Graph Regularized Semi-Nonnegative Matrix Factorization Under Sparse Constraints for Clustering

Caiping Wang[1], Guangwei Qu[2], Junjian Zhao[2], Yasong Chen[2]*

Abstract

Non-negative matrix factorization (NMF) is an effective local feature extraction algorithm with non-negative matrix constraints. In order to obtain a NMF-based algorithm with better clustering performance and stronger robustness, this paper propose a new non-negative matrix factorization method called Graph Regular- ized Semi-NMF under sparse constraints (Semi-GNMFSC). This model embeds a Laplacian regularization term on the basis of Semi-NMF, keeps the corre- lation information of high-dimensional space samples, and maps effectively to low-dimensional space, thus improving the learning ability of algorithm space and making full use of the inherent geometry of data distribution. Note that GNMF algorithm is deficient in robustness, that is, it is susceptible to problems such as noise. So, by adding l1 norm sparse constraint to the basis matrix and coefficient matrix of the model respectively, the sparsity of the low-dimensional representa- tion matrix can be improved, clearer data can be obtained to approximate the high-dimensional matrix, and problems such as the influence of noise introduced in data reconstruction and the reduction of data clustering performance can be solved, and the adjustment of data eigenvalues and sparse constraints in the matrix can be realized. More importantly, the iterative optimization scheme of Semi-GNMFSC is derived in this paper, and the convergence of the algorithm is proved theoretically. In addition, clustering experiments have been conducted on 5 different types and sizes of public image datasets, and compared with K-means, PCA, and other NMF variants to verify the superiority of the Semi-GNMFSC algorithm in the three clustering performance indicators of ACC, NMI, and Purity.

BACKGROUND

Image recognition is a prominent area of study within the realm of computer vision, with a plethora of algorithms proposed for the purpose of recognizing images within pristine datasets.[7,20,47] However, in numerous practical scenarios, images are inevitably impacted by occlu- sions, weather conditions, and environmental factors.[48,23] Moreover, images in reality are inherently noisy. This will have varying degrees of impact on subsequent processing tasks such as image clustering, segmentation, fea- ture extraction, and edge detection. Therefore, the selection of an appropriate method for image noise reduction to eliminate interference is a critical step in the field of image recognition.

The dimensionality of image data typically presents a challenge, impacting the accuracy of classification and the computational time required for related calculations on target images. So, it is imperative to reduce the dimensionality of image data. Note that the objective of dimensionality reduction technology is not only to eliminate redundant dimensions but also to preserve valuable dimensions, to achieve low-rank approximation for high-dimensional data.

[1]Department of Information, Zhangjiakou Cigarette Factory Co., Ltd, No. 9, North Diamond Road, Qiaodong District, Zhangjiakou, 075000, Hebei, China.
[2]*School of Mathematical Sciences, TianGong University, Binshui West Road,Xiqing District, Tianjin, 300387, Tianjin, China.
Address for Correspondence to: Yasong Chen2*2*School of Mathematical Sciences, TianGong University, Binshui West Road,Xiqing District, Tianjin, 300387, Tianjin, China. E-mail: chenyasong@tiangong.edu.cn
Financial Disclosures: None

Of course, many methods meet the above requirements. Next, we will list some classic dimensionality reduction methods. Prin- cipal component analysis (PCA) [22] stands as one of the most conventional techniques for dimensionality reduction, characterized by its succinct concept: to reduce the dataset's dimensionality while retaining maximal variability (i.e. statistical information). Linear discriminant analysis (LDA)[29,45] is a supervised dimensionality reduction technique, in contrast to PCA which is unsupervised, as it considers the category output for each sample in its data set. The fundamental concept of LDA can be succinctly summarized as "minimizing intra-class variance and maximizing inter-class variance post-projection". Local linear embedding (LLE) [38,44] is also a very important dimension reduction method. Since LLE maintains local features of samples during dimensionality reduction, it is widely used in image recognition, high-dimensional data visualization and other fields. Compared with traditional PCA, LDA (or other similar dimensionality reduction methods) focuses on sample variance and LLE focuses on maintaining local linear features of samples during dimensionality reduction. Besides, independent component analysis (ICA)[19,11] can reduce the dimensionality of samples with a non-Gaussian distribution. It is worth noting that the primary distinction between ICA and PCA lies in the fact that ICA aims to identify the direction of maximum independence, and in contrast, PCA seeks to identify the direction of maximum variance. These algorithms represent fundamentally different solution models, yet both are essentially linear processing methods.

The aforementioned dimensionality reduction methods are considered to be clas- sic in the field. However, it has been widely acknowledged that image data inherently possess nonnegative attributes. Therefore, it is natural to impose a constraint ensuring that the data obtained through dimensionality reduction remains nonnegative. In 1994, the method of Non-negative Matrix Factorization (NMF) was introduced by [36] although it was not yet referred to as NMF at that time. This approach involved utilizing non-negative linear combina- tions of variables to represent factors, ensuring the physical significance of the data being factorized. The weighted least squares algorithm was employed for model opti- mization in [36]. However, due to computational complexity, this form of NMF did not gain widespread adoption. It was not until 1999 that the concept of NMF was first introduced by [25] in Nature and applied to face image representation. Then, the principles of NMF have received attention in various fields.

Although the NMF algorithm has the natural characteristics of sparse repre- sentation, its sparsity is still not enough. Based on the NMF model, Hoyer et al. proposed Non-negative Sparse Coding (NSC) [17] and non-negative matrix factorization with sparse constraints Method (NMFSC) [17] making the results of NMF algorithm further sparse, and then improving the recognition rate of the algorithm. From then on, the research on sparse NMF algorithm has become more and more popular [34,13,30,31] Note that NMF is essentially an unsuper- vised method and cannot exploit label information. The authors in [28] proposed a new semi-supervised matrix factorization method, namely constrained nonnegative matrix factorization (CNMF). This method introduced label informa- tion as an additional constraint, which could be applied to a wide range of practical problems. Many studies have found that high-dimensional data is usually located in the nonlinear low-dimensional manifold space. Aiming at the advantages of manifold learning, many NMF methods based on manifold learning have been proposed, such as Graph regularized nonnegative matrix factorization (GNMF) [6] neigh-borhood preserving orthogonal projection nonnegative matrix factorization [28] robust graph reconstruction-based nonnegative matrix factorization [18], and sparse dual graph-regularized deep nonnegative matrix factorization [17] . There are still many research results based on GNMF, please refer to [26,27,46] for details.

Thus, we can summarize the following three advantages of NMF for use in this paper: (1) NMF is adept at processing large-scale data in matrix form, yielding decomposition results with clear physical significance; (2) The implementation of the NMF algorithm is straightforward and resource-efficient; (3) By integrating concepts such as graph theory, orthogonality, and sparsity, the

applicability and accuracy of NMF can be effectively enhanced, particularly in addressing clustering challenges posed by noisy images. However, due to NMF making the raw data, basis matrix, and coefficient matrix all non-negative, this to some extent limits its performance. So, the Semi-NMF model was proposed in [10]. The idea of Semi-NMF allows the (part of) factorization of the original data matrix to be negative (such as allowing the basis matrix to be negative or the coefficient matrix to be nonnegative), and expands the application scope of the NMF method. Moreover, most of the NMF- based algorithms are sensitive to noisy data [7,10,18,37] so selecting the best denoising method in noisy image clustering based on NMF models is necessary. A natural question is whether Semi-NMF can bring different results in noise processing?

Driven by the above ideas, this paper combines them to study "clustering research on graph regularized semi-nonnegative matrix factorization under sparse constraints" for regular or noisy image data. It is worth noting that, in this study, the concept of "part constitutes a whole" of NMF is visually demonstrated. For example, when considering a face image, it can be vectorized to form a nonnegative matrix and then undergo matrix factorization using NMF. Interestingly, the resulting base image does not display a complete head image but instead focuses on specific parts of the face such as eyes, mouth, eyebrows, etc. Additionally, we will conduct simulations on the image data incorporating various types of noise and implementing diverse denoising techniques. The objective is to determine the most effective denoising method for different types of noise through experimental analysis, thereby offering valuable insights for future denoising preprocessing procedures.

The main contributions of this paper are as follows:
(1) We propose a novel model, Semi-GNMFSC, based on NMF, which is par- ticularly well-suited for sampling data in low-dimensional manifolds embedded in high-dimensional spaces and for clustering image data with redundant information.
(2) Firstly, the Semi-GNMFSC model relaxes the nonnegative constraint on the basis matrix, thereby enhancing the algorithm's applicability. Secondly, integrating a graph regularization term with sparsity constraint, the geometric structure of both data and feature manifolds is preserved, ensuring sparsity of the factor matrix and yielding improved local features for significantly enhanced clustering performance.
(3) By adding sparsity constraints to the coefficient matrix or basis matrix, we propose two variations of the semi-GNMFSC model, namely Semi-GNMFSCU and Semi-GNMFSCV, and evaluate their clustering performance through experimental results. Our findings indicate that adding sparsity constraint to the basis matrix yields optimal clustering performance and robustness for the model, thus providing valuable insights for future research in this area.

The rest of this paper is structured as follows. Section 2 briefly reviews the basic algorithms of NMF and describes variant theories of the NMF algorithms used in this paper. Section 3 provides a detailed introduction to two important new models of Semi-GNMFSC and their corresponding update rules. More importantly, the con- vergence proofs of two new algorithms' update rules are also given. Building on the preparatory work of the previous three sections, Section 4 presents many convincing numerical experiments. Finally, Section 5 makes a brief summary and outlook of this paper.

Auxiliary Work

In this section, firstly, we provide definitions or explanations of some commonly used symbols, and secondly, we also provide a brief review of NMF, Semi-NMF, and GNMF. Given n images and vectorized representation of each image, these images can be represented by a matrix $Y = [y_1, y_2, \ldots, y_N] \in R_+^{M\times N}$, and each column of Y represents a sample vector. Here and in all subsequent representations, $R^{M\times N}$ represents the set of all matrices of $M \times N$ whose elements are real, and $R_+^{M\times N}$ represents the set of all matrices of $M \times N$ whose elements are real and non-negative. Let $U = (u_1, \ldots, u_k) \in R_+^{M\times K}$ and $V = (v_1, \ldots, v_k) \in R_+^{N\times K}$ denote the basis matrix and coefficient matrix obtained by factorization of matrix Y , respectively.

