}^T \mathbf{X}) + \text{Tr}(\mathbf{V}^T \mathbf{V} - 2\mathbf{V}^T \mathbf{U}^T \mathbf{X} + \mathbf{X}^T \mathbf{U} \mathbf{U}^T \mathbf{X}) \\ & + \alpha \text{Tr}(\mathbf{V} \mathbf{L} \mathbf{V}^T) + \text{Tr}(\beta^2 \mathbf{U} \mathbf{U}^T \mathbf{G}_2 \mathbf{U} \mathbf{U}^T - 2\beta \mathbf{U} \mathbf{U}^T \mathbf{G}_2 \mathbf{X} \mathbf{X}^T) \\ & + \lambda (\text{Tr}(\mathbf{V}^T \mathbf{V} \mathbf{1}_n) - \text{Tr}(\mathbf{V}^T \mathbf{V})) - \text{Tr}(\Lambda_1 \mathbf{U}^T) - \text{Tr}(\Lambda_2 \mathbf{V}^T)\end{aligned}\quad (26)$$

where \mathbf{G}_2 is a diagonal matrix whose diagonal elements are as follows:

$$(\mathbf{G}_2)_{jj} = 1 / \left\| \left(\mathbf{X} \mathbf{X}^T - \beta \mathbf{U} \mathbf{U}^T \right)_i \right\|_2 \quad (27)$$

Taking partial derivatives of $\mathcal{L}(\mathbf{U}, \mathbf{V})$ with respect to \mathbf{U} and \mathbf{V} , respectively, gives the following:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \mathbf{U}} = & 2\mathbf{U} \mathbf{V} \mathbf{V}^T - 4\mathbf{X} \mathbf{V}^T + 2\mathbf{X} \mathbf{X}^T \mathbf{U} + 2\beta^2 \mathbf{U} \mathbf{U}^T \mathbf{G}_2 \mathbf{U} + 2\beta^2 \mathbf{G}_2 \mathbf{U} \mathbf{U}^T \mathbf{U} \\ & - 2\beta \mathbf{X} \mathbf{X}^T \mathbf{G}_2 \mathbf{U} - 2\beta \mathbf{G}_2 \mathbf{X} \mathbf{X}^T \mathbf{U} - \Lambda_1\end{aligned}\quad (28)$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{V}} = 2\mathbf{U}^T \mathbf{U} \mathbf{V} - 4\mathbf{U}^T \mathbf{X} + 2\mathbf{V} + 2\alpha \mathbf{V} \mathbf{L} + 2\lambda \mathbf{V} \mathbf{1}_n - 2\lambda \mathbf{V} - \Lambda_2 \quad (29)$$

To ensure the non-negativity of the coefficient matrix \mathbf{V} , the optimal Laplacian matrix \mathbf{L} in Equation (29) is factorized as $\mathbf{L} = \mathbf{L}^+ - \mathbf{L}^-$, where $\mathbf{L}^+ = (|\mathbf{L}| + \mathbf{L})/2$ and $\mathbf{L}^- = (|\mathbf{L}| - \mathbf{L})/2$.

Equation (29) can be rewritten as follows:

$$\frac{\partial \mathcal{L}}{\partial \mathbf{V}} = 2\mathbf{U}^T \mathbf{U} \mathbf{V} - 4\mathbf{U}^T \mathbf{X} + 2\mathbf{V} + 2\alpha \mathbf{V} \mathbf{L}^+ - 2\alpha \mathbf{V} \mathbf{L}^- + 2\lambda \mathbf{V} \mathbf{1}_n - 2\lambda \mathbf{V} - \Lambda_2 \quad (30)$$

According to the KKT conditions $(\Lambda_1)_{ij} \mathbf{U}_{ij} = 0$ and $(\Lambda_2)_{ij} \mathbf{V}_{ij} = 0$, we derive the following:

$$\begin{aligned}(2\mathbf{U} \mathbf{V} \mathbf{V}^T - 4\mathbf{X} \mathbf{V}^T + 2\mathbf{X} \mathbf{X}^T \mathbf{U} + 2\beta^2 \mathbf{U} \mathbf{U}^T \mathbf{G}_2 \mathbf{U} + 2\beta^2 \mathbf{G}_2 \mathbf{U} \mathbf{U}^T \mathbf{U} \\ - 2\beta \mathbf{X} \mathbf{X}^T \mathbf{G}_2 \mathbf{U} - 2\beta \mathbf{G}_2 \mathbf{X} \mathbf{X}^T \mathbf{U} - \Lambda_1)_{ij} \mathbf{U}_{ij} = 0\end{aligned}\quad (31)$$

$$\left(2\mathbf{U}^T \mathbf{U} \mathbf{V} - 4\mathbf{U}^T \mathbf{X} + 2\mathbf{V} + 2\alpha \mathbf{V} \mathbf{L}^+ - 2\alpha \mathbf{V} \mathbf{L}^- + 2\lambda \mathbf{V} \mathbf{1}_n - 2\lambda \mathbf{V} - \Lambda_2 \right)_{ij} \mathbf{V}_{ij} = 0 \quad (32)$$

Therefore, we obtain the following:

$$\begin{aligned}(2\mathbf{U} \mathbf{V} \mathbf{V}^T - 4\mathbf{X} \mathbf{V}^T + 2\mathbf{X} \mathbf{X}^T \mathbf{U} + 2\beta^2 \mathbf{U} \mathbf{U}^T \mathbf{G}_2 \mathbf{U} + 2\beta^2 \mathbf{G}_2 \mathbf{U} \mathbf{U}^T \mathbf{U} \\ - 2\beta \mathbf{X} \mathbf{X}^T \mathbf{G}_2 \mathbf{U} - 2\beta \mathbf{G}_2 \mathbf{X} \mathbf{X}^T \mathbf{U} - \Lambda_1)_{ij} \mathbf{U}_{ij}^2 = 0\end{aligned}\quad (33)$$

$$\left(2\mathbf{U}^T \mathbf{U} \mathbf{V} - 4\mathbf{U}^T \mathbf{X} + 2\mathbf{V} + 2\alpha \mathbf{V} \mathbf{L}^+ - 2\alpha \mathbf{V} \mathbf{L}^- + 2\lambda \mathbf{V} \mathbf{1}_n - 2\lambda \mathbf{V} - \Lambda_2 \right)_{ij} \mathbf{V}_{ij}^2 = 0 \quad (34)$$

From Equations (33) and (34), the update formulas for \mathbf{U} and \mathbf{V} are as follows:

$$\mathbf{U}_{ij}^{t+1} \leftarrow \mathbf{U}_{ij}^t \sqrt{\frac{(2\mathbf{X} \mathbf{V}^T + \beta \mathbf{X} \mathbf{X}^T \mathbf{G}_2 \mathbf{U} + \beta \mathbf{G}_2 \mathbf{X} \mathbf{X}^T \mathbf{U})_{ij}}{(\mathbf{U} \mathbf{V} \mathbf{V}^T + \mathbf{X} \mathbf{X}^T \mathbf{U} + \beta^2 \mathbf{U} \mathbf{U}^T \mathbf{G}_2 \mathbf{U} + \beta^2 \mathbf{G}_2 \mathbf{U} \mathbf{U}^T \mathbf{U})_{ij}}} \quad (35)$$

$$\mathbf{V}_{ij}^{t+1} \leftarrow \mathbf{V}_{ij}^t \sqrt{\frac{(2\mathbf{U}^T \mathbf{X} + \alpha \mathbf{V} \mathbf{L}^- + \lambda \mathbf{V})_{ij}}{(\mathbf{U}^T \mathbf{U} \mathbf{V} + \mathbf{V} + \alpha \mathbf{V} \mathbf{L}^+ + \lambda \mathbf{V} \mathbf{1}_n)_{ij}}} \quad (36)$$

3.7. Convergence Analysis

In this section, the convergence of \mathbf{U} and \mathbf{V} under the update rules in Equations (35) and (36) in the ASNMF-SRP algorithm is proven using the auxiliary function method.

Firstly, we prove that the coefficient matrix \mathbf{V} is non-increasing under the update rule in Equation (36). According to the objective function Equation (24), we can derive a functional $F_{ij}(\mathbf{V}_{ij})$ regarding the coefficient matrix \mathbf{V} as follows:

$$F_{ij}(\mathbf{V}_{ij}) = \text{Tr}(\mathbf{V}^T \mathbf{U}^T \mathbf{U} \mathbf{V} - 4\mathbf{V}^T \mathbf{U}^T \mathbf{X} + \mathbf{V}^T \mathbf{V} + \alpha \mathbf{V} \mathbf{L}^+ \mathbf{V}^T - \alpha \mathbf{V} \mathbf{L}^- \mathbf{V}^T + \lambda \mathbf{V}^T \mathbf{V} \mathbf{1}_n - \lambda \mathbf{V}^T \mathbf{V}) \quad (37)$$

Proposition 1. The constructed function $G(\mathbf{V}_{ij}, \mathbf{V}_{ij}^t)$ is an auxiliary function of $F_{ij}(\mathbf{V}_{ij})$.

$$\begin{aligned} G(\mathbf{V}_{ij}, \mathbf{V}_{ij}^t) &= \sum_{ij} (\mathbf{U}^T \mathbf{U} \mathbf{V}^t)_{ij} \frac{\mathbf{V}_{ij}^2}{\mathbf{V}_{ij}^t} - 4 \sum_{ij} (\mathbf{U}^T \mathbf{X})_{ij} \mathbf{V}_{ij}^t \left(1 + \log \frac{\mathbf{V}_{ij}}{\mathbf{V}_{ij}^t} \right) \\ &\quad + \sum_{ij} \mathbf{V}_{ij}^2 + \alpha \sum_{ij} (\mathbf{V}^t \mathbf{L}^+)_{ij} \frac{\mathbf{V}_{ij}^2}{\mathbf{V}_{ij}^t} - \alpha \sum_{ijl} (\mathbf{L}_{jl}^- \mathbf{V}_{ij}^t \mathbf{V}_{il}^t) \left(1 + \log \frac{\mathbf{V}_{jl} \mathbf{V}_{il}}{\mathbf{V}_{jl}^t \mathbf{V}_{il}^t} \right) \\ &\quad + \lambda \sum_{ij} (\mathbf{V}^t \mathbf{1}_n)_{ij} \frac{\mathbf{V}_{ij}^2}{\mathbf{V}_{ij}^t} - \lambda \sum_{ij} \mathbf{V}_{ij}^t \mathbf{V}_{ij}^t \left(1 + \log \frac{\mathbf{V}_{ij} \mathbf{V}_{ij}}{\mathbf{V}_{ij}^t \mathbf{V}_{ij}^t} \right) \end{aligned} \quad (38)$$

Proof. When $\mathbf{V}_{ij} = \mathbf{V}_{ij}^t$, we have $G(\mathbf{V}_{ij}, \mathbf{V}_{ij}) = F_{ij}(\mathbf{V}_{ij})$. The following proves the case $G(\mathbf{V}_{ij}, \mathbf{V}_{ij}^t) \geq F_{ij}(\mathbf{V}_{ij})$.

When \mathbf{A} and \mathbf{B} are symmetric matrices, we have the following:

$$\begin{aligned} \sum_{i=1}^n \sum_{j=1}^m \frac{(\mathbf{A} \mathbf{S}^t \mathbf{B})_{ij} \mathbf{S}_{ij}^2}{\mathbf{S}_{ij}^t} &\geq \text{Tr}(\mathbf{S}^T \mathbf{A} \mathbf{S} \mathbf{B}), \\ \forall \mathbf{A} \in \mathbb{R}_+^{n \times n}, \mathbf{B} \in \mathbb{R}_+^{m \times m}, \mathbf{S}^t \in \mathbb{R}_+^{n \times m}, \mathbf{S} \in \mathbb{R}_+^{n \times m} \end{aligned} \quad (39)$$

From Equation (39), we obtain the following:

$$\text{Tr}(\mathbf{V}^T \mathbf{U}^T \mathbf{U} \mathbf{V}) \leq \sum_{ij} (\mathbf{U}^T \mathbf{U} \mathbf{V}^t)_{ij} \frac{\mathbf{V}_{ij}^2}{\mathbf{V}_{ij}^t} \quad (40)$$

$$\text{Tr}(\mathbf{V}^T \mathbf{V}) \leq \sum_{ij} \mathbf{V}_{ij}^2 \quad (41)$$

$$\alpha \text{Tr}(\mathbf{V} \mathbf{L}^+ \mathbf{V}^T) \leq \alpha \sum_{ij} (\mathbf{V}^t \mathbf{L}^+)_{ij} \frac{\mathbf{V}_{ij}^2}{\mathbf{V}_{ij}^t} \quad (42)$$

$$\lambda \text{Tr}(\mathbf{V}^T \mathbf{V} \mathbf{1}_n) \leq \lambda \sum_{ij} (\mathbf{V}^t \mathbf{1}_n)_{ij} \frac{\mathbf{V}_{ij}^2}{\mathbf{V}_{ij}^t} \quad (43)$$

From $x > 1 + \log x (x > 0)$, we obtain the following:

$$-4 \text{Tr}(\mathbf{V}^T \mathbf{U}^T \mathbf{X}) = -4 \text{Tr}(\mathbf{X}^T \mathbf{U} \mathbf{V}) \leq -4 \sum_{ij} (\mathbf{U}^T \mathbf{X})_{ij} \mathbf{V}_{ij}^t \left(1 + \log \frac{\mathbf{V}_{ij}}{\mathbf{V}_{ij}^t} \right) \quad (44)$$

$$-\alpha \text{Tr}(\mathbf{V} \mathbf{L}^- \mathbf{V}^T) \leq -\alpha \sum_{ijl} (\mathbf{L}_{jl}^- \mathbf{V}_{ij}^t \mathbf{V}_{il}^t) \left(1 + \log \frac{\mathbf{V}_{jl} \mathbf{V}_{il}}{\mathbf{V}_{jl}^t \mathbf{V}_{il}^t} \right) \quad (45)$$

$$-\lambda \text{Tr}(\mathbf{V}^T \mathbf{V}) \leq -\lambda \sum_{ij} \mathbf{V}_{ij}^t \mathbf{V}_{ij}^t \left(1 + \log \frac{\mathbf{V}_{ij} \mathbf{V}_{ij}}{\mathbf{V}_{ij}^t \mathbf{V}_{ij}^t} \right) \quad (46)$$

From Equations (40)–(46), it follows that $G(\mathbf{V}_{ij}, \mathbf{V}_{ij}^t) \geq F_{ij}(\mathbf{V}_{ij})$. Therefore, $G(\mathbf{V}_{ij}, \mathbf{V}_{ij}^t)$ is an auxiliary function of $F_{ij}(\mathbf{V}_{ij})$, and Proposition 1 is proven.

Let $\frac{\partial G(\mathbf{V}, \mathbf{V}^t)}{\partial \mathbf{V}_{ij}} = 0$, we obtain the following:

$$\mathbf{V}_{ij}^{t+1} \leftarrow \mathbf{V}_{ij}^t \sqrt{\frac{(2\mathbf{U}^T \mathbf{X} + \alpha \mathbf{V} \mathbf{L}^- + \lambda \mathbf{V})_{ij}}{(\mathbf{U}^T \mathbf{U} \mathbf{V} + \mathbf{V} + \alpha \mathbf{V} \mathbf{L}^+ + \lambda \mathbf{V} \mathbf{1}_n)_{ij}}} \quad (47)$$

Thus, the coefficient matrix \mathbf{V} in the ASNMF-SRP algorithm is non-increasing under the update rule in Equation (36). Similarly, it can be proven that the basis matrix \mathbf{U} is non-increasing under the update rule in Equation (35). Therefore, \mathbf{U} and \mathbf{V} converge under the update rules in Equations (35) and (36). The specific implementation of the ASNMF-SRP algorithm is described in Algorithm 1.

Algorithm 1 ASNMF-SRP

Input: Initial matrix $\mathbf{X} = (x_1, x_2, \dots, x_n) \in \mathbb{R}_+^{m \times n}$, number of classes r , neighborhood parameter k , regularization parameters α , β and λ , balance parameters w_1 and w_2 parameter d , threshold ε , maximum iterations t .

Output: Basis matrix \mathbf{U} and coefficient matrix \mathbf{V} .

1. Initialization: $t = 0$, Randomly generate basis matrix $\mathbf{U} \in \mathbb{R}^{m \times r}$ and coefficient matrix $\mathbf{V} \in \mathbb{R}^{r \times n}$;
 2. Obtain optimal Laplacian matrix \mathbf{L} according to Equations (11)–(17);
 3. For $t = 1, 2, 3, \dots, \text{maxIter}$
 4. $\mathbf{U}_{ij}^{t+1} \leftarrow \mathbf{U}_{ij}^t \sqrt{\frac{(2\mathbf{X} \mathbf{V}^T + \beta \mathbf{X} \mathbf{X}^T \mathbf{G}_2 \mathbf{U} + \beta \mathbf{G}_2 \mathbf{X} \mathbf{X}^T \mathbf{U})_{ij}}{(\mathbf{U} \mathbf{V} \mathbf{V}^T + \mathbf{X} \mathbf{X}^T \mathbf{U} + \beta^2 \mathbf{U} \mathbf{U}^T \mathbf{G}_2 \mathbf{U} + \beta^2 \mathbf{G}_2 \mathbf{U} \mathbf{U}^T \mathbf{U})_{ij}}}$;
 5. $\mathbf{V}_{ij}^{t+1} \leftarrow \mathbf{V}_{ij}^t \sqrt{\frac{(2\mathbf{U}^T \mathbf{X} + \alpha \mathbf{V} \mathbf{L}^- + \lambda \mathbf{V})_{ij}}{(\mathbf{U}^T \mathbf{U} \mathbf{V} + \mathbf{V} + \alpha \mathbf{V} \mathbf{L}^+ + \lambda \mathbf{V} \mathbf{1}_n)_{ij}}}$;
 6. if $\|\mathbf{U}^t - \mathbf{U}^{t-1}\|_\infty \leq \varepsilon$ and $\|\mathbf{V}^t - \mathbf{V}^{t-1}\|_\infty \leq \varepsilon$
Break and return (\mathbf{U}, \mathbf{V}) ;
 7. End if
 8. End for
-

□

3.8. Time Complexity Analysis

When inputting a data matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$, where m is the feature dimension of samples and n is the number of samples, assume the number of classes for this dataset is r , and $r \ll \min\{m, n\}$. After t iterations, the time complexity of ASNMF-SRP is as follows:

(1) The autoencoder-like NMF part: complexities for updating \mathbf{U} and \mathbf{V} are $O(tm^2r + tmr^2)$ and $O(tm^2r + tmr^2)$;

(2) The higher-order graph regularization part: complexity is $O(mn^2 + me^d)$, where e denotes the average degree of nodes;

(3) The feature preservation part: complexity is $O(tm^2r + t(m^2 + m))$;

(4) The sparse constraint term: complexity is $O(tn^2r)$.

Therefore, the overall time complexity of ASNMF-SRP is $O(tm^2r + t(m^2 + m) + tn^2r + mn^2 + me^d)$.

4. Experiments

In this section, clustering experiments will be conducted on 8 public datasets to evaluate the clustering performance of the ASNMF-SRP algorithm. The experiments are implemented in Python 3.11 with the computer environment being Intel(R) Core (TM) i5-1135G7 @ 2.40 GHz 2.42 GHz, 16 GB RAM, and Windows 11 64-bit operating system.

4.1. Dataset

To verify the effectiveness of the algorithm, we selected 8 public datasets for comparative experiments. MSRA25 [37] dataset contains 1799 face images from 12 individuals. Semeion, Krvs, PenDigits, and Vehicle datasets are from the UCI Machine Learning Repository (<https://archive.ics.uci.edu> (accessed on 23 March 2025)). Hitech dataset is from the CLUTO toolkit (<https://conservancy.umn.edu/items/4fbef165-f964-41ed-a239-86a8f931ffbe> (accessed on 26 March 2025)). COIL20 (<http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php> (accessed on 15 December 2024)) and COIL100 (<http://www.kaggle.com/jessicali9530/coil100/downloads/coil100.zip/2> (accessed on 15 December 2024)) datasets record images of 20 and 100 objects, respectively. The basic information of these 8 datasets is shown in Table 2.

Table 2. Introduction of the 8 datasets.

NO.	Dataset	Samples (n)	Features (m)	Classes (r)	Data Type	Image Size
1	MSRA25	1799	256	12	Face dataset	16×16
2	Semeion	1593	256	10	Digit images	16×16
3	COIL20	1440	1024	20	Object images	32×32
4	COIL100	7200	1024	100	Object images	32×32
5	Krvs	3196	36	2	Network detection	—
6	Hitech	2301	2216	6	Technology news	—
7	PenDigits	3498	16	10	Handwritten digits	—
8	Vehicle	846	18	4	Vehicle contours	—

4.2. Clustering Performance Evaluation Metrics

To compare the clustering performance of various algorithms, this paper selects four commonly used clustering performance evaluation metrics. Among them, larger values for these four clustering evaluation metrics indicate better clustering performance of the corresponding algorithm.