The symbol $\|\cdot\|_1$ is the representation of the norm of l_1, which is the sum of the absolute values of each element in the target matrix. Specifically, the norm of l_1 of the matrix $A \in R^{M \times N}$ is defined as:

$$\|A\|_1 = \sum_{i=1}^{M} \sum_{j=1}^{N} |a_{ij}|,$$

the symbol $\|\cdot\|_F$ is the representation of the Frobenius norm, which can be viewed as a generalization of the l_2 norm of vectors, that is, the square root of all elements:

$$\|A\|_F = \sqrt{Tr(AA^T)} = \sqrt{\sum_{i,j} a_{ij}^2},$$

Tr (·) is the symbolic representation of the trace of a matrix, and the trace of a square matrix $A \in R^{n \times n}$ is defined as the sum of diagonal elements, i.e.,

$$Tr(A) = \sum_{i=1}^{n} a_{ii} = a_{11} + a_{22} + \cdots + a_{nn}.$$

This paper uses the following properties of Tr: (1) Tr (A + B) = Tr(A) + Tr(B); (2) Tr(rA) = r Tr(A), where r is a scalar; (3) Tr (AB) = Tr (BA), where A and B have appropriate expressions; (4) Tr (A^T) = Tr(A).

The symbol W is a symbolic representation of the weight matrix, and W_{ij} is used to measure the proximity of two points y_i and y_j. proposed three definition methods: (1) 0-1 Weighting, (2) Heat Kernel Weighting, (3) Dot-Product Weighting [7]. For simplicity, 0-1 Weighting is used to define the weight matrix W in this paper. The symbol D is the symbolic representation of the diagonal matrix whose entries are the sum of column elements in symmetric matrix W. That is, $D_{jj} = \sum_l W_{jl}$, written in matrix form:

$$D = \begin{pmatrix} D_{11} & & & \\ & D_{22} & & \\ & & \ddots & \\ & & & D_{NN} \end{pmatrix}.$$

The symbol L is the symbolic representation of the graph Laplacian matrix with $L = D - W$, which is the degree matrix minus the adjacency matrix. λ is the regularization parameter that controls the smoothness of the new representation. α is the parameter used to control the effect of sparse constraints. $\mathcal{I}$ is an all-1 matrix.

NMF

The objective of the NMF algorithm is to identify two nonnegative matrices, U and V, such that their product closely approximates the original matrix Y, i.e., $Y \approx U V^T$. The objective function of NMF can be expressed as follows:

$$\min_{U,V} \|Y - UV^T\|_F^2 . \quad (1)$$
$$\text{s.t. } U \geq 0, V \geq 0$$

The decomposition can be interpreted as follows: the column vectors of the original data matrix Y are expressed as weighted combinations of all the column vectors in the factor matrix $U=[u_{ik}] \in R_+^{M \times K}$ with the weighting coefficients being the elements of the corresponding column vectors in factor matrix $V^T=[v_{kj}] \in R_+^{K \times N}$. Therefore, U is referred to as a basis matrix, and V is referred to as a coefficient matrix. When $K < N$, utilizing the coefficient matrix instead of the original matrix can effectively achieve dimensionality reduction and compression of large-scale raw data. This approach not only minimizes storage space but also reduces computational costs, making it a valuable technique for handling high-dimensional datasets. To solve (1), the update rule proposed by [25] is shown below:

$$u_{ik} \leftarrow u_{ik} \frac{(YV)_{ik}}{(UV^TV)_{ik}},$$
$$v_{jk} \leftarrow v_{jk} \frac{(Y^TU)_{jk}}{(VU^TU)_{jk}}.$$

Semi-NMF

It is well-established that the data Y, U, and V in NMF must adhere to non-negativity. However, various practical applications, such as sensor generated data, do not necessarily conform to this constraint. Semi-NMF extends the applicability of NMF by relaxing the nonnegative constraints on the data. Unlike the classic NMF, Semi-NMF only imposes nonnegative constraints on the original coefficient matrix V, potentially extending the applicability of Semi-NMF to a wider range of problems (see [10]) for details). Simultaneously, the effectiveness of the concept of "semi-nonnegative"

has been demonstrated through experiments by [38]. As a result, the objective function based on Semi-NMF now takes the following form:

$$\min_{U,V} \|Y\text{-}UV^T\|_F^2 . \quad (2)$$
$$\text{s.t. } V \geqslant 0$$

To solve (2), the update rule proposed by [11] is shown below:

$$U \leftarrow Y V (V^T V) - 1,$$
$$v_{jk} \leftarrow v_{jk} \sqrt{\frac{(Y^T U)^+_{jk} + [V(U^T U)^-]_{jk}}{(Y^T U)^-_{jk} + [V(U^T U)^+]_{jk}}} .$$

Throughout this paper, the symbols A^+_{jk} *and* A^-_{jk} in a matrix $A = [A_{i,j}]$ are presented in the following way:

$$\begin{cases} A^+_{jk} = (|A_{jk}| + A_{jk})/2, \\ A^-_{jk} = (|A_{jk}| - A_{jk})/2, \end{cases}$$

in which case $A^+ = [A^+_{jk}]$, $A^- = [A^-_{jk}]$,*and* $A_{jk} = A^+_{jk} - A^-_{jk}$.

It is under the guidance of this positive result of Semi-NMF that we will incorporate the idea of "semi-nonnegative" into our data processing.

GNMF

The authors in [7] have incorporated spectral graph theory [7,1,40] and manifold learning theory [3,2,4] into the NMF algorithm, resulting in the proposal of the GNMF algorithm. GNMF is required to maintain the geometric. structure of the samples in the low-dimensional space while performing matrix factorization. Assume that the two data points y_i and y_j are adjacent points in the original space, then under low-dimensional space, the corresponding z_i and z_j are also neighboring points (let $za = [v_{a1}, \ldots, v_{ak}]^T$ be the low-dimensional representation of y_a). In general, we use Euclidean distance to measure the "dissimilarity" between the low-dimensional representation of two data points with z_i and z_j , i.e., $d(z_i, z_j) = \|z_i - z_j\|^2$. It is easy to see that the difference between z_i and z_j also depends on the W matrix in the beginning of this section. With the help of weight matrix W , one can use the following formula to measure the smoothness of the low-dimensional representation:

$$R1 = \frac{1}{2}\sum_{i,j=1}^{N} \| z_i - z_j \|^2 W_{ij} = \sum_{i=1}^{N} Z_i^T Z_i D_{ii} - \sum_{i,j=1}^{N} Z_i^T Z_j W_{ij}$$
$$= Tr(V^T D V) - Tr(V^T W V) = Tr(V^T L V) .$$

By minimizing R_1, if the similarity between data points y_i and y_j is high, their corresponding low dimensional space values z_i and z_j are also the same. This geometry-based regularization function is integrated with the original NMF objective function to obtain GNMF, which is defined by the following objective function:

$$\min_{U,V} \|Y - UV^T\|_F^2 + \lambda Tr(V^T L V). \quad (3)$$
$$s.t.\ U \geq 0,\ V \geq 0$$

To solve Eq.(3), the update rules proposed by [6] is shown below:

$$u_{ik} \leftarrow u_{ik} \frac{(YV)_{ik}}{(U V^T V)_{ik}},$$
$$v_{jk} \leftarrow v_{jk} \frac{(Y^T U + \lambda W V)_{jk}}{(V U^T U + \lambda D V)_{jk}} .$$

Semi-GNMFSC

In this section, based on the auxiliary work in the previous section, integrating semi non-negative, graph regularization, and sparse constraints together, a novel NMF based method called Semi-GNMFSC model will be presented. We will first provide a detailed introduction to the specific form of our improved model, followed by corresponding multiplication update rules, and finally provide the convergence theorem and its proof.

Objective function

Due to the strong performance of GNMF and Semi-NMF, we have integrated these two approaches. Initially, we introduce the concept of Semi-NMF into the original NMF framework. This relaxation of nonnegative constraints on the original matrix and basis matrix expands the applicability of NMF to a wider range of scenarios. For instance, in this study, although the image dataset is non-negative, negative values are introduced during the denoising process of noisy

data using wavelet transform. This results in the original matrix not satisfying the constraint of NMF, making NMF meaningless under interpretability. However, we are only committed to reducing dimensionality through the concept of NMF. As for whether the factorized basis matrix is non-negative, it is not important compared to the non-negative coefficient matrix that we need more. In such cases, Semi-NMF can address these issues. Additionally, in order to effectively handle data sampled from a submanifold embedded in a high-dimensional ambient space, we incorporate the intrinsic geometric structure of data distribution into the objective function as an additional regularization term based on Semi-NMF. Then the following formula is obtained:

$$\min_{U,V} \|Y - UV^T\|_F^2 + \lambda Tr(V^T L V). \quad (4)$$
$$\text{s.t. } V \geq 0$$

Note that in [16] they combined the NMF algorithm with sparse coding to establish Non-Negative Sparse Coding (NNSC) and improved the corresponding application performance. So, after integrating the concepts of GNMF, Semi-NMF algorithms, and the sparsity of matrices ignored in previous research, we will introduce the concept of sparsity into model (4) to strengthen sparsity constraints on the basis matrix U or coefficient matrix V, in order to obtain a decomposed matrix U or V that is sparse and achieve better clustering performance. So, we will obtain the following two new models.

(1) Apply sparse constraints to the coefficient matrix V.

$$\mathcal{J}Semi{-}GNMFSC_V = \min_{U,V} \|Y - UV^T\|_F^2 + \lambda Tr(V^T L V) + \alpha \|V\|_1. \quad (5)$$
$$s.t.\ V \geq 0$$

(2) Apply sparse constraints to the basic matrix U.

$$\mathcal{J}Semi{-}GNMSC_U = \min_{U,V} \|Y - UV^T\|_F^2 + \lambda Tr(V^T L V) + \alpha \|U\|_1. \quad (6)$$
$$s.t.\ V \geq 0$$

The above two new models are collectively referred to as Semi-GNMFSC model.

Update Rules

In this subsection, we will present the solution to model (5) and model (6). As we have seen, (5) and (6) are not jointly convex for U and V, so we cannot have a closed- form solution. The above minimization problem can be solved by using the iterative algorithm updated alternately by U and V.