4.2.1. Clustering Accuracy (ACC)

$$ACC = \frac{\sum_{i=1}^n \delta(\text{map}(s_i), r_i)}{n} \quad (48)$$

where n is the total number of samples in the input data, r_i is the true class label of the data, s_i represents the result after clustering by the algorithm, and $\text{map}(\cdot)$ denotes the mapping function. The expression $\delta(\text{map}(s_i), r_i)$ in Equation (48) is as follows:

$$\delta(\text{map}(s_i), r_i) = \begin{cases} 1, & \text{map}(s_i) = r_i \\ 0, & \text{map}(s_i) \neq r_i \end{cases} \quad (49)$$

From Equations (48) and (49), ACC is an evaluation metric used to measure the consistency between clustering results and true labels, with a value range of $[0, 1]$.

4.2.2. Adjusted Rand Index (ARI)

$$\text{ARI} = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{x_i}{2} \sum_j \binom{y_j}{2} \right] / \binom{n}{2}}{0.5 \left[\sum_i \binom{x_i}{2} + \sum_j \binom{y_j}{2} \right] - \left[\sum_i \binom{x_i}{2} \sum_j \binom{y_j}{2} \right] / \binom{n}{2}} \quad (50)$$

where x_i represents the total number of samples in the i -th cluster after clustering, y_j represents the total number of samples in the true j -th class, and n_{ij} denotes the total number of samples common to both the cluster x_i and the true class y_j . From Equation (50), the value range of ARI is $[-1, 1]$.

4.2.3. Normalized Mutual Information (NMI)

$$\text{NMI}(N, N^*) = \frac{\text{MI}(N, N^*)}{\max(H(N), H(N^*))} \quad (51)$$

where N^* represents the data labels obtained through clustering, N represents the actual data labels, and $H(\cdot)$ is defined as the entropy function.

The mutual information $\text{MI}(\cdot)$ expression is as follows:

$$\text{MI}(N, N^*) = \sum_{n_i \in N, n_j^* \in N^*} p(n_i, n_j^*) \log \frac{p(n_i, n_j^*)}{p(n_i)p(n_j^*)} \quad (52)$$

where $p(n_i)$ and $p(n_j^*)$ represent the marginal probabilities of the true class and clustering results, respectively, and $p(n_i, n_j^*)$ is the joint probability of the true class n_i and cluster n_j^* . From Equations (51) and (52), the value range of NMI is $[0, 1]$.

4.2.4. Clustering Purity (PUR)

$$\text{PUR} = \sum_{i=1}^r \frac{\max_j (n_i^j)}{n} \quad (53)$$

where n_i represents the total number of samples, r is the number of clusters, and n_i^j represents the number of samples in cluster i that belong to the true class j . From Equation (53), the value range of PUR is $[0, 1]$.

4.3. Comparison Algorithms and Parameter Settings

The detailed descriptions of the 9 algorithms participating in the comparative experiments are as follows:

(1) NMF [6] factorizes the original non-negative matrix into the product of two non-negative low-rank submatrices to achieve the goal of representing the whole by parts.

(2) ONMF [9] imposes orthogonality constraints on the factorized submatrices based on NMF.

(3) Hx-NMF [38] combines logarithm with NMF to improve the robustness of the algorithm.

(4) EMMF [39] is an algorithm based on entropy minimization matrix factorization.

(5) GNMF [11] encodes the local geometric structure between data into the objective function to improve the quality of low-dimensional representations.

(6) RMMMF [40] uses the $l_{2,1}$ -norm to enhance the algorithm's robustness.

(7) DRCC [27] jointly optimizes the local geometric structures of samples and features by incorporating them into the objective function.

(8) FR-NMF [28] improves the clustering performance of the algorithm by introducing a feature preservation term.

(9) LS-NMF [24] imposes sparse constraints on the basis matrix and coefficient matrix using the l_{log} -(pseudo) norm.

In the experimental process, we perform a grid search over the algorithm's regularization parameter in the range $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0, 10^1, 10^2, 10^3\}$.

During the experiments, the parameter values for the higher-order graph regularization parameter α , feature relationship preservation regularization parameter β , and sparse constraint regularization parameter λ in the ASNMF-SRP algorithm are shown in Table 3.

Table 3. Parameter values of ASNMF-SRP on different datasets.

No.	Dataset	Higher-Order Graph Regularization α	Feature Relationship Preservation β	Sparse Constraint λ
1	MSRA25	1000	1000	0.001
2	Semeion	1	100	0.0001
3	COIL20	1000	1000	0.001
4	COIL100	1000	1000	0.001
5	Krvs	100	10	0.0001
6	Hitech	1000	10	0.0001
7	PenDigits	100	100	0.001
8	Vehicle	1	1	0.001

4.4. Results and Analysis

Due to fluctuations in the clustering results of various algorithms, to obtain representative clustering outcomes, each algorithm was executed 20 times on the 8 datasets during the experiments. The mean and standard deviation were calculated as the final comparative clustering results. Tables 4–7 present the specific values of ACC, ARI, NMI, and PUR for these 10 algorithms on the 8 datasets, respectively. In the tables, bold numbers indicate the optimal clustering results on that dataset, and the I-P (Improvement Percentage) column shows the percentage by which the ASNMF-SRP algorithm improves over the best result among the other 9 algorithms.

Table 4. ACC of 10 algorithms on 8 datasets (mean \pm standard deviation).

Algorithm Dataset	NMF	ONMF	Hx-NMF	EMMF	GNMF	RMNMF	DRCC	FR-NMF	LS-NMF	ASNMF-SRP	I-P
MSRA25	0.50842 ± 0.023	0.49375 ± 0.026	0.52026 ± 0.029	0.50592 ± 0.018	0.53938 ± 0.032	0.55550 ± 0.034	0.28974 ± 0.029	0.51659 ± 0.022	0.54019 ± 0.022	0.57904 ± 0.045	4.24% --
Semeion	0.52508 ± 0.042	0.56959 ± 0.032	0.50807 ± 0.042	0.51965 ± 0.040	0.59209 ± 0.039	0.27916 ± 0.057	0.61601 ± 0.037	0.53540 ± 0.037	0.60251 ± 0.036	0.68063 ± 0.050	10.49% --
COIL20	0.66406 ± 0.029	0.68531 ± 0.028	0.65812 ± 0.030	0.65142 ± 0.019	0.76844 ± 0.013	0.24385 ± 0.094	0.79017 ± 0.034	0.65028 ± 0.026	0.77361 ± 0.014	0.84174 ± 0.012	6.53% --
COIL100	0.47026 ± 0.014	0.49159 ± 0.011	0.46877 ± 0.012	0.47882 ± 0.018	0.48738 ± 0.014	0.41235 ± 0.012	0.46099 ± 0.014	0.47044 ± 0.015	0.48306 ± 0.011	0.64173 ± 0.010	30.54% --
Krvs	0.51909 ± 0.003	0.53742 ± 0.012	0.52223 ± 0.002	0.52137 ± 0.003	0.53082 ± 0.012	0.51810 ± 0.007	0.55594 ± 0.004	0.52552 ± 0.023	0.53387 ± 0.015	0.56813 ± 0.003	2.19% --
Hitech	0.23385 ± 0.002	0.23375 ± 0.005	0.23403 ± 0.008	0.23544 ± 0.004	0.23105 ± 0.004	0.26273 ± 0.001	0.24087 ± 0.004	0.22603 ± 0.007	0.23268 ± 0.002	0.25367 ± 0.002	−3.45% --
PenDigits	0.66216 ± 0.036	0.70729 ± 0.048	0.68092 ± 0.040	0.66630 ± 0.038	0.67973 ± 0.053	0.65183 ± 0.037	0.73119 ± 0.043	0.66791 ± 0.034	0.68533 ± 0.052	0.80442 ± 0.032	10.02% --
Vehicle	0.38794 ± 0.019	0.43777 ± 0.002	0.40142 ± 0.026	0.39096 ± 0.021	0.44397 ± 0.008	0.35916 ± 0.003	0.41194 ± 0.017	0.43570 ± 0.023	0.44368 ± 0.009	0.45236 ± 0.002	1.89% --

Table 5. ARI of 10 algorithms on 8 datasets (mean \pm standard deviation).

Algorithm Dataset	NMF	ONMF	Hx- NMF	EMMF	GNMF	RMNMF	DRCC	FR- NMF	LS- NMF	ASNMF- SRP	I-P
MSRA25	0.34575	0.32933	0.35734	0.33482	0.40605	0.40058	0.12013	0.35562	0.40596	0.44662	9.99%
	± 0.026	± 0.027	± 0.027	± 0.020	± 0.037	± 0.034	± 0.023	± 0.019	± 0.034	± 0.052	--
Semeion	0.31198	0.35051	0.30655	0.31134	0.44280	0.11855	0.41651	0.31943	0.45921	0.48809	6.29%
	± 0.033	± 0.024	± 0.031	± 0.030	± 0.032	± 0.047	± 0.026	± 0.029	± 0.030	± 0.034	--
COIL20	0.57989	0.62582	0.57627	0.57069	0.74160	0.16450	0.73372	0.56682	0.74234	0.80244	8.10%
	± 0.026	± 0.023	± 0.034	± 0.026	± 0.018	± 0.093	± 0.036	± 0.024	± 0.016	± 0.007	--
COIL100	0.39584	0.44157	0.39549	0.40697	0.42371	0.30329	0.39167	0.39709	0.42067	0.53573	21.32%
	± 0.016	± 0.014	± 0.017	± 0.019	± 0.010	± 0.015	± 0.016	± 0.012	± 0.011	± 0.016	--
Krvs	0.00107	0.00579	0.00158	0.00147	0.00393	−0.00038	0.01201	0.00410	0.00505	0.01814	51.04%
	± 0.000	± 0.004	± 0.000	± 0.000	± 0.003	± 0.001	± 0.002	± 0.006	± 0.005	± 0.002	--
Hitech	−0.00095	0.00092	0.00021	0.00034	−0.00041	0.00016	0.00252	0.00066	−0.00049	0.00689	173.41%
	± 0.001	± 0.001	± 0.001	± 0.001	± 0.001	± 0.001	± 0.002	± 0.001	± 0.001	± 0.000	--
PenDigits	0.52361	0.55428	0.52352	0.52329	0.54137	0.50238	0.57474	0.52889	0.55809	0.68487	19.16%
	± 0.029	± 0.040	± 0.037	± 0.022	± 0.045	± 0.049	± 0.037	± 0.024	± 0.039	± 0.035	--
Vehicle	0.08169	0.12462	0.09207	0.08322	0.13103	0.06364	0.09691	0.11321	0.13221	0.12014	−9.13%
	± 0.013	± 0.003	± 0.018	± 0.018	± 0.009	± 0.004	± 0.014	± 0.012	± 0.007	± 0.001	--

Table 6. NMI of 10 algorithms on 8 datasets (mean \pm standard deviation).

Algorithm Dataset	NMF	ONMF	Hx- NMF	EMMF	GNMF	RMNMF	DRCC	FR- NMF	LS- NMF	ASNMF- SRP	I-P
MSRA25	0.56935	0.56296	0.57773	0.55467	0.65111	0.60295	0.23745	0.57715	0.64613	0.71512	9.83%
	± 0.021	± 0.028	± 0.023	± 0.021	± 0.031	± 0.029	± 0.033	± 0.017	± 0.031	± 0.026	--
Semeion	0.44162	0.48847	0.44312	0.44938	0.60790	0.20171	0.54014	0.44938	0.61489	0.63282	2.92%
	± 0.025	± 0.020	± 0.026	± 0.023	± 0.020	± 0.074	± 0.018	± 0.022	± 0.019	± 0.021	--
COIL20	0.76112	0.79591	0.76067	0.75423	0.88538	0.31375	0.89131	0.75546	0.88500	0.91529	2.69%
	± 0.015	± 0.010	± 0.018	± 0.017	± 0.012	± 0.131	± 0.011	± 0.015	± 0.012	± 0.006	--
COIL100	0.75258	0.76835	0.75400	0.75646	0.77226	0.70061	0.74641	0.73117	0.76948	0.83835	8.56%
	± 0.005	± 0.005	± 0.006	± 0.006	± 0.004	± 0.009	± 0.006	± 0.005	± 0.004	± 0.003	--
Krvs	0.00060	0.00397	0.00091	0.00094	0.00265	0.00203	0.00818	0.00592	0.00352	0.01250	52.81%
	± 0.000	± 0.003	± 0.000	± 0.000	± 0.002	± 0.003	± 0.001	± 0.006	± 0.003	± 0.001	--
Hitech	0.00799	0.00989	0.00854	0.01049	0.00865	0.00558	0.01203	0.00786	0.00799	0.01935	60.85%
	± 0.001	± 0.002	± 0.002	± 0.001	± 0.002	± 0.001	± 0.002	± 0.002	± 0.002	± 0.000	--
PenDigits	0.68251	0.69332	0.66576	0.67684	0.69751	0.64294	0.69615	0.68480	0.70953	0.80120	12.92%
	± 0.022	± 0.019	± 0.026	± 0.018	± 0.027	± 0.043	± 0.016	± 0.021	± 0.018	± 0.021	--
Vehicle	0.11678	0.19000	0.13648	0.12714	0.19062	0.08408	0.14858	0.16404	0.19181	0.18544	−3.32%
	± 0.014	± 0.006	± 0.019	± 0.018	± 0.015	± 0.005	± 0.019	± 0.018	± 0.013	± 0.000	--

Table 7. PUR of 10 algorithms on 8 datasets (mean \pm standard deviation).

Algorithm Dataset	NMF	ONMF	Hx- NMF	EMMF	GNMF	RMNMF	DRCC	FR- NMF	LS- NMF	ASNMF- SRP	I-P
MSRA25	0.52985	0.52176	0.53755	0.52287	0.56587	0.57918	0.30698	0.53849	0.56651	0.61479	6.15%
	± 0.021	± 0.023	± 0.025	± 0.020	± 0.026	± 0.027	± 0.030	± 0.017	± 0.025	± 0.035	--
Semeion	0.53763	0.58804	0.53431	0.53807	0.63726	0.28738	0.63625	0.54862	0.64369	0.69739	8.34%
	± 0.036	± 0.025	± 0.030	± 0.031	± 0.024	± 0.061	± 0.027	± 0.028	± 0.026	± 0.031	--
COIL20	0.69135	0.71340	0.68628	0.67997	0.80715	0.24903	0.82802	0.67753	0.80892	0.86455	4.41%
	± 0.024	± 0.023	± 0.021	± 0.021	± 0.017	± 0.095	± 0.017	± 0.023	± 0.018	± 0.011	--
COIL100	0.52663	0.54705	0.52816	0.53379	0.54654	0.47951	0.51849	0.51637	0.54247	0.69599	27.23%
	± 0.013	± 0.010	± 0.011	± 0.013	± 0.012	± 0.012	± 0.012	± 0.012	± 0.009	± 0.006	--
Krvs	0.52245	0.53742	0.52289	0.52261	0.53137	0.52237	0.55594	0.53360	0.53387	0.56813	2.19%
	± 0.000	± 0.012	± 0.001	± 0.000	± 0.011	± 0.000	± 0.004	± 0.016	± 0.015	± 0.003	--
Hitech	0.26693	0.27034	0.26758	0.27017	0.26877	0.26380	0.27099	0.26788	0.26606	0.28525	5.26%
	± 0.003	± 0.003	± 0.003	± 0.003	± 0.003	± 0.001	± 0.004	± 0.005	± 0.003	± 0.001	--
PenDigits	0.69262	0.72340	0.69447	0.69118	0.70675	0.67226	0.73872	0.69626	0.71256	0.81095	9.78%
	± 0.026	± 0.031	± 0.031	± 0.026	± 0.035	± 0.035	± 0.029	± 0.024	± 0.035	± 0.023	--
Vehicle	0.39285	0.43777	0.40573	0.39681	0.44397	0.37145	0.41832	0.43853	0.44368	0.45236	1.89%
	± 0.015	± 0.002	± 0.020	± 0.018	± 0.008	± 0.005	± 0.020	± 0.022	± 0.009	± 0.002	--

By comparing the data in Tables 4–7, it can be seen that ASNMF-SRP exhibits excellent clustering performance. The specific data analysis is as follows:

(1) On image datasets (MSRA25, Semeion, COIL20, and COIL100), the values of all four clustering evaluation metrics for ASNMF-SRP are higher than those of the other nine algorithms. This indicates that ASNMF-SRP holds certain advantages in handling clustering problems related to images.

(2) On non-image datasets (Krvs, Hitech, PenDigits, and Vehicle), ASNMF-SRP fails to achieve optimal values in ACC on the Hitech dataset and in ARI and NMI on the Vehicle dataset. However, in most cases, ASNMF-SRP demonstrates optimal performance.

(3) FR-NMF incorporates an additional feature relationship preservation term compared to NMF. According to the data in Tables 4–7, FR-NMF outperforms NMF in clustering performance on some datasets. ASNMF-SRP not only includes feature relationship preservation but also incorporates components such as autoencoder-like NMF, higher-order graph regularization, and sparse constraints. The data in Tables 4–7 show that ASNMF-SRP achieves higher values across all four clustering evaluation metrics on these eight datasets compared to NMF and FR-NMF. This demonstrates that through the synergistic effects of various regularization terms, ASNMF-SRP not only enhances clustering performance but also improves generalization capability.

This paper employs the t-SNE method to project the low-dimensional representation matrices obtained by the ten algorithms on the MSRA25, COIL20, and Hitech datasets into a two-dimensional space for visualization.

From Figures 5–7, it can be observed that on the MSRA25, COIL20, and Hitech datasets, the low-rank representation matrix obtained after dimensionality reduction using the ASNMF-SRP algorithm exhibits stronger discriminative power. Therefore, the ASNMF-SRP algorithm demonstrates excellent clustering performance.

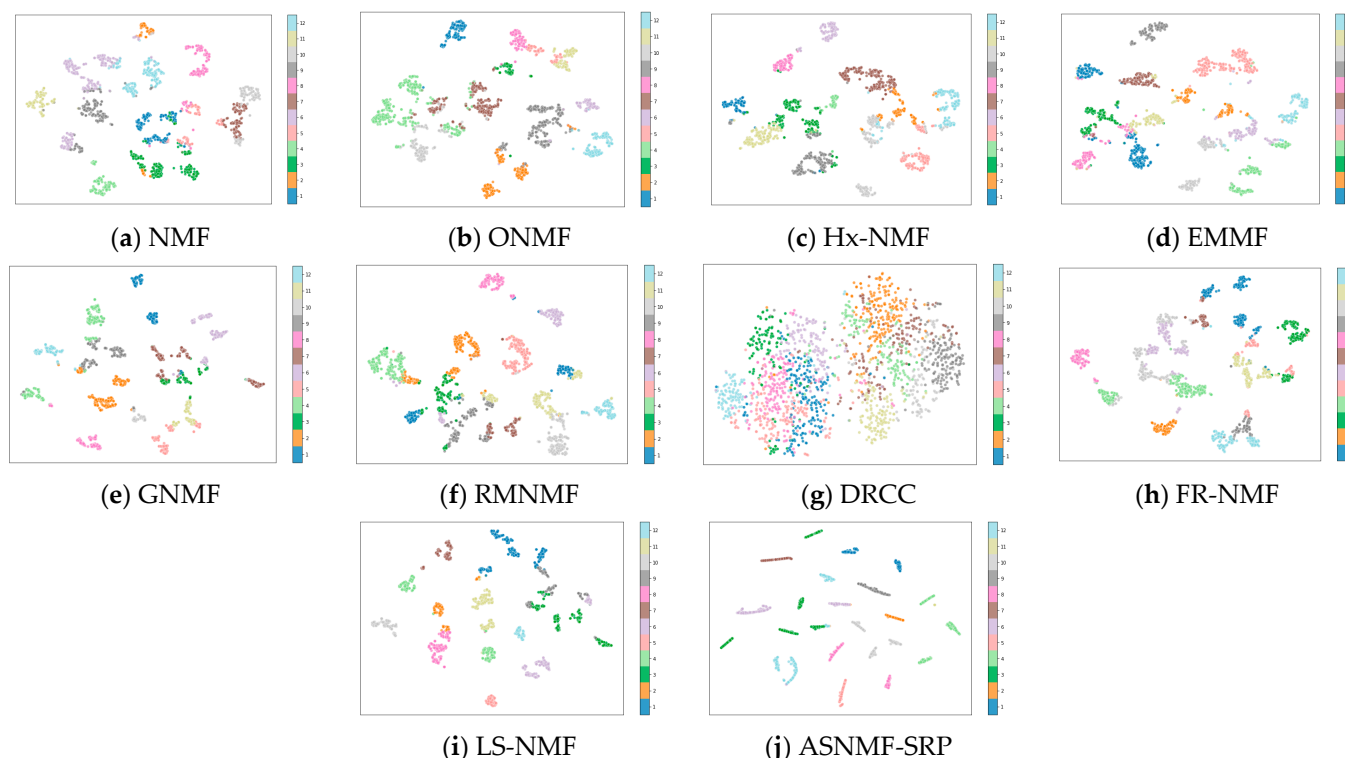


Figure 5. Visual comparison of low-dimensional representation matrices of various algorithms on the MSRA25 dataset.