Efficient algorithm for solving model (5)

The model (5) can be re-expressed in the following sense:

$$\mathcal{J}Semi{-}GNMFSC_V = \mathrm{Tr}((Y - U V^T)(Y - U V^T)^T) + \lambda \mathrm{Tr}(V^T L V) + \alpha \|V\|_1$$
$$= Tr(Y Y^T) - 2Tr(Y^T U V^T) + Tr(U V^T V U^T) + \lambda\, Tr(V^T L V) + \alpha \|V\|_1, \quad (7)$$

and the optimization of U is equivalent to optimization of the following functions:

$$\mathcal{O}_1 = -2\, Tr(Y^T U V^T) + Tr(V U^T U V^T).$$

Since the matrix U does not have any restriction, it is straightforward to take the partial derivative of $\mathcal{O}_1$ concerning U and set it to zero, i.e.,

$$\frac{\partial \mathcal{O}_1}{\partial U} = -2 Y V + 2 U V^T V = 0,$$

and then the following update rule of U is arrived:

$$U \leftarrow Y V (V^T V)^{-1}. \quad (8)$$

Therefore, the update rule and convergence proof of U in this paper is consistent with that in Semi-NMF [11] The difference is the update rule and convergence proof of coefficient matrix V.

Next, we give the updating rule for V. The optimization of V is equivalent to the optimization of the following functions:

$$\mathcal{O}_2 = -2\, Tr(Y^T U V^T) + Tr(U V^T V U^T) + \lambda\, Tr(V^T L V) + \alpha \|V\|_1.$$

Let ϕ_{jk} be the Lagrangian multiplier that constrains $V \geq 0$ and $\Phi = [\phi_{jk}]$, so that one can obtain the following Lagrangian function:

$$\mathcal{L}_1 = -2\, Tr(Y^T U V^T) + Tr(U V^T V U^T) + \lambda Tr(V^T L V) + \alpha \|V\|_1 - Tr(\Phi V^T).$$

The first partial derivatives of $\mathcal{L}_1$ with respect to V will lead to

$$\frac{\partial L_1}{\partial V} = -2 Y^T U + 2 V U^T U + 2\lambda L V + \alpha \mathcal{I}_1 - \mu,$$

and with the help of KKT condition [43] $\phi_{jk} v_{jk} = 0$,

$$(-2Y^T U + 2V U^T U + 2\lambda LV + \alpha I_1)_{jk} V_{jk} = 0 \quad (9)$$

follows. According to this method, it can be derived that:

$$\begin{cases} [Y^T U]_{jk} = (Y^T U)^+_{jk} - (U^T U)^-_{jk}, \\ [VU^Y U]_{jk} = [V(U^T U)]^+_{jk} - [V(U^T U)]^-_{jk}. \end{cases}$$

We can obtain the following update rule:

$$v_{jk} \leftarrow v_{jk} \sqrt{\frac{2(Y^T U)^+_{jk} + 2[V(U^T U)]^-_{jk} + 2\lambda(WV)_{jk}}{2(Y^T U)^+_{jk} + 2[V(U^T U)]^+_{jk} + 2\lambda(DV)_{jk} + \alpha}} \quad (10)$$

Therefore, (10) reduces to

$$(-2Y^T U + 2V U^T U + 2\lambda LV + \alpha I_1)_{jk} V^2_{jk} = 0, \quad (11)$$

and (11) is identical to (9). In fact, both of (11) and (9) require that at least one of the two factors is equal to zero. The first factors in both equations are the same, and for the second factors V_{jk} and V^2_{jk} *if* $Vjk = 0$, *then* $V^2_{jk} = 0$, and vice versa. Thus, if (11) holds, (9) also holds and vice versa. So, it is true that (11) is identical to (9). **Theorem 1.** The objective function *JSemi−GNMFSCV* in Eq.(5) is nonincreasing under the updating rules in Eq.(8) and Eq.(10).The Euclidean distance is invariant under these updates if and only if U and V are at a stationary point of the distance.

Theorem 1 guarantees the convergence of the iterations in Eq.(8) and Eq.(10), so the final solution will be a local optimum. Our proof basically follows the idea of Semi-NMF in Ding et al (2008) and will be given in section 3.4.

Efficient algorithm for solving model (6)

The model (6) can be re-expressed in the following sense:

$$JSemi{-}GNMFSC_U = Tr[(Y-UV^T)(Y-UV^T)^T] + \lambda\, Tr(V^T LV) + \alpha \| U \|_1 = Tr(YY^T) - 2Tr(Y^T U V^T) + Tr(U V^T VU^T) + \lambda\, Tr(V^T LV) + \alpha \| U \|_1. \quad (12)$$

Note that the matrix properties $Tr(X) = Tr(X^T)$, $Tr(XY) = Tr(YX)$ for suitable X and Y. *Let* ψ and ϕ be the Lagrange multipliers satisfying $\Phi = [\phi_{jk}]$, then the Lagrange function $\mathcal{L}_2$ of (12) is as follows:

$$\mathcal{L}_2 = \mathcal{J}\, Semi{-}GNMFSC_U - Tr(\Psi U^T) - Tr(\Phi V^T). \quad (13)$$

Taking the partial derivative of the above function $\mathcal{L}_2$ with respect to U and V will yield the following two expressions.

$$\frac{\partial L_2}{\partial U} = -2YV + 2UV^T V + \alpha \mathcal{A} - \Psi, \quad (14)$$

$$\frac{\partial L_2}{\partial V} = -2Y^T U + 2VU^T U + 2\lambda LV - \Phi. \quad (15)$$

Similarly, $UV^T V$ in (14) can be represented as:

$$[UV^T V]_{ik} = [U(V^T V)]^+_{ik} - [U(V^T V)]^-_{ik},$$

as well as $VU^T U$ in (15) can be represented as:

$$[VU^T U]_{ik} = [VU^T U]^+_{ik} - [VU^T U]^-_{ik},$$

by using the KKT condition $\psi_{ik} U_{ik} = 0$ and $\phi_{jk} V_{jk} = 0$, we can obtain the following update rules:

$$u_{ik} \leftarrow u_{ik} \sqrt{\frac{2[YV]_{ik} + 2[U(V^T V)]^-_{ik}}{2[U(V^T V)]^+_{ik} + \alpha}}, \quad (16)$$

$$vjk \leftarrow vjk \sqrt{\frac{(Y^T U)^+_{jk} + [V(U^T U)]^-_{jk} + \lambda(WV)_{jk}}{(Y^T U)^-_{jk} + [V(U^T U)]^+_{jk} + \lambda(DV)_{jk}}}. \quad (17)$$

Theorem 2. The objective function $\mathcal{J}$ Semi−GNMF SC_U in Eq.(6) is nonincreasing under the updating rules in Eq.(16) and Eq.(17).The Euclidean distance is invariant under these updates if and only if U and V are at a stationary point of the distance.

Theorem 2 guarantees the convergence of the iterations in Eq.(16) and Eq.(17), and the proof will be given in section 3.4.

Analysis of complexity

After deducing the multiplication update rules of *JSemi−GNMFSCV* and *JSemi−GNMFSCU* , we will form the following program process. For simplicity, we on or *JSemi−GNMFSCV* , while the program process for *JSemi−GNMFSCU* is similar.

Algorithm 1 *Semi-GNMFSC$_V$* algorithm description
Input: Initial data matrix $X = [x_1, \ldots, x_N] \in \boldsymbol{R}_{\pm}^{\boldsymbol{M}\times\boldsymbol{N}}$, Graph regularization parameter λ, Sparsity parameter α
Output: Locally optimal solution matrix U, Corresponding coefficient matrix V
1: Initialization: The initial matrix is randomly selected $U \in \boldsymbol{R}_{+}^{\boldsymbol{M}\times\boldsymbol{K}}$,V+ ∈ R$^{N\times K}$;
2: The initial graph matrix W is constructed from k nearest neighbours, D=D$_{jj}$, L=D− W;
3: Fix U and update V according to the formula (8);
4: Fix V and update U according to the formula (10);
5: If it is less than the threshold or exceeds the given number of iterations, the algorithm terminates. Otherwise, 3;

Meanwhile through Algorithm 1, we will have the computational complexity of the proposed algorithm. The steps that affect the complexity of Algorithm 1 mainly consist of step 2, step 3, and step 4. Step 2 is the computation of weight matrix for constructing the data graph, and its complexity is $O(N^2M)$, in step 3, the complexity of calculating U with one iteration is $O(MNK + NK^2)$, and in step 4, the complexity of calculating V with one iteration is $O(MNK + NK^2 + KM^2)$. In summary, the complexity of *Semi-GNMFSC$_V$* algorithm is $O[t(MNK + N^2M + NK^2 + KM^2)]$. Algorithm of *Semi-GNMFSC$_U$* has the same complexity.

Convergence proof of Semi-GNMFSC

In this subsection, we will prove the convergence of (5) and (6). We first introduce the following definition.

Definition 1.[24] $Z(H, H')$ is an auxiliary function for $J(H)$ if the conditions

$$Z(H, H') \geq J(H), \quad Z(H, H) = J(H),$$

are satisfied.

The auxiliary function is very useful because of the following lemma. Lemma 1. If $Z(H, H^{(t)})$ is an auxiliary function , then $J(H)$ is nonincreasing under the update

$$H^{(t+1)} = \underset{H}{arg\,min} Z(H, H^{(t)}). \qquad (18)$$

Proof. $J(H^{(t+1)}) \leq Z(H^{(t+1)}, H^{(t)}) \leq Z(H^{(t)}, H^{(t)}) = J(H^{(t)})$

Note that $J(H^{(t+1)}) = J(H^{(t)})$ only if $H^{(t)}$ is a local minimum of $Z(H, H^{(t)})$. If the derivatives of J exist and are continuous in a small neighborhood of $H^{(t)}$ this also implies that the derivatives $\nabla J(H^{(t)}) = 0$. Thus, by iterating the update in (18) we obtain a sequence of estimates that converge to a local minimum $H_{min} = arg\,min_H J(H)$ of the objective function:

$$J(H_{min}) \leq \ldots J(H^{t+1}) \leq J(H^t) \ldots \leq J(H_2) \leq J(H_1) \leq J(H_0).$$

According to Lemma 2 (see below), $Z(H, H')$ defined in (22) is an auxiliary function of J and its minimum is given by (23). Accordingto (18), $H^{(t+1)} \leftarrow H$ *and* $H^{(t)} \leftarrow H'$.

Lemma 2. For any matrices $A \in R_+^{n\times n}$, $B \in R_+^{k\times k}$, $S \in R_+^{n\times k}, S' \in R_+^{n\times k}$, with A and B symmetric, the following inequality holds:

$$\sum_{i=1}^{n}\sum_{p=1}^{K}\frac{(AS'B)_{ip}S_{ip}^2}{S'_{ip}} \geq T_r(S^TASB). \qquad (19)$$

Proof. Let $S_{ip} = S'_{ip}u_{ip}$. Using an explicit index, the difference Δ between the left-hand side and the right-hand side can be written as:

$$\Delta = \sum_{i,j=1}^{n}\sum_{p,q=1}^{k} A_{ij}\, S'_{jq}B_{qp}S'_{ip}\,(u_{ip}^2 - u_{ip}u_{jq}),$$

because A and B are symmetric, this is equal to:

$$\Delta = \sum_{i,j=1}^{n}\sum_{p,q=1}^{k} A_{ij}\, S'_{jq}B_{qp}S'_{ip}\left(\frac{u_{ip+}^2 + u_{jq}^2}{2} - u_{ip}u_{jq}\right)$$

$$= \frac{1}{2}\sum_{i,j=1}^{n}\sum_{p,q=1}^{k} A_{ij}\, S'_{jq}B_{qp}S'_{ip}(u_{ip-} - u_{jq})^2 \geq = 0.$$

When B = I and S is a column vector, (19) reduces to the result in [23].

Convergence proof of Model (5)

To prove Theorem 1, we need to show that (5) does not increase under the update step in (8) and (10). Since the regular and sparse terms in (5) as well as the non-negative constraint terms are only related to V, our updated formula for U in Semi-GNMFSCv is exactly the same as that in Semi-NMF. Therefore, we can use the convergence proof of Semi-NMF to show that (5) is not increased under the update step in (8). See [9] for details. Now, we just need to prove that (5) is not incremented under the update step in (10). Meanwhile, in order to facilitate comparison with the process in [9], we rewrite JSemi−GN M F SCV as follows:

$$J(H) = Tr(-2H^T B + HAH^T + \lambda H^T DH - \lambda H^T W H) + \alpha \|H\|_1, \quad (20)$$

where $A = U^T U$, $B = Y^T U$, and $H = V$. Finally, $J(H)$ can be written as follows:

$$J(H) = Tr(-2[H^T B]^+ + 2[H^T B]^- + [HA]^+ H^T - [HA]^- H^T + \lambda H^T DH - \lambda H^T W H) + \alpha \|H\|_1. \quad (21)$$

Lemma 3. Given the objective function $J(H)$defined in (21) with all matrices are nonnegative. Then the following function

$$Z(H,H') = -\sum_{ik} 2B_{ik}^+ H'_{ik}\left(1 + Log\frac{H_{ik}}{H'_{ik}}\right) + \sum_{ik} B_{ik}^- \frac{H_{ik}^2 + H'^2_{ik}}{H'_{ik}} + \sum_{ik} \frac{(H'A^+)_{ik} H_{ik}^2}{H'_{ik}} - \sum_{ikl} A_{kl}^- H'_{ik} H'_{i\ell}\left(1 + log\frac{H_{ik}H_{i\ell}}{H'_{ik}H_{i\ell}}\right) + \lambda\sum_{ik}\frac{(DH')_{ik}H_{ik}^2}{H'_{ik}} - \lambda\sum_{ikl} W_{li} H'_{ik} H'_{lk}\left(1 + log\frac{H_{ik}H_{lk}}{H'_{ik}H'_{lk}}\right) + \alpha\sum_{ik}\frac{H_{ik}^2 + H'^2_{ik}}{2H'_{ik}}, \quad (22)$$

is an auxiliary function for $J(H)$, that is, the auxiliary function $Z(H,H')$ satisfies Definition 1. Furthermore, it is a convex function in H and its global minimum is

$$H_{ik} = arg\ min_H\ Z(H,H') = H'_{ik}\sqrt{\frac{2B_{ik}^+ + 2(H'A^-)_{ik} + 2\lambda(WH')_{ik}}{2B_{ik}^- + 2(H'A^+)_{ik} + 2\lambda(DH')_{ik} + \alpha}}. \quad (23)$$

Proof. The function $J(H)$ is composed of positive and negative terms. To establish the validity of the auxiliary, it is imperative to determine the upper bound for the positive term and the lower bound for the negative term. We firstly establish an upper bound for the positive term. $J(H)$ consists of four positive terms, i.e., the second, third, fifth, and seventh terms in (22). The qualifications for the second and seventh items are proved by the following two formulas:

$$Tr(H^T B^-) = \sum_{ik} H_{ik} B_{ik}^- \le \sum_{ik} B_{ik}^- \frac{H_{ik}^2 + H'^2_{ik}}{2H'_{ik}}, \quad (24)$$

$$||H||_1 = \sum_{ik} H_{ik} \le \sum_{ik} \frac{H_{ik}^2 + H'^2_{ik}}{2H'_{ik}}. \quad (25)$$

The above two formulas use the inequality $a \le (a^2 + b^2)/2b$ for any $a > 0$, $b > 0$.

For the third and fifth terms in J(H) (the third item: $A = I$ and $B = A^+$, the fifth item: $B = I$ and $A = D$), by using Lemma 2, we obtain the upper bounds estimation:

$$Tr(HA^+H^T) \le \sum_{ik} \frac{(H'A^+)_{ik} H_{ik}^2}{H'_{ik}}, \quad (26)$$

$$Tr(H^T DH) \le \sum_{ik} \frac{(DH')_{ik} H_{ik}^2}{H'_{ik}}, \quad (27)$$

Until now, $J(H)$ remains three negative terms' lower bounds to estimate, which are the first, fourth and sixth terms. We will get the lower bounds by using the inequality $z \ge 1 + log\ z$ (for any $z > 0$). Then,

$$\frac{H_{ik}}{H'_{ik}} \ge 1 + log\ \frac{H_{ik}}{H'_{ik}}, \quad (28)$$

And

$$\frac{H_{ik}H_{i\ell}}{H'_{ik}H'_{i\ell}} \ge 1 + log\ \frac{H_{ik}H_{i\ell}}{H'_{ik}H'_{i\ell}}, \quad (29)$$

Via (28), the first term in $J(H)$ is estimated in the following way:

$$Tr(H^T B^+) = \sum_{ik} B_{ik}^+ H_{ik} \ge \sum_{ik} B_{ik}^+ H'_{ik}\left(1 = log\frac{H_{ik}}{H'_{ik}}\right). \quad (30)$$

Via (29), the fourth and sixth terms in $J(H)$ are estimated by

$$Tr(HA^-H^T) \ge \sum_{ik\ell} A_{k\ell}^- H'_{ik} H'_{i\ell}\left(1 + log\frac{H_{ik}H_{i\ell}}{H'_{ik}H'_{i\ell}}\right), \quad (31)$$

$$\lambda\, Tr(H^T WH) \geq \lambda \sum_{ik\ell} W_{\ell i} H'_{ik} H'_{\ell k} \left(1 + log \frac{H_{ik} H_{i\ell}}{H'_{ik} H'_{i\ell}}\right). \quad (32)$$

Putting (24), (25), (26), (27), (30), (31), (32) together, we get an auxiliary function $Z\,(H, H')$ such that $J(H) \leq Z\,(H, H')$ and $J(H) = Z(H, H)$.

To find the minimum of Z (H, H'), we take $\frac{\partial Z(H,H')}{\partial H_{ik}}$ *as follows*:

$$\frac{\partial Z(H,H')}{\partial H_{ik}} = -2B^+_{ik} \frac{H'_{ik}}{H_{ik}} + 2B^-_{ik} + \frac{H'_{ik}}{H_{ik}} \frac{2(H'A^+)_{ik} H_{ik}}{H_{ik}} - 2\frac{(H'A^-)_{ik} H'_{ik}}{H_{ik}} + 2\lambda \frac{(DH')_{ik} H_{ik}}{H'_{ik}} - 2\lambda \frac{(WH')_{ik} H'_{ik}}{H_{ik}} + \alpha \frac{H_{ik}}{H'_{ik}}. \quad (33)$$

$$Let\ J_1 = \sum_{ik\ell} A^-_{k\ell} H'_{ik} H'_{i\ell} \left(1 + log \frac{H_{ik} H_{i\ell}}{H'_{ik} H'_{i\ell}}\right) = \sum_{ts\ell} A^-_{s\ell} H'_{ts} H'_{t\ell} \left(1 + log \frac{H_{ts} H_{t\ell}}{H'_{ts} H'_{t\ell}}\right), \text{then}$$

$$\frac{\partial J_1}{\partial H_{ik}} = \left[\sum_\ell A^-_{kk} H'_{ik} H'_{i\ell} \left(1 + log \frac{H_{ik} H_{k\ell}}{H'_{ik} H'_{i\ell}}\right)\right]'_{H_{ik}} + \left[\sum_s A^-_{sk}\ H'_{is} H'_{ik} \left(1 + log \frac{H_{is} H_{ik}}{H'_{is} H'_{ik}}\right)\right]'_{ik} - \left[A^-_{kk}\ H'_{ik} H'_{ik} \left(1 + log \frac{H_{ik} H_{ik}}{H_{ik} H_{ik}}\right)\right]'_{H_{ik}} = A^-_{k\ell}\ H'_{ik} H'_{i\ell} \frac{1}{H_{ik}} + A^-_{sk}\ H'_{is} H'_{ik} \frac{1}{H_{ik}} = \frac{2\,(H'A^-)_{ik} H'_{ik}}{H_{ik}}.$$

$$\text{Let}\ \ J_2 = \sum_{ik\ell} W_{\ell i} H'_{ik} H'_{ik} \left(1 + log \frac{H_{ik} H_{\ell k}}{H'_{ik} H'_{ik}}\right) = \sum_{ts\ell} W^-_{\ell t} H'_{ts} H'_{\ell s} \left(1 + log \frac{H_{ts} H_{\ell s}}{H'_{ts} H'_{\ell s}}\right),$$

Then

$$\frac{\partial J_1}{\partial H_{ik}} = \left[\sum_\ell W^-_{\ell i} H'_{ik} H'_{\ell k} \left(1 + log \frac{H_{ik} H_{\ell k}}{H'_{ik} H'_{\ell k}}\right)\right]'_{H_{ik}} + \left[\sum_s W^-_{it}\ H'_{tk} H'_{ik} \left(1 + log \frac{H_{tk} H_{ik}}{H'_{tk} H'_{ik}}\right)\right]'_{H_{ik}} - \left[W^-_{ii}\ H'_{ik} H'_{ik} \left(1 + log \frac{H_{ik} H_{ik}}{H'_{ik} H'_{ik}}\right)\right]'_{H_{ik}} = W^-_{\ell i}\ H'_{ik} H'_{\ell k} \frac{1}{H_{ik}} + W^-_{it}\ H'_{tk} H'_{ik} \frac{1}{H_{ik}} = \frac{2\,(W^-H')_{ik} H'_{ik}}{H_{ik}}.$$

The Hessian matrix obtained by the second derivatives is

$$\frac{\partial^2 Z\,(H, H')}{\partial H_{ik} \partial H_{j\ell}} = \delta_{ij} \delta_{k\ell} Y_{ik}$$

With

$$Y_{ik} = \frac{2[(B^+)_{ik} + (H'A^-)_{ik} + \lambda (WH')_{ik}] H'_{ik}}{H^2_{ik}} + \frac{2B^-_{ik} + 2\,(H'A^+)_{ik} + 2\lambda (DH')_{ik} + \alpha}{H'_{ik}}.$$

Therefore, $Z\,(H, H')$ is a convex function of H. Thus, we obtain the global minimum of $Z\,(H, H')$ by

$$\partial Z\,(H, H')\,/\partial H_{ik} = 0,$$

in (33), and then (23) follows.