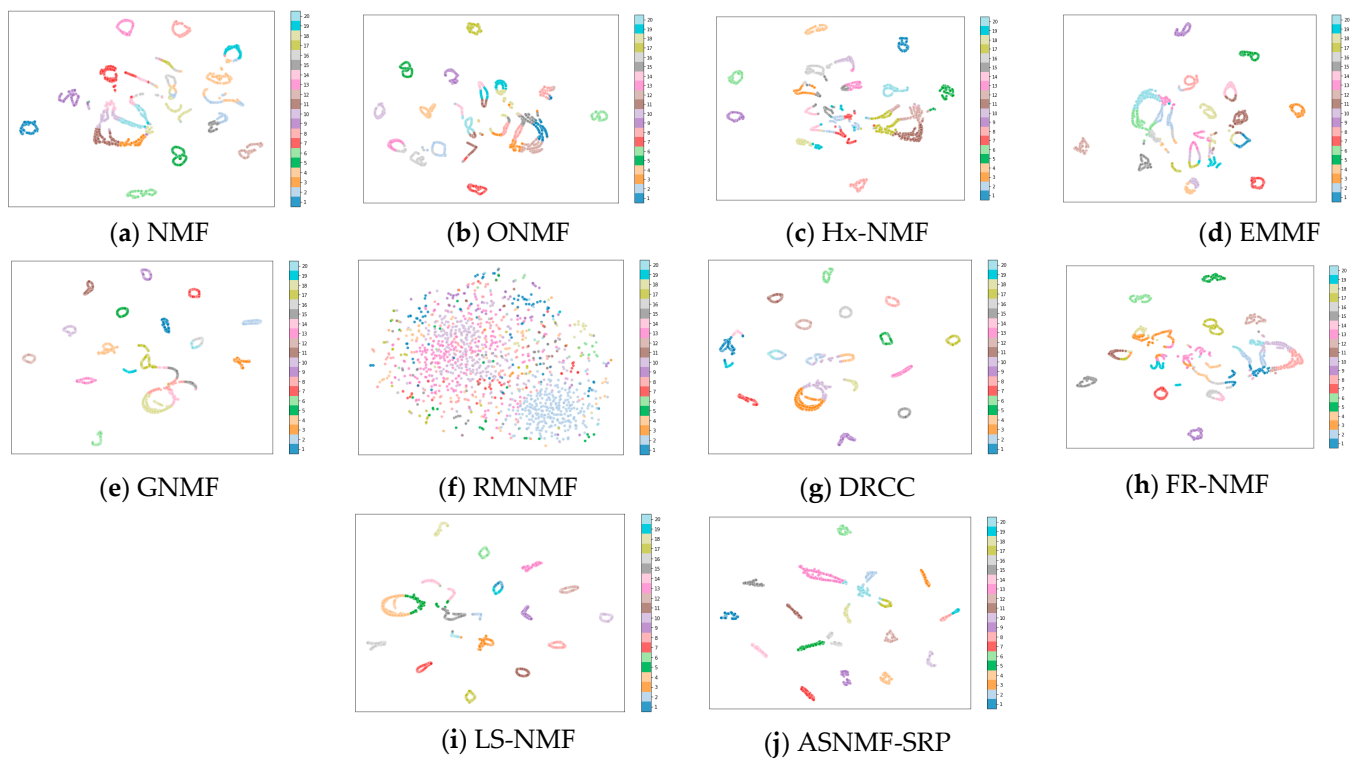


Figure 6. Visual comparison of low-dimensional representation matrices of various algorithms on the COIL20 dataset.

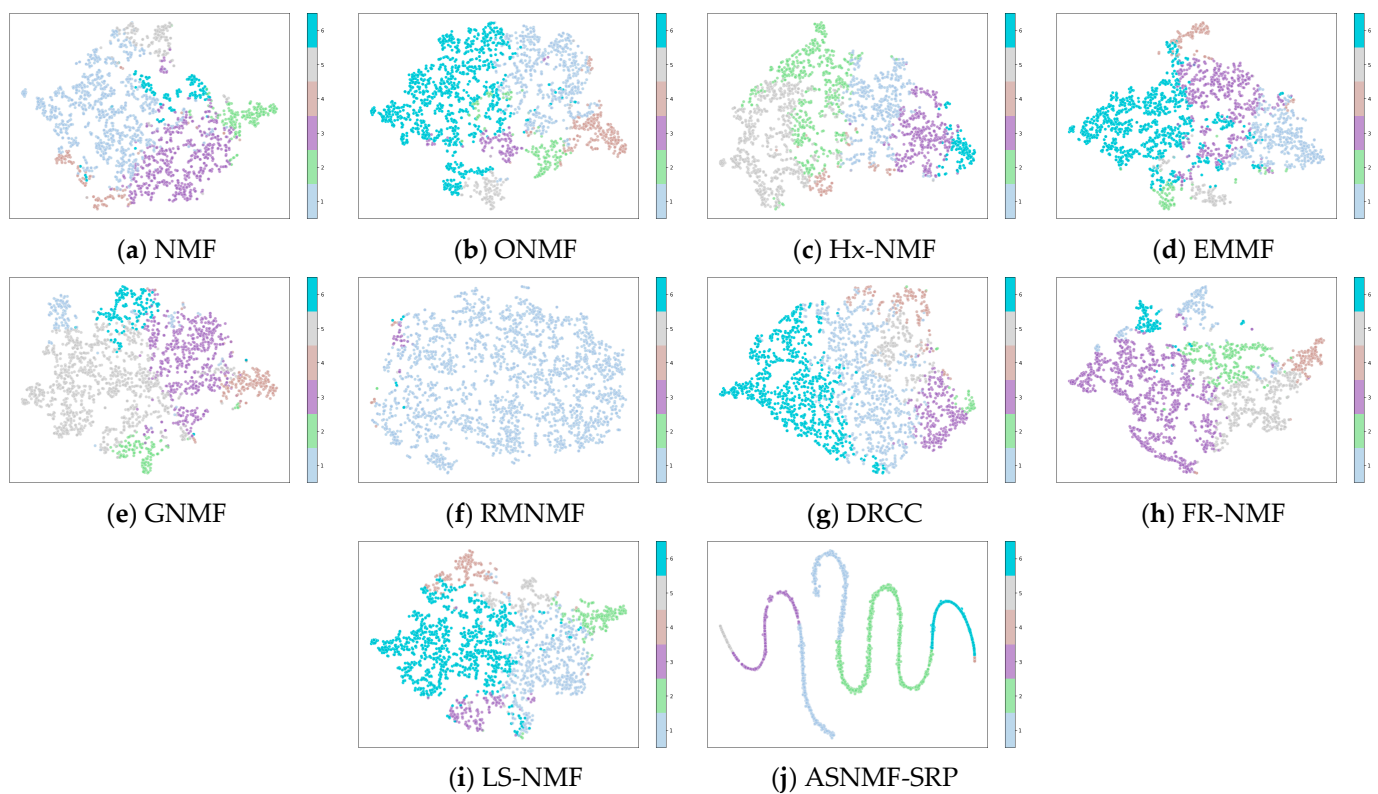


Figure 7. Visual comparison of low-dimensional representation matrices of various algorithms on the Hitech dataset.

4.5. Analysis of the Impact of Autoencoder-like NMF on Clustering Performance

Unlike traditional NMF, the objective function of ASNMF-SRP includes a “decoder–encoder” module. To investigate the influence of this autoencoder-like NMF architecture on clustering performance, we removed the “encoder” part in Equation (24), causing ASNMF-SRP to degenerate into ASNMF-SRP-1. The objective function of ASNMF-SRP-1 is as follows:

$$\min_{\mathbf{U}, \mathbf{V} \geq 0} \|\mathbf{X} - \mathbf{UV}\|_F^2 + \alpha \text{Tr}(\mathbf{VLV}^T) + \|\mathbf{XX}^T - \beta \mathbf{UU}^T\|_{2,1} + \lambda (\text{Tr}(\mathbf{V}^T \mathbf{V} \mathbf{1}_n) - \text{Tr}(\mathbf{V}^T \mathbf{V})) \quad (54)$$

From Table 8, it can be seen that the clustering performance of ASNMF-SRP is significantly better than that of ASNMF-SRP-1. On the clustering evaluation metrics ACC, ARI, NMI, and PUR, it shows average improvements of 12.03%, 17.02%, 10.81%, and 10.56%, respectively. This indicates that introducing an NMF component with a structure similar to an autoencoder has a positive effect on ASNMF-SRP, effectively enhancing its clustering performance.

Table 8. Clustering results of ASNMF-SRP and ASNMF-SRP-1 (mean \pm standard deviation).

Dataset	ACC		ARI		NMI		PUR	
	ASNMF-SRP-1	ASNMF-SRP	ASNMF-SRP-1	ASNMF-SRP	ASNMF-SRP-1	ASNMF-SRP	ASNMF-SRP-1	ASNMF-SRP
MSRA25	0.49680	0.57904	0.33356	0.44662	0.55612	0.71512	0.51656	0.61479
	± 0.021	± 0.045	± 0.016	± 0.052	± 0.019	± 0.026	± 0.019	± 0.035
Semeion	0.64724	0.68063	0.48198	0.48809	0.64292	0.63282	0.68908	0.69739
	± 0.010	± 0.050	± 0.007	± 0.034	± 0.010	± 0.021	± 0.011	± 0.031
COIL20	0.80552	0.84174	0.76313	0.80244	0.88651	0.91529	0.83587	0.86455
	± 0.015	± 0.012	± 0.017	± 0.007	± 0.008	± 0.006	± 0.013	± 0.011
COIL100	0.48007	0.64173	0.40311	0.53573	0.73255	0.83835	0.52556	0.69599
	± 0.009	± 0.010	± 0.011	± 0.016	± 0.005	± 0.003	± 0.007	± 0.006
Krvs	0.52839	0.56813	0.00393	0.01814	0.00396	0.01250	0.53339	0.56813
	± 0.017	± 0.003	± 0.004	± 0.002	± 0.004	± 0.001	± 0.012	± 0.003
Hitech	0.23301	0.25367	0.00014	0.00689	0.00893	0.01935	0.26847	0.28525
	± 0.005	± 0.002	± 0.001	± 0.000	± 0.001	± 0.000	± 0.003	± 0.001
PenDigits	0.66533	0.80442	0.53130	0.68487	0.68684	0.80120	0.69495	0.81095
	± 0.040	± 0.032	± 0.030	± 0.035	± 0.020	± 0.021	± 0.026	± 0.023
Vehicle	0.44746	0.45236	0.13445	0.12014	0.20032	0.18544	0.44888	0.45236
	± 0.004	± 0.002	± 0.003	± 0.001	± 0.008	± 0.000	± 0.006	± 0.002