We can now prove the convergence of Theorem 1 by Lemma 1 and Lemma 3.

Proof of Theorem 1: Replacing $Z\,(H, H^{(t)})$ in (18) by (22) results in the following update rule:

$$H^{(t+1)} = H'_{ik} \sqrt{\frac{2B^+_{ik} + 2\,(H'A^-)_{ik} + 2\lambda\,(WH')_{ik}}{2B^-_{ik} + 2\,(H'A^+)_{ik} + 2\lambda\,(DH')_{ik} + \alpha}}.$$

Since (22) is an auxiliary function, J is nonincreasing under this update rule according to Lemma 1. Let $A = U^T U$, $B = Y^T F$, and $H = V$, we have (10).

Convergence proof of Model (6)

To prove Theorem 2, we need to prove that (6) does not increase under the updating steps of (16) and (17). Since (6) contains only $\alpha \| U \|_1$ *but* $\alpha \| V \|_1$ is missing. The update rule of V in (6) is similar to that in (5), just removing $\alpha \sum_{ik} \frac{H^2_{ik} H'^2_{ik}}{2H'_{ik}}$ term in auxiliary function, and others are the same, so we omit the details here. Now, the remaining is to show the update rule for U in (16) is exactly the update rule in (18) with a proper auxiliary function by the following Lemma.

Lemma 4. The function

$$Z\,(H, H') = \sum_{ik} H^-_{ik} \frac{B^2_{ik} + B'^2_{ik}}{2B'_{ik}} + \sum_{ik} \frac{H'_{ik})^+ A(H^2_{ik})^+}{(H'_{ik})^+} + \alpha \sum_{ik} \frac{H^2_{ik} + H'^2_{ik}}{2H'_{ik}} \sum_{ik} 2H^+_{ik} B'_{ik} \left(1 + log \frac{B_{ik}}{B'_{ik}}\right) - \sum_{ik\ell} A_{k\ell} (H'_{ik})^- (H'_{i\ell})^- \left(1 + log \frac{H^-_{ik} H^-_{i\ell}}{(H'_{ik})^- (H'_{i\ell})^-}\right) \quad (34)$$

is an auxiliary function for the objective function of $J(H)$ in (6):

$$J\,(H) = Tr[-2[B^T H]^+ + 2[B^T H]^- + [HA(H^T)]^+ - [HA(H^T)]^-] + \alpha \| H \|_1 \quad (35)$$

with $A = V^T V$, $B = Y V$, and $H = U$.

Proof.

$$T_r\,(B^T H^-) = \sum_{ik} H^-_{ik} B_{ik} \leq \sum_{ik} H^-_{ik} \frac{B^2_{ik} B'^2_{ik}}{2B'_{ik}},$$

$$T_r[H^+ A (H^T)^+] \leq \sum_{ik} \frac{(H'_{ik})^+ (AH^2_{ik})^+}{(H'_{ik})+},$$

$$||H||_1 = \sum_{ik} H_{ik} \leq \sum_{ik} \frac{H^2_{ik} H'^2_{ik}}{2H'_{ik}}.$$

The above are the positive terms in the auxiliary function, and then give the negative ones:

$$T_r\ (B^T\ H^+) = \sum_{ik} H_{ik}^+\ B_{ik} \geq \sum_{ik} H_{ik}^+ B_{ik}' \left(1 = log\ \frac{B_{ik}}{B_{ik}'}\right),$$

$$T_r[H^-A(H^T)^-\ ] \geq \sum_{ik\ell} A_{k\ell}\ H_{ik}')^-(H_{i\ell}')^-$$

$$\left(1 + log\ \frac{H_{ik}^- H_{i\ell}^-}{(H_{ik}')^-(H_{i\ell}')^-}\right).$$

From the above inequality we can get: *Z (H, H') ≥ J(H).* That is, the auxiliary function is valid.

The analysis of the rest of this section is the same as section 3.4.1.

EXPERIMENTS AND RESULTS ANALYSIS

This section will conduct experiments on image datasets with different noise levels based on the models and update algorithms outlined in Section 3 to verify the effectiveness of denoising methods for clustering.

Datasets

Five commonly utilized datasets will be employed in the experimental analysis, and a comprehensive depiction of these datasets used in the clustering task is provided in Table 1.

Table 1 : Statistics of the data sets

Databases	Samples (N)	Features (M)	Classes (C)
ORL	400	1024	40
COIL20	1440	1024	20
Yale	165	1024	15
PIE	2586	1024	68
AR	2600	1024	100

• COIL20 dataset [35] The dataset includes 20 types of images depicting various objects, each captured at 5-degree intervals in the horizontal direction, resulting in a total of 72 images capturing different angles within a 360-degree range. Consequently, the dataset consists of a total of 1440 grayscale images.

• Yale dataset [1] This dataset consists of 15 categories of images, each with 11 images, resulting in a total of 165 grayscale images depicting human facial features. Each person's portrait exhibits different characteristics, such as the presence or absence of glasses, different lighting directions (left, center, right), and emotional expression (joyful, neutral, melancholic).

• AR dataset [33] The dataset comprises 50 male and 50 female subjects, each with 26 images, resulting in a total of 2600 images.

• PIE dataset [41] The dataset comprises 2586 images captured by 68 individuals, and each of them is presented with 42 grayscale images featuring four distinct facial expressions and varying lighting conditions.

• ORL dataset [39] The dataset comprises 400 images featuring 40 distinct individuals, with 10 photographs captured for each individual. These images were obtained under varying conditions, including different lighting, facial expressions, and details.

Comparison algorithms

In order to fairly demonstrate the performance of the Semi-GNMFSC algorithm, we will compare it with K-means, PCA, and six others classic NMF algorithms. Here is an introduction to these classic algorithms being compared.

• K-means [42,21] The algorithm operates on the original matrix and does not perform operations such as extracting and utilizing the information contained in the original matrix .

• PCA [32] Principal component analysis, which is widely used in data dimensionality reduction, can extract the main components of the data set.

• NMF [24,25] The core purpose of this algorithm is to represent the original matrix as the product of two non-negative factor matrices, and the dimensions of the two factor matrices are much smaller than the

original matrix. The following algorithms (including those in this paper) are based on this idea.

• NMFSC [17,12] The algorithm aims to produce sparse representations and represent the data as a linear combination of a small number of basis vectors.

• ONMF(basic matrix orthogonal) [9] This algorithm adds orthogonal constraints to the basic matrix based on NMF.

• NeNMF [14] It applies Nesterov's optimal gradient method to alternatively optimize one factor with another.

• Semi-NMF [10] Unlike classical NMF, Semi-NMF only requires non-negative constraints on the coefficient matrix V.

• GNMF [6] This algorithm combines graph theory, manifold assumption, and NMF algorithm to construct neighborhood graphs that preserve the inherent geometric structure of the data.

The other implementation details of this paper are presented as follows:
(1). After factorizing the matrices, this paper will use the K-means algorithm to perform clustering analysis on the target dataset.
(2). For models with graph constraint in GNMF and Semi-GNMFSC, the 0-1 weighting scheme is adopted and the number of nearest neighbors is set to 20.
(3). For all comparison algorithms, experimental operations will be conducted based on the original papers or source code of these algorithms. At the same time, the parameters involved in the corresponding paper or code will also be set according to the values in the original text to obtain the best performance of the comparison algorithms.
(4). To reduce the randomness caused by initialization, repeat each algorithm 10 times and report the average clustering results of these 10 runs.

Evaluation indicators

In order to measure the clustering performance, we will use three commonly used clustering evaluation indicators: clustering accuracy (ACC), normalized mutual information (NMI) and purity (Purity). The process of calculating these three indicators is achieved by comparing the obtained labels with the real labels.
ACC is usually used to measure the percentage of correct labels obtained. Given a data set containing n images, for each sample image, let l_i be the clustering label obtained by applying some algorithm, and r_i be the label provided by the data set, then the ACC is defined by

$$AAC = \frac{\sum_{i=1}^{n} \delta\ (r_i, map(l_i))}{n},$$

where $\delta(x, y)$ is the classic delta function ($\delta(x, y)$ equals to *1 if* $x = y$ and equals to 0 otherwise). The function map (l_i) is the mapping function that maps each clustering label l_i to the equivalent label from the data set.
Mutual information is usually used to measure the similarity of two clusters. Suppose there are two clustering results C and C', the mutual information is defined by

$$M\ I\ (C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j).\ log\ \frac{p(c_i, c'_j)}{p(c_i).c'_j},$$

where $p\ (c_i)$ and $p\ (c'_j)$ represent the probability that the sample belongs to class c_i and class c'_j ,respectively. The probability $p\ (c_i,\ c'_j)$ represents the joint probability that the sample belongs to both class ci and class c'_j. So the definition of NMI is as follows:

$$N\ M\ I\ (C, C') = \frac{M\ I\ (C, C')}{max\ (H\ (C), H\ (C'))}.$$

The Purity indicator is generally used to measure the purity between the clustered labels and the true labels, that is, whether the data is classified into one category after clustering, with a high probability of being classified into the same category. The formulas for Purity is defined as follows:

$$Purity = \sum_{i=1}^{k} \frac{max_j\left(n_i^j\right)}{N},$$

Whare n_i^j denotes the number of j input classes that are assigned to the *i-th* class, and N is the total number of samples.
It is easy to see that the ranges of the above are

both within [0, 1], and the larger the value, the better the performance.