4.6. Analysis of the Impact of Higher-Order Graph Regularization on Clustering Performance

To further examine the influence of higher-order neighborhood relationships among samples on clustering performance, we set $w_2 = 0$ in Equation (17), thereby reducing ASNMF-SRP to ASNMF-SRP-2. At this point, the objective function of ASNMF-SRP-2 is as follows:

$$\min_{\mathbf{U}, \mathbf{V} \geq 0} \|\mathbf{X} - \mathbf{UV}\|_F^2 + \|\mathbf{V} - \mathbf{U}^T \mathbf{X}\|_F^2 + \alpha \text{Tr}(\mathbf{VL}_1 \mathbf{V}^T) + \|\mathbf{XX}^T - \beta \mathbf{UU}^T\|_{2,1} + \lambda (\text{Tr}(\mathbf{V}^T \mathbf{V} \mathbf{1}_n) - \text{Tr}(\mathbf{V}^T \mathbf{V})) \quad (55)$$

From Table 9, it can be observed that ASNMF-SRP exhibits superior clustering performance over ASNMF-SRP-2 on most datasets, indicating its stronger generalization ability. This also suggests that higher-order neighborhood relationships among samples can influence the clustering performance of the algorithm. However, on a few datasets (e.g., Krvs), ASNMF-SRP does not perform well. Therefore, future work should further explore how to

design an optimal Laplacian matrix to effectively encode multi-order topological structure information, thereby improving the algorithm’s universality.

Table 9. Clustering results of ASNMF-SRP and ASNMF-SRP-2 (mean \pm standard deviation).

Dataset	ACC		ARI		NMI		PUR	
	ASNMF-SRP-2	ASNMF-SRP	ASNMF-SRP-2	ASNMF-SRP	ASNMF-SRP-2	ASNMF-SRP	ASNMF-SRP-2	ASNMF-SRP
MSRA25	0.57518	0.57904	0.44437	0.44662	0.72012	0.71512	0.61354	0.61479
	± 0.032	± 0.045	± 0.038	± 0.052	± 0.019	± 0.026	± 0.025	± 0.035
Semeion	0.67803	0.68063	0.48380	0.48809	0.62883	0.63282	0.69567	0.69739
	± 0.044	± 0.050	± 0.029	± 0.034	± 0.018	± 0.021	± 0.025	± 0.031
COIL20	0.83865	0.84174	0.80155	0.80244	0.91285	0.91529	0.86194	0.86455
	± 0.013	± 0.012	± 0.008	± 0.007	± 0.004	± 0.006	± 0.013	± 0.011
COIL100	0.63818	0.64173	0.54346	0.53573	0.83103	0.83835	0.69085	0.69599
	± 0.007	± 0.010	± 0.014	± 0.016	± 0.003	± 0.003	± 0.004	± 0.006
Krvs	0.56884	0.56813	0.01850	0.01814	0.01274	0.01250	0.56884	0.56813
	± 0.000	± 0.003	± 0.000	± 0.002	± 0.000	± 0.001	± 0.000	± 0.003
Hitech	0.24744	0.25367	0.00424	0.00689	0.01893	0.01935	0.27912	0.28525
	± 0.001	± 0.002	± 0.001	± 0.000	± 0.000	± 0.000	± 0.001	± 0.001
PenDigits	0.79936	0.80442	0.67517	0.68487	0.79476	0.80120	0.80499	0.81095
	± 0.029	± 0.032	± 0.028	± 0.035	± 0.017	± 0.021	± 0.022	± 0.023
Vehicle	0.45219	0.45236	0.12018	0.12014	0.18540	0.18544	0.45219	0.45236
	± 0.001	± 0.002	± 0.001	± 0.001	± 0.001	± 0.000	± 0.001	± 0.002

4.7. Robustness Analysis of ASNMF-SRP

To evaluate the robustness of ASNMF-SRP, we added salt-and-pepper noise with intensities of 10%, 20%, 30%, and 40% to the MSRA25, COIL20, and COIL100 datasets. To further compare the contributions of Equations (19) and (20) to the robustness of ASNMF-SRP, we conducted comparative experiments between ASNMF-SRP and ASNMF-SRP-3. The objective function of ASNMF-SRP-3 is as follows:

$$\min_{U, V \geq 0} \|X - UV\|_F^2 + \|V - U^T X\|_F^2 + \alpha \text{Tr}(VLV^T) + \|XX^T - \beta uu^T\|_F^2 + \lambda (\text{Tr}(V^T V \mathbf{1}_n) - \text{Tr}(V^T V)) \quad (56)$$

From Figure 8, it can be seen that as the noise intensity increases, the clustering evaluation metric values of both ASNMF-SRP and ASNMF-SRP-3 show a declining trend. However, the clustering evaluation metric values of ASNMF-SRP are consistently higher than those of ASNMF-SRP-3. Therefore, adopting the $l_{2,1}$ -norm to measure reconstruction errors in the “feature relationship preservation” component can enhance the robustness of ASNMF-SRP.

4.8. Parameter Sensitivity Analysis

ASNMF-SRP has three hyperparameters: the higher-order graph regularization coefficient α , the balance parameter for feature relationship preservation β , and the regularization parameter for sparse constraint λ . In the parameter sensitivity analysis experiment, the value ranges for α and β were set to $\{10^0, 10^1, 10^2, 10^3\}$, and the range for λ was $\{10^{-4}, 10^{-3}\}$. The parameter sensitivity concerning α , β , and λ is shown in Figure 9.

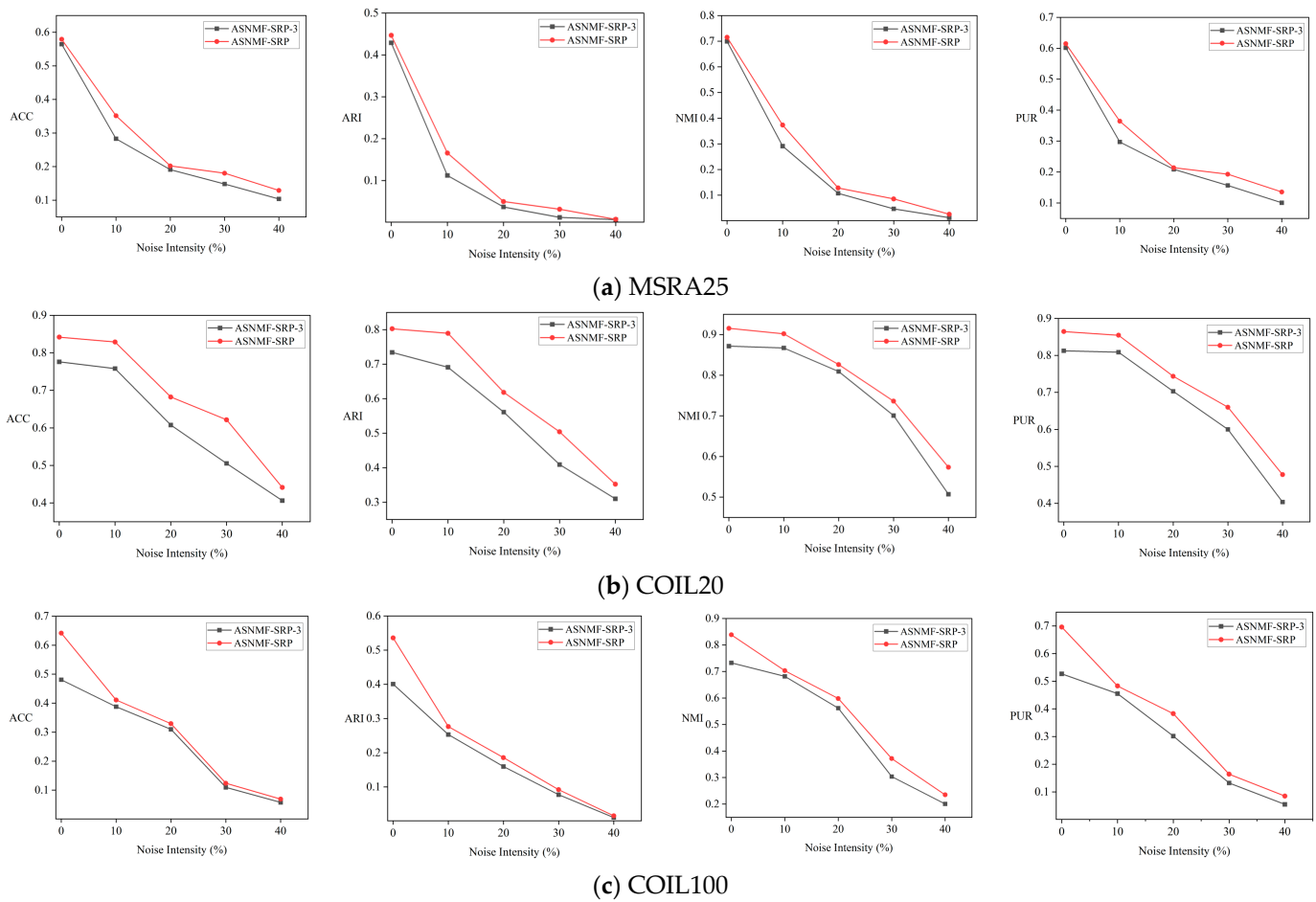


Figure 8. Comparison of clustering results between ASNMF-SRP and ASNMF-SRP-3 under different noise intensities.

From Figure 9, we can obtain the following:

(1) On the MSRA25, COIL20, COIL100, and Hitech datasets, ASNMF-SRP demonstrates favorable clustering performance when the higher-order regularization parameter $\alpha = 1000$; on the Semeion and Vehicle datasets, ASNMF-SRP exhibits good clustering performance when parameter α takes smaller values.

(2) With parameters α and β determined, different values of the sparse constraint regularization parameter λ significantly affect the clustering performance of the ASNMF-SRP algorithm. For example, on the COIL100 dataset, when $\lambda = 0.0001$, the clustering results of ASNMF-SRP overall outperform those when $\lambda = 0.001$.