Parameter Selection

The performance demonstration of algorithms cannot be achieved without the help of good parameters. So, this subsection will use experiments to demonstrate the selection of parameters. The regularization parameter λ and the sparsity parameter α are two highly crucial parameters in the algorithm, whose values will directly affect the convergence speed and performance of the algorithm. Therefore, this paper will evaluate the influence of different parameter values on performance through a large number of experiments. We will test the effects of different parameter values on the performance of the proposed algorithm on five classical data sets. Meanwhile, consistent with tradition, α and λ will be within the scope of the following set: $\{10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$. The experimental results are summarized and the clustering performance results corresponding to different parameter values in the Semi-GNMFSCv algorithm are shown as follows.

From the Figure 1 to 5, it can be seen that the clustering performance of Semi-GNMFSC algorithm changes with the different values of α and λ parameters. According to the Figure 1, in the COIL20 data set, when the sparsity parameter $\alpha = 1$ and the regularization parameter $\lambda = 100$, the values of NMI, ACC and Purity are all the highest, which indicates that the performance of Semi-GNMFSCv algorithm is optimal under $\alpha = 1$ and $\lambda = 100$. Similarly (omitting details), observing the experimental results on the other four data sets, the performance of the algorithm is also optimal when $\alpha = 1$, $\lambda = 100$. Furthermore (omitting details), we will get the experimental results of the parameter selection for the Semi-GNMFSCu algorithm, where the selection results for

Figure 1 : The COIL20 dataset, Semi-GNMFSCv algorithm clustering performance and the relationship between the parameter value.

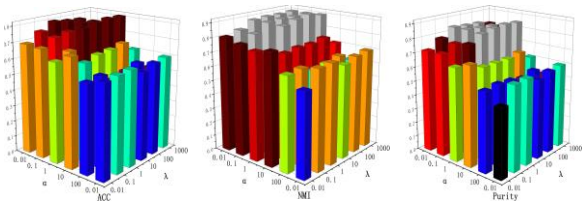


Figure 2 : The ORL dataset, Semi-GNMFSCv algorithm clustering performance and the relationship between the parameter value.

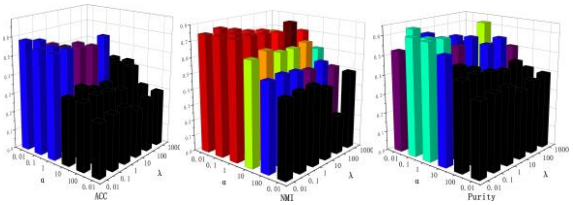


Figure 3 : The Yale dataset, Semi-GNMFSCv algorithm clustering performance and the relationship between the parameter value.

parameters α and λ are also $\alpha = 1$ and $\lambda = 100$. So, in the following experiments, we will set $\alpha = 1$, $\lambda = 100$.

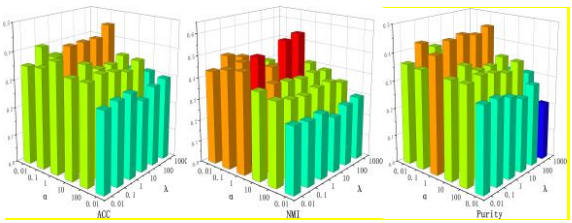


Figure 4 : The AR dataset, Semi-GNMFSCv algorithm clustering performance and the relationship between the parameter value.

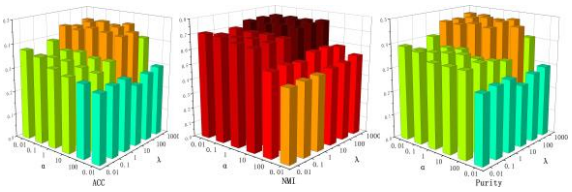


Figure 5 : The PIE dataset, Semi-GNMFSCv algorithm clustering performance and the relationship between the parameter value.

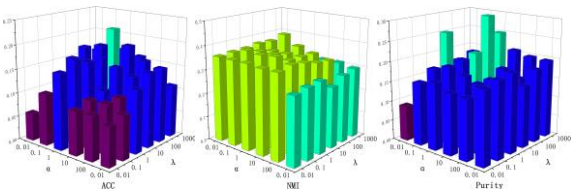
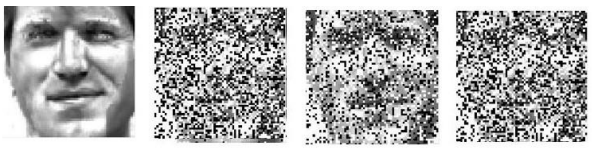


Figure 6 : Images with varying degrees of Salt and pepper noise



(a) Original image, (b) 20%, (c) 40%, (d) 60%

Figure 7 : Images with varying degrees of Gaussian noise

(a) Original image (b) 0.001 (c) 0.01 (d) 0.1

Using the two forms of noisy images mentioned above, we will conduct clustering experiments on COIL20 and Yale datasets and form the following four tables.

Discussion on clustering performance

In this subsection, we show the effectiveness of Semi-GNMFSC algorithm by comparing a large number of clustering experiments. In these experiments, the dimension K of the feature subspace was taken as 5, 10, 20, 30, 40 and 50, respectively.

Note that the decimal values in the two columns on the far right of all tables are the results of the algorithms proposed in this paper.

For the purpose of providing a clearer view of experimental results, the best clustering values are in bold for all datasets. Please refer to the detailed values provided in Table 2- Table 6.

Table 2 : Clustering results of different algorithms on COIL20 data

K	K-means	PCA	NMF	NMFSC	Semi-NMF	GNMF	ONMF	NeNMF	Semi-GNMFSC$_V$	Semi-GNMFSC$_U$
5	0.5088	0.6543	0.5243	0.5868	0.5863	0.6833	0.6121	0.5521	0.6681	0.6588
10	0.4811	0.6052	0.5043	0.5768	0.5713	0.6521	0.5839	0.5354	0.6558	0.6781
20	0.4527	0.5843	0.4775	0.5721	0.5689	0.6479	0.5732	0.5023	0.7826	0.7954
30	0.4202	0.5633	0.4522	0.5567	0.5201	0.6084	0.5631	0.4932	0.8132	0.8267
40	0.4055	0.5454	0.4324	0.5299	0.5132	0.5933	0.5412	0.4746	0.8109	0.8043
50	0.3790	0.5041	0.3853	0.4975	0.4025	0.5764	0.4866	0.4574	0.7868	0.7917
						NMI				
5	0.7169	0.7643	0.7243	0.7168	0.7313	0.8333	0.8233	0.7132	0.7760	0.7675
10	0.7267	0.7526	0.7134	0.7080	0.7180	0.8321	0.8132	0.7025	0.8325	0.8316
20	0.7051	0.7439	0.7011	0.6951	0.7041	0.8145	0.8020	0.7023	0.8966	0.8999
30	0.6923	0.7367	0.6991	0.6877	0.6906	0.8088	0.7924	0.6924	0.9202	0.9245
40	0.6888	0.7243	0.6865	0.6799	0.6853	0.8012	0.7832	0.6824	0.8934	0.8985
50	0.6711	0.7143	0.6853	0.6675	0.6825	0.7983	0.7866	0.6810	0.8283	0.8396
						Purity				
5	0.5172	0.6359	0.5543	0.6168	0.6113	0.7133	0.6933	0.6033	0.6556	0.6467
10	0.5107	0.6243	0.5483	0.6132	0.6007	0.7076	0.6924	0.5944	0.7229	0.7211
20	0.5081	0.6047	0.5300	0.5825	0.5925	0.6833	0.6888	0.5888	0.8306	0.8335
30	0.4981	0.5943	0.5265	0.5775	0.5844	0.6744	0.6732	0.5732	0.8556	0.8610
40	0.4834	0.5888	0.5125	0.5682	0.5450	0.6678	0.6611	0.5611	0.8354	0.8348
50	0.4711	0.5742	0.5153	0.5566	0.5325	0.6549	0.6566	0.5566	0.7632	0.8396

Table 3 : Clustering results of different algorithms on ORL data

					ACC					
K	K-means	PCA	NMF	NMFSC	Semi-NMF	GNMF	ONMF	NeNMF	Semi-GNMFSC$_V$	Semi-GNMFSC$_U$
5	0.3525	0.4843	0.2743	0.4368	0.3813	0.4475	0.3925	0.4632	0.4525	0.4775
10	0.4511	0.4721	0.2610	0.4250	0.3675	0.4600	0.3875	0.4525	0.4825	0.4975
20	0.4213	0.4443	0.1961	0.4175	0.3750	0.4525	0.3732	0.4432	0.4575	0.4613
30	0.4122	0.4477	0.1640	0.4050	0.4150	0.4433	0.3628	0.4332	0.5075	0.4925
40	0.4011	0.4819	0.1553	0.3975	0.4025	0.4232	0.3766	0.4032	0.4888	0.4900
50	0.3959	0.4743	0.1478	0.3932	0.3921	0.4123	0.3621	0.3932	0.4794	0.4650
					NMI					
5	0.5491	0.6043	0.5643	0.5868	0.5813	0.6698	0.6258	0.6133	0.664	0.6743
10	0.5353	0.5911	0.5550	0.5750	0.5923	0.6853	0.6175	0.6062	0.693	0.6932
20	0.5251	0.5842	0.5311	0.5851	0.5841	0.6537	0.5919	0.5941	0.6798	0.6803
30	0.5115	0.5755	0.5262	0.564	0.5706	0.6433	0.5888	0.5804	0.6921	0.6864
40	0.5011	0.5639	0.5153	0.5475	0.5625	0.6365	0.5766	0.5739	0.6700	0.6713
50	0.4921	0.5511	0.5003	0.5358	0.5401	0.6274	0.5675	0.5518	0.6672	0.6671
					Purity					
5	0.5175	0.5243	0.5001	0.5668	0.5413	0.4498	0.425	0.4962	0.4950	0.5800
10	0.4981	0.5702	0.5025	0.4650	0.3875	0.5075	0.4923	0.5800	0.5975	0.6000
20	0.4888	0.5511	0.4900	0.5525	0.4125	0.5425	0.4872	0.5611	0.5775	0.5750
30	0.4752	0.5473	0.4829	0.5375	0.4625	0.5033	0.4575	0.5437	0.5525	0.5400
40	0.4645	0.5299	0.4753	0.4975	0.4025	0.4932	0.4866	0.5123	0.5410	0.5425
50	0.4534	0.5145	0.4625	0.4875	0.3850	0.4833	0.4455	0.5078	0.5350	0.5375

Based on the findings presented in the table above, it is evident that the Semi-GNMFSC model outperforms other models to a significant degree. This superiority can be primarily attributed to the synergistic impact of sparse constraint and graph regularization constraint. Through summarizing and scrutinizing the experimental outcomes across the aforementioned five datasets, a comprehensive analysis of the results can be derived.

(1) Firstly, the Semi-GNMFSCU algorithm has exhibited superior performance across all five datasets and displayed robust stability in handling diverse types of data. Specifically, in the COIL20 dataset, compared to the worst performing NMF, Semi-GNMFSCU has achieved the highest improvements in ACC, NMI, and Purity by 41.27%, 22.79%, and 6.85% respectively. In the ORL dataset, compared with the worst performing NMF, Semi-GNMFSC$_U$ has achieved the highest improvements in ACC, NMI, and Purity by 33.47%, 16.02% and 7.5% respectively. In the PIE dataset, compared with the worst-performing Semi-NMF, Semi-GNMFSCU has achieved the highest improvements in ACC, NMI, and Purity by 11.56%, 14.46% and 13.79% respectively. In the AR dataset, compared with the worst-performing GNMF, Semi- GNMFSCU has achieved the highest improvements in ACC, NMI, and Purity by 38.75%, 34.38%, and 35.95% respectively. In the Yale dataset, compared to the worst performing Semi-NMF, Semi-GNMFSC$_U$ has achieved the highest improvements in ACC, NMI, and Purity by 17.01%, 18.7% and 19.85% respectively. The superior performance of the Semi-GNMFSCU algorithm can be attributed to the

Table 4 : Clustering results of different algorithms on PIE data

K	K-means	PCA	NMF	NMFSC	Semi-NMF	GNMF	ONMF	NeNMF	Semi-GNMFSC$_V$	Semi-GNMFSC$_U$
					ACC					
5	0.0950	0.1432	0.0770	0.0768	0.0668	0.1783	0.0720	0.1032	0.1356	0.1499
10	0.0938	0.1143	0.1108	0.1005	0.0836	0.1740	0.0788	0.0988	0.1816	0.1729
20	0.0949	0.1042	0.0975	0.0921	0.0789	0.1613	0.0692	0.0932	0.1860	0.1724
30	0.0958	0.0999	0.0896	0.0901	0.0701	0.1532	0.0669	0.0830	0.1707	0.1857
40	0.0854	0.0970	0.0824	0.0899	0.0699	0.1468	0.0572	0.0762	0.1533	0.1553
50	0.0810	0.0888	0.0653	0.0795	0.0683	0.1436	0.0566	0.0731	0.1439	0.1566
					NMI					
5	0.2215	0.2543	0.1938	0.1933	0.1915	0.3611	0.1716	0.2632	0.3040	0.3109
10	0.2242	0.2369	0.2635	0.2367	0.2609	0.3535	0.1976	0.2724	0.3626	0.3494
20	0.2190	0.2211	0.2511	0.2151	0.2541	0.3433	0.1820	0.2723	0.3547	0.3490
30	0.2136	0.2143	0.2439	0.2091	0.2406	0.3307	0.1770	0.2838	0.3321	0.3441
40	0.2067	0.2009	0.2265	0.2000	0.2453	0.3274	0.1732	0.2724	0.3433	0.3542
50	0.2011	0.2000	0.2153	0.1975	0.1825	0.3133	0.1666	0.2510	0.3261	0.3271
					Purity					
5	0.1148	0.1443	0.0903	0.0938	0.0764	0.2056	0.0863	0.1133	0.1622	0.1785
10	0.1127	0.1341	0.1295	0.1142	0.0951	0.1760	0.0939	0.1022	0.1781	0.1765
20	0.1155	0.1167	0.1000	0.1025	0.0825	0.1833	0.0833	0.0988	0.1921	0.1966
30	0.1079	0.1043	0.1067	0.1065	0.0731	0.1692	0.0825	0.0924	0.1799	0.1982
40	0.0881	0.0944	0.0825	0.0973	0.0650	0.1533	0.0733	0.0811	0.1842	0.1939
50	0.0711	0.0943	0.0853	0.0877	0.0625	0.1422	0.0633	0.0766	0.2061	0.2004

effects of sparse constraint and graph regularization constraint. Incorporating a sparse constraint into the basis matrix U leads to a more sparsely represented basis matrix, thereby enhancing the local representation capability of the model. Furthermore, the inclusion of a graph regularization term effectively constrains the coefficient matrix to better preserve latent geometric structure information within the dataset.

(2) Secondly, Semi-GNMFSC$_V$ exhibits a slightly lower performance in clustering compared to the Semi-GNMFSC$_U$ model. The reason analysis is as follows. Although the addition of sparse constraints to the coefficient matrix in the Semi-GNMFSCV model results in a sparser matrix representation, it is crucial to note that while the coefficient matrix V serves as a substitute for the original data matrix after dimensionality reduction, there is a risk of over-reduction. That is to say, the incorporation of sparse constraints do not necessarily enhance local representation effects like in the basis matrix U , instead, it has the potential to compromise essential latent information within the data and consequently diminish clustering performance. The experimental results also confirmed that the clustering performance of the Semi-GNMFSC$_V$ model is remarkably inferior to that of the Semi-GNMFSC$_U$ model. Consequently, the Semi-GNMFSC$_U$ model exhibits strong applicability and generalization. In future research, we will furthermore investigate how to adequately harness the advantages of sparse constraint and graph regularization constraint in coordination to enhance the adaptability and robustness of the model in various complex scenarios.

(3) Thirdly, the Semi-GNMFSC algorithm exhibits

Table 5 : Clustering results of different algorithms on AR data

K	K-means	PCA	NMF	NMFSC	Semi-NMF	GNMF	ONMF	NeNMF	Semi-GNMFSC$_V$	Semi-GNMFSC$_U$
						ACC				
5	0.3798	0.3744	0.2762	0.2732	0.2494	0.0417	0.3196	0.2827	0.3863	0.3988
10	0.3857	0.4316	0.3071	0.3268	0.2423	0.0810	0.3387	0.3339	0.4512	0.4685
20	0.3774	0.4243	0.3643	0.3875	0.3161	0.1429	0.3431	0.3245	0.4583	0.4958
30	0.3792	0.4107	0.3425	0.3851	0.3268	0.2101	0.3631	0.4732	0.4845	0.5024
40	0.3647	0.4243	0.4054	0.3786	0.4048	0.2399	0.3412	0.5068	0.4571	0.5107
50	0.3821	0.4337	0.4381	0.4411	0.3768	0.2964	0.3388	0.5125	0.4464	0.5095
						NMI				
5	0.637	0.6543	0.6422	0.6220	0.5998	0.3498	0.6604	0.6277	0.6886	0.6936
10	0.6478	0.6679	0.6504	0.6641	0.5774	0.4297	0.6797	0.6689	0.7293	0.7313
20	0.6597	0.6812	0.6916	0.6999	0.6346	0.5603	0.6899	0.7410	0.7462	0.7533
30	0.6667	0.6831	0.7143	0.7078	0.6373	0.6258	0.7024	0.7621	0.7526	0.7605
40	0.6676	0.6843	0.7173	0.7032	0.6853	0.6618	0.7532	0.7640	0.7482	0.7671
50	0.6511	0.6743	0.7053	0.7075	0.6025	0.6833	0.7167	0.7991	0.7411	0.7643
						Purity				
5	0.3298	0.3643	0.2964	0.2911	0.2762	0.0417	0.3470	0.2994	0.3792	0.4226
10	0.3300	0.4544	0.328	0.3494	0.2583	0.0810	0.3768	0.3542	0.4512	0.4815
20	0.3107	0.4781	0.3929	0.4071	0.3423	0.1429	0.3917	0.4649	0.4833	0.5024
30	0.3081	0.4523	0.4000	0.4375	0.4625	0.2327	0.3832	0.4732	0.4845	0.5089
40	0.3177	0.4632	0.4125	0.4175	0.4850	0.2535	0.3902	0.5125	0.4744	0.5190
50	0.3011	0.4243	0.4053	0.4975	0.4025	0.2542	0.4129	0.5318	0.4464	0.5155

strong adaptability in datasets of varying scales and complexities. Whether dealing with tiny-scale, straight forward data or large-scale, intricate data, the algorithm consistently achieves effective clustering and delivers satisfactory outcomes. These findings underscore the broad applicability of the Semi-GNMFSC algorithm and its capacity to showcase advantages in diverse practical scenarios, thereby offering crucial support for data analysis and dimensionality reduction.

(4) Finally, it is essential to highlight that while the Semi-GNMFSC$_U$ model demonstrates superior performance in handling datasets such as COIL20, Yale and AR. Next, in the presence of noise, we will mainly test our algorithm on the well performing (without noise) COIL2 and Yale datasets, as a special case to illustrate the advantages of our algorithm. Of course, even with these well performing datasets, there is still room for improvement in our algorithms. Therefore, additional additional optimization remains a crucial area for future research, thus paving the way for different avenues of investigation.

Discussion on Noise Robustness

In the above experiments, the performance of the proposed method is better than that of the comparison methods. In order to further prove the robustness of Semi-GNMFSC model, we carry out experiments on noisy data. In particular, in our experiments, we consider two types of noise including salt and pepper noise and Gaussian noise. For simplicity, consider only the case where the feature dimension K = 30, and then we will conduct experiments on the COIL20 and Yale datasets. First, salt and pepper noise with noise level (density) 20%, 40% and 60% is added to the dataset, respectively.