(3) α and λ jointly constrain the coefficient matrix V , where α preserves the manifold structure of samples while λ balances the strength of sparsity, both optimizing the coefficient matrix through synergistic effects. Simultaneously, α and β collaboratively optimize the local geometric structure of data in both sample space and feature space. In parameter settings, the values of α and β are generally greater than that of λ to enhance manifold structure preservation capability, while the value of λ is kept relatively small to ensure appropriate sparse constraint intensity.

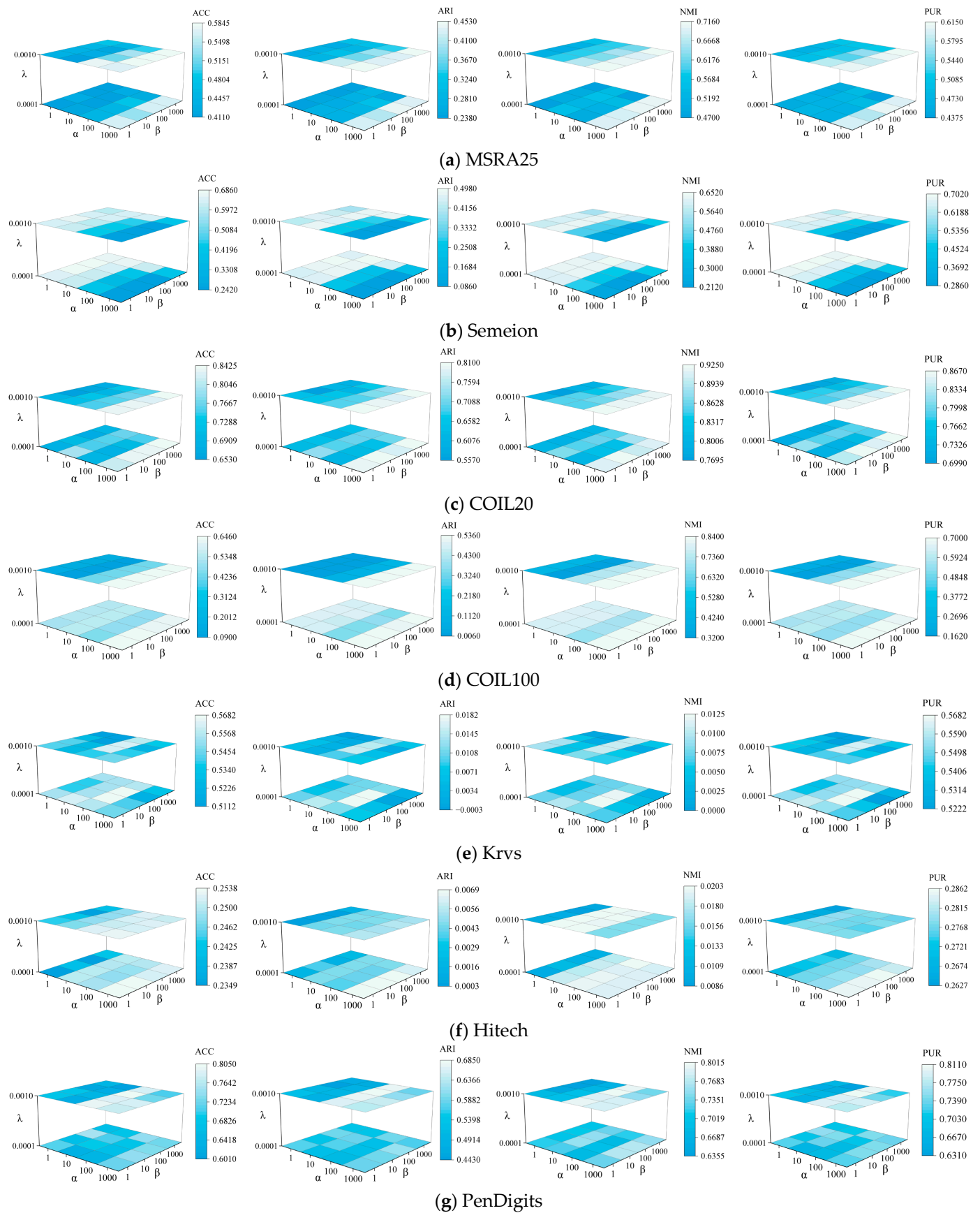


Figure 9. Cont.

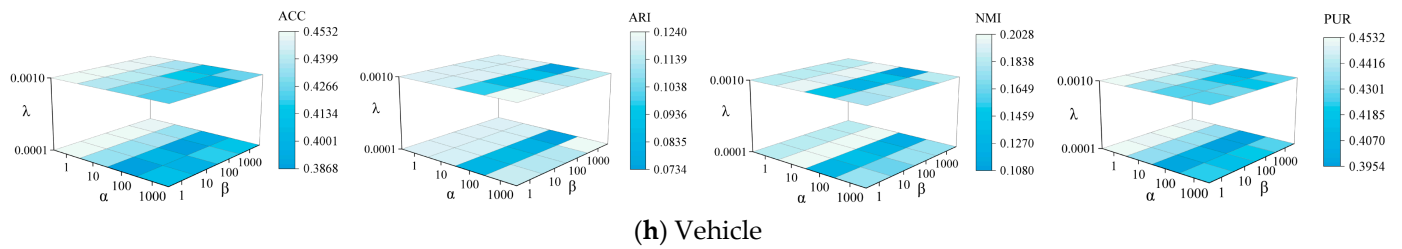


Figure 9. Parameter sensitivity analysis of ASNMF-SRP on various datasets.

4.9. Empirical Convergence

This section will experimentally analyze the convergence of the ASNMF-SRP algorithm on 8 datasets to further verify the accuracy of the theoretical analysis in Section 3.7. In the empirical convergence experiment, the values of hyperparameters α , β , and λ in ASNMF-SRP remain consistent with those in Table 3.

From Figure 10, it can be observed that the ASNMF-SRP algorithm achieves convergence on all eight datasets. Specifically, on the MSRA25, Semeion, COIL20, COIL100, Krvs, Hitech, and PenDigits datasets, the ASNMF-SRP algorithm exhibits a relatively fast convergence rate during the initial iterations, with its corresponding objective function value rapidly decreasing, ultimately achieving convergence within 20 iterations. On the Vehicle dataset, the ASNMF-SRP algorithm shows a rapid decline in the objective function value during the initial iterations, but subsequently, the rate of decrease slows down, ultimately achieving convergence around 40 iterations.

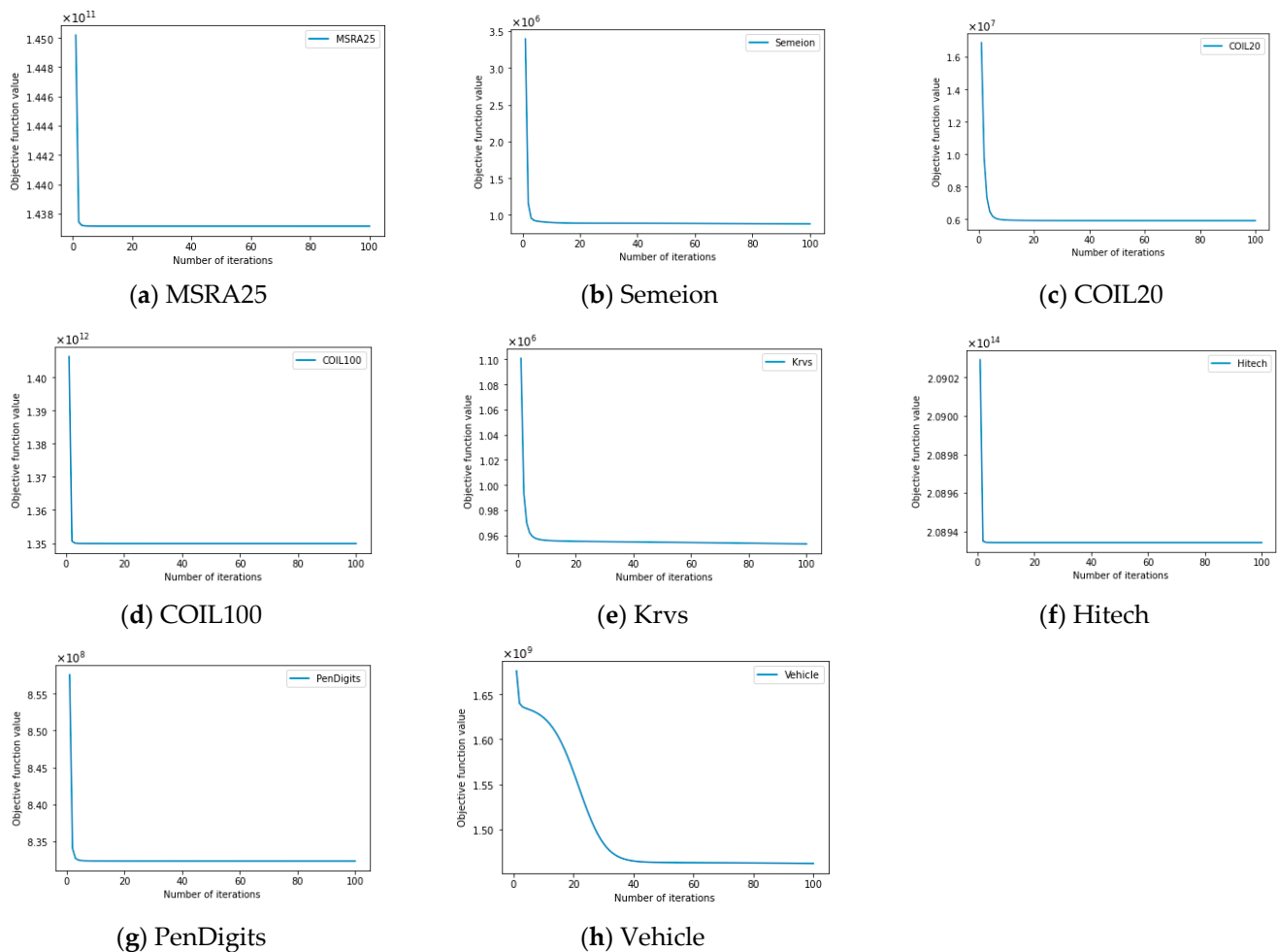


Figure 10. Convergence curves of ASNMF-SRP on various datasets.

5. Conclusions

This paper proposes an autoencoder-like sparse non-negative matrix factorization with structure relationship preservation (ASNMF-SRP). By integrating the principle of autoencoders, the algorithm enhances the optimization of the coefficient matrix through a linear “decoder-encoder” approach, which not only improves the factorization stability of the coefficient matrix but also further uncovers the latent representations of the original data. In the structural learning of the sample space, ASNMF-SRP effectively captures higher-order topological information between samples by constructing the optimal Laplacian matrix. In feature relationship learning, ASNMF-SRP employs the $l_{2,1}$ -norm to define the reconstruction error of feature correlations between the basis matrix and the original data matrix, ensuring consistency in feature relationships between the low-dimensional space and the original high-dimensional space. Furthermore, a sparse constraint based on inner product representation is imposed on the coefficient matrix, further enhancing the clustering performance of ASNMF-SRP. Finally, comparative experiments between ASNMF-SRP and nine other advanced clustering algorithms demonstrate, as evidenced by the data in Tables 4–7, that ASNMF-SRP achieves favorable clustering performance. In future research work, we will conduct in-depth exploration from the following three aspects: (1) To further optimize the construction process of high-order graph regularization, enhancing the model’s capability to represent complex structures; (2) to extend ASNMF-SRP into a deep NMF framework, enabling better exploration of latent features in data; and (3) to design a multi-view extension model for ASNMF-SRP, thereby broadening the applicability of the algorithm.

Author Contributions: Conceptualization, L.Z. and H.G.; methodology, L.Z.; software, L.Z.; validation, L.Z. and H.G.; formal analysis, L.Z. and H.G.; investigation, L.Z. and H.G.; resources, L.Z. and H.G.; data curation, L.Z.; writing—original draft preparation, L.Z.; writing—review and editing, L.Z. and H.G.; visualization, L.Z.; supervision, H.G.; project administration, L.Z. and H.G.; funding acquisition, H.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the National Social Science Fund of China (NO. 21BTJ042), the Gansu Provincial Natural Science Foundation (NO. 23JRR1186), and the Gansu Provincial Universities’ Young Doctor Support Program (NO. 2025QB-058).

Data Availability Statement: Publicly available datasets were analyzed in this study.

Acknowledgments: The authors sincerely appreciate the editors and reviewers for their valuable comments and professional suggestions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kim, S.; Shin, W.; Kim, H.W. Predicting online customer purchase: The integration of customer characteristics and browsing patterns. *Decis. Support Syst.* **2024**, *177*, 114105. [\[CrossRef\]](#)
2. Zeng, Y.; Chen, J.; Pan, Z.; Yu, W.; Yang, Y. Integrating single-cell multi-omics data through self-supervised clustering. *Appl. Soft Comput.* **2025**, *169*, 112541. [\[CrossRef\]](#)
3. Mardani, K.; Maghooli, K.; Farokhi, F. Segmentation of coronary arteries from X-ray angiographic images using density based spatial clustering of applications with noise (DBSCAN). *Biomed. Signal Process. Control* **2025**, *101*, 107175. [\[CrossRef\]](#)
4. Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52. [\[CrossRef\]](#)
5. Zhang, Q.; Wang, Y.; Levine, M.D.; Yuan, X.; Wang, L. Multisensor video fusion based on higher order singular value decomposition. *Inf. Fusion* **2015**, *24*, 54–71. [\[CrossRef\]](#)
6. Lee, D.D.; Seung, H.S. Learning the parts of objects by non-negative matrix factorization. *Nature* **1999**, *401*, 788–791. [\[CrossRef\]](#)
7. Hoyer, P.O. Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res.* **2004**, *5*, 1457–1469. [\[CrossRef\]](#)
8. Kong, D.; Ding, C.; Huang, H. Robust Nonnegative Matrix Factorization Using L_{21} -norm. In Proceedings of the 20th ACM International Conference on Information and Knowledge Management, Glasgow, UK, 24–28 October 2011; pp. 673–682. [\[CrossRef\]](#)

9. Ding, C.; Li, T.; Peng, W. 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; pp. 126–135. [\[CrossRef\]](#)
10. Belkin, M.; Niyogi, P.; Sindhiani, V. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. *J. Mach. Learn. Res.* **2006**, *7*, 2399–2434.
11. Cai, D.; He, X.; Han, J.; Huang, T.S. Graph Regularized Nonnegative Matrix Factorization for Data Representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 1548–1560. [\[CrossRef\]](#)
12. Wu, B.; Wang, E.; Zhu, Z.; Chen, W.; Xiao, P. Manifold NMF with L2,1 norm for clustering. *Neurocomputing* **2018**, *273*, 78–88. [\[CrossRef\]](#)
13. Li, X.; Cui, G.; Dong, Y. Graph Regularized Non-Negative Low-Rank Matrix Factorization for Image Clustering. *IEEE Trans. Cybern.* **2017**, *47*, 3840–3853. [\[CrossRef\]](#)
14. Liu, Z.; Zhu, F.; Xiong, H.; Chen, X.; Pelusi, D.; Vasilakos, A.V. Graph regularized discriminative nonnegative matrix factorization. *Eng. Appl. Artif. Intell.* **2025**, *139*, 109629. [\[CrossRef\]](#)
15. Huang, S.; Xu, Z.; Zhao, K.; Ren, Y. Regularized nonnegative matrix factorization with adaptive local structure learning. *Neurocomputing* **2020**, *382*, 196–209. [\[CrossRef\]](#)
16. Ren, X.; Yang, Y. Semi-supervised symmetric non-negative matrix factorization with graph quality improvement and constraints. *Appl. Intell.* **2025**, *55*, 397. [\[CrossRef\]](#)
17. Mohammadi, M.; Berahmand, K.; Azizi, S.; Sheikhpour, R.; Khosravi, H. Semi-Supervised Adaptive Symmetric Nonnegative Matrix Factorization for Multi-View Clustering. *IEEE Trans. Netw. Sci. Eng.* **2025**; early access. [\[CrossRef\]](#)
18. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. *Adv. Neural Inf. Process. Syst.* **2016**, *29*, 3844–3852. [\[CrossRef\]](#)
19. Wang, D.; Ren, F.; Zhuang, Y.; Liang, C. Robust high-order graph learning for incomplete multi-view clustering. *Expert Syst. Appl.* **2025**, *280*, 127580. [\[CrossRef\]](#)
20. Zhan, S.; Jiang, H.; Shen, D. Co-regularized optimal high-order graph embedding for multi-view clustering. *Pattern Recogn.* **2025**, *157*, 110892. [\[CrossRef\]](#)
21. Wang, Y.; Yao, H.; Zhao, S. Auto-encoder based dimensionality reduction. *Neurocomputing* **2016**, *184*, 232–242. [\[CrossRef\]](#)
22. Chen, Y.; Qu, G.; Zhao, J. Orthogonal graph regularized non-negative matrix factorization under sparse constraints for clustering. *Expert Syst. Appl.* **2024**, *249*, 123797. [\[CrossRef\]](#)
23. Meng, Y.; Shang, R.; Jiao, L.; Zhang, W.; Yang, S. Dual-graph regularized non-negative matrix factorization with sparse and orthogonal constraints. *Eng. Appl. Artif. Intell.* **2018**, *69*, 24–35. [\[CrossRef\]](#)
24. Peng, C.; Zhang, Y.; Chen, Y.; Kang, Z.; Chen, C.; Cheng, Q. Log-based sparse nonnegative matrix factorization for data representation. *Knowl.-Based Syst.* **2022**, *251*, 109127. [\[CrossRef\]](#)
25. Xiong, W.; Ma, Y.; Zhang, C.; Liu, S. Dual graph-regularized sparse robust adaptive non-negative matrix factorization. *Expert Syst. Appl.* **2025**, *281*, 127594. [\[CrossRef\]](#)
26. Shang, F.; Jiao, L.C.; Wang, F. Graph dual regularization non-negative matrix factorization for co-clustering. *Pattern Recognit.* **2012**, *45*, 2237–2250. [\[CrossRef\]](#)
27. Gu, Q.; Zhou, J. Co-clustering on manifolds. In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, 28 June–1 July 2009; pp. 359–368. [\[CrossRef\]](#)
28. Hedjam, R.; Abdesselam, A.; Melgani, F. NMF with feature relationship preservation penalty term for clustering problems. *Pattern Recognit.* **2021**, *112*, 107814. [\[CrossRef\]](#)
29. Salahian, N.; Tab, F.A.; Seyedi, S.A.; Chavoshinejad, J. Deep Autoencoder-like NMF with Contrastive Regularization and Feature Relationship Preservation. *Expert Syst. Appl.* **2023**, *214*, 119051. [\[CrossRef\]](#)
30. Lee, D.D.; Seung, H.S. Algorithms for non-negative matrix factorization. In Proceedings of the 14th International Conference on Neural Information Processing Systems, Denver, CO, USA, 1 January 2000; pp. 535–541.
31. Sun, B.J.; Shen, H.; Gao, J.; Ouyang, W.; Cheng, X. A non-negative symmetric encoder-decoder approach for community detection. In Proceedings of the 2017 ACM Conference on Information and Knowledge Management, New York, NY, USA, 6–10 November 2017; pp. 597–606. [\[CrossRef\]](#)
32. Li, T.; Zhang, R.; Yao, Y.; Liu, Y.; Ma, J.; Tang, J. Graph regularized autoencoding-inspired non-negative matrix factorization for link prediction in complex networks using clustering information and biased random walk. *J. Supercomput.* **2024**, *80*, 14433–14469. [\[CrossRef\]](#)
33. Zhang, H.; Kou, G.; Peng, Y.; Zhang, B. Role-aware random walk for network embedding. *Inf. Sci.* **2024**, *652*, 119765. [\[CrossRef\]](#)
34. Hosein, M.; Massoud, B.Z.; Christian, J. A Fast Approach for Overcomplete Sparse Decomposition Based on Smoothed L~(0) Norm. *IEEE Trans. Signal Process.* **2009**, *57*, 289–301. [\[CrossRef\]](#)
35. Lu, X.; Wu, H.; Yuan, Y.; Yan, P.; Li, X. Manifold Regularized Sparse NMF for Hyperspectral Unmixing. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 2815–2826. [\[CrossRef\]](#)

36. 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](#)]
37. Wang, J.; Wang, L.; Nie, F.; Li, X. A novel formulation of trace ratio linear discriminant analysis. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *33*, 5568–5578. [[CrossRef](#)]
38. Wang, Q.; He, X.; Jiang, X.; Li, X. Robust Bi-stochastic Graph Regularized Matrix Factorization for Data Clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 390–403. [[CrossRef](#)] [[PubMed](#)]
39. Chen, M.; Li, X. Entropy Minimizing Matrix Factorization. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 9209–9222. [[CrossRef](#)] [[PubMed](#)]
40. Huang, J.; Nie, F.; Huang, H.; Ding, C.H.Q. Robust Manifold Nonnegative Matrix Factorization. *ACM Trans. Knowl. Discov. Data* **2013**, *8*, 11. [[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.