Next, we add Gaussian noise with a mean of 0 and variances within {0.001, 0.01, 0.1} to the dataset, called light, medium, and heavy noise conditions in this test. Part of the data sets with noise image is displayed as follows:

Table 6 : Clustering results of different algorithms on Yale data

K	K-means	PCA	NMF	NMFSC	Semi-NMF	GNMF	ONMF	NeNMF	Semi-GNMFSCv	Semi-GNMFSCu
ACC										
5	0.3394	0.4143	0.3242	0.3758	0.400	0.3333	0.3879	0.3818	0.4061	0.4176
10	0.3136	0.3752	0.3273	0.3697	0.3455	0.3091	0.3870	0.3576	0.4161	0.4073
20	0.3276	0.3443	0.3455	0.3818	0.3152	0.3212	0.3455	0.3488	0.4444	0.4212
30	0.3394	0.3633	0.3636	0.3030	0.2606	0.3394	0.3424	0.3558	0.3981	0.4052
40	0.3055	0.3454	0.3324	0.3299	0.2132	0.3433	0.3112	0.3746	0.3771	0.3833
50	0.3394	0.3541	0.3212	0.3515	0.2303	0.3152	0.3818	0.3030	0.3868	0.3897
NMI										
5	0.4201	0.4447	0.3561	0.4504	0.4494	0.3873	0.4572	0.4479	0.4831	0.4972
10	0.3825	0.4026	0.3868	0.4431	0.4203	0.3727	0.4436	0.4438	0.4833	0.4818
20	0.3988	0.4088	0.3933	0.4099	0.3725	0.3743	0.4018	0.4097	0.4851	0.4798
30	0.4077	0.4265	0.4035	0.3924	0.2911	0.3818	0.4126	0.4222	0.4719	0.4781
40	0.4188	0.4578	0.3965	0.4499	0.3153	0.3012	0.4432	0.4324	0.4529	0.4694
50	0.4017	0.4299	0.4125	0.4293	0.2854	0.3716	0.4311	0.3854	0.4483	0.4555
Purity										
5	0.3515	0.4059	0.3242	0.4000	0.4061	0.3333	0.3697	0.3879	0.4115	0.4186
10	0.3636	0.3822	0.3636	0.3879	0.3697	0.3212	0.4000	0.3697	0.4128	0.4033
20	0.3439	0.3647	0.3758	0.3700	0.3273	0.3394	0.3576	0.3758	0.4222	0.4211
30	0.3558	0.3743	0.3817	0.3555	0.2788	0.3333	0.3424	0.3769	0.4694	0.4773
40	0.3434	0.3688	0.3125	0.3682	0.3450	0.3678	0.3213	0.3611	0.4084	0.4094
50	0.3577	0.3811	0.3428	0.3576	0.2424	0.3273	0.3761	0.3273	0.3682	0.3702

Table 7 : Clustering results of COIL20 data set containing salt-and-pepper noise

Noise Density	K-means	PCA	NMF	ONMF	GNMF	NMFSC	Semi-NMF	NeNMF	Semi-GNMFSCv	Semi-GNMFSCu
ACC										
20%	0.4317	0.4998	0.4896	0.6229	0.6924	0.4882	0.5181	0.5366	0.7153	0.7177
40%	0.4196	0.4835	0.4713	0.4924	0.4486	0.4711	0.4333	0.4982	0.6160	0.6098
60%	0.2729	0.2617	0.2521	0.2833	0.1847	0.2333	0.1549	0.2444	0.3701	0.3753
NMI										
20%	0.6598	0.6871	0.6777	0.7516	0.8291	0.6711	0.657	0.6724	0.8455	0.846
40%	0.5074	0.5623	0.5674	0.5896	0.5318	0.5641	0.5309	0.5123	0.6921	0.6999
60%	0.3787	0.3399	0.3012	0.3507	0.3521	0.3029	0.3199	0.3749	0.4354	0.4321
Purity										
20%	0.4498	0.5020	0.5177	0.5316	0.5291	0.4723	0.457	0.4619	0.7455	0.7500
40%	0.4274	0.4483	0.4674	0.4819	0.4318	0.4508	0.4308	0.4488	0.5921	0.5853
60%	0.2297	0.2471	0.2512	0.2604	0.2584	0.2308	0.2185	0.2231	0.3554	0.3642

Table 8 : Clustering results of Yale data set containing salt-and-pepper noise

Noise Density	K-means	PCA	NMF	ONMF	GNMF	NMFSC	Semi-NMF	NeNMF	Semi-GNMFSCV	Semi-GNMFSCU
ACC										
20%	0.4317	0.4998	0.4896	0.6229	0.6924	0.4882	0.5181	0.5366	0.7153	0.7177
40%	0.4196	0.4835	0.4713	0.4924	0.4486	0.4711	0.4333	0.4982	0.6160	0.6098
60%	0.2729	0.2617	0.2521	0.2833	0.1847	0.2333	0.1549	0.2444	0.3701	0.3753
NMI										
20%	0.6598	0.6871	0.6777	0.7516	0.8291	0.6711	0.6570	0.6724	0.8455	0.8460
40%	0.5074	0.5623	0.5674	0.5896	0.5318	0.5641	0.5309	0.5123	0.6921	0.6999
60%	0.3787	0.3399	0.3012	0.3507	0.3521	0.3029	0.3199	0.3749	0.4354	0.4321
Purity										
20%	0.4498	0.5020	0.5177	0.5316	0.5291	0.4723	0.4570	0.4619	0.7455	0.7500
40%	0.4274	0.4483	0.4674	0.4819	0.4318	0.4508	0.4308	0.4488	0.5921	0.5853
60%	0.2297	0.2471	0.2512	0.2604	0.2584	0.2308	0.2185	0.2231	0.3554	0.3642

From the above experimental results, it can be seen that the Semi-GNMFSC algorithm proposed in this paper has strong robustness. By observing the above tables, we can draw the following conclusions:

(1). As the level of noise fluctuates from light to heavy, there is a significant decrease in the performance of all algorithms, indicating that noise interference has a detrimental impact on image clustering. Semi-GNMFSCV and Semi-GNMFSCU demonstrate robustness against noise by achieving favorable results across multiple datasets. For the COIL20 dataset with pepper and salt noise, the Semi-GNMFSCU algorithm has demonstrated significant improvements in clustering indicators com- pared to other methods: ACC increased by up to 28.6%, NMI increased by up to 19.27%, and Purity increased by up to 30.02%. Similarly, in the presence of Gaussian noise, the Semi-GNMFSCU algorithm showed improved performance with an increase of up to 15.64% in ACC, up to 21.26% in NMI, and up to 22.47% in Purity when compared with other methods on the COIL20 dataset. Furthermore, on the Yale dataset containing various levels of Gaussian noise and pepper-and-salt noise, the Semi-GNMFSCU algorithm exhibits superior clustering performance. Of course, for the Semi-GNMFSCV algorithm, its performance's improvement is also more comprehensive (omitting details). Notably, even under noisy conditions, our algorithms effectively extract hidden feature information from data while maintaining strong accuracy and reliability.

(2). Based on the tables above, it is evident that the Semi-GNMFSCU model demonstrates superior robustness in a comprehensive evaluation. This indicates its ability to maintain consistent performance across diverse and complex scenarios, showcasing strong resistance to interference and a high level of generalization. It is worth noting that the Semi-GNMFSCV model also exhibits commendable robustness, ranking closely behind the Semi-GNMFSCU model. Overall, both models display impressive resilience. Through our experiments, we can conclude that when facing a large amount of contaminated redundant data in the original matrix, introducing sparse constraints into the basis matrix U and using the Semi-GNMFSCU algorithm can effectively identify the potential expressive feature data. Consequently, this approach mitigates the impact of contaminated data and enhances overall model robustness.

(3). The Semi-GNMFSCV model, which introduces

Table 9 : Clustering results of COIL20 data set containing Gaussian noise

Noise Variance	K-means	PCA	NMF	ONMF	GNMF	NMFSC	Semi-NMF	NeNMF	Semi-GNMFSCv	Semi-GNMFSCu
ACC										
0.001	0.5454	0.5877	0.5642	0.5703	0.6222	0.5681	0.6090	0.5933	0.6899	0.7018
0.01	0.5372	0.5643	0.5444	0.5333	0.5583	0.5181	0.5653	0.5699	0.6475	0.6470
0.1	0.5242	0.5200	0.5021	0.4950	0.5190	0.4874	0.5283	0.5321	0.6006	0.6111
NMI										
0.001	0.6866	0.7453	0.7345	0.7516	0.8308	0.6957	0.6976	0.8122	0.8888	0.8992
0.01	0.6528	0.7095	0.6953	0.7567	0.8150	0.7183	0.7284	0.7439	0.8428	0.842
0.1	0.6032	0.6879	0.6687	0.7203	0.7911	0.7068	0.6974	0.7092	0.8015	0.8194
Purity										
0.001	0.6798	0.6999	0.6877	0.6916	0.7291	0.6711	0.6627	0.7124	0.7955	0.7966
0.01	0.4674	0.5075	0.4874	0.5896	0.5318	0.5141	0.5309	0.5439	0.6921	0.6800
0.1	0.3787	0.3904	0.3812	0.3507	0.3521	0.3429	0.3499	0.3524	0.4354	0.4483

Table 10 : Clustering results of Yale data set containing Gaussian noise

Noise Variance	K-means	PCA	NMF	ONMF	GNMF	NMFSC	Semi-NMF	NeNMF	Semi-GNMFSCv	Semi-GNMFSCu
ACC										
0.001	0.3394	0.3777	0.3576	0.3879	0.3212	0.3311	0.3467	0.3563	0.3909	0.4013
0.01	0.2839	0.3102	0.2921	0.3161	0.3152	0.3033	0.3064	0.3222	0.3800	0.3778
0.1	0.2027	0.2200	0.2145	0.2252	0.2109	0.2009	0.2082	0.2209	0.2788	0.2875
NMI										
0.001	0.3519	0.4285	0.4266	0.4383	0.3977	0.3899	0.3924	0.4075	0.4451	0.4506
0.01	0.3476	0.3827	0.3222	0.4041	0.3616	0.3852	0.3550	0.3799	0.4185	0.4198
0.1	0.3130	0.3608	0.3069	0.3762	0.3574	0.3616	0.3271	0.3421	0.3836	0.3954
Purity										
0.001	0.3398	0.3598	0.3577	0.3616	0.3595	0.3511	0.3570	0.3603	0.3955	0.3975
0.01	0.2974	0.3089	0.3071	0.3196	0.3099	0.3141	0.3169	0.3198	0.3521	0.3500
0.1	0.2187	0.2271	0.2212	0.2307	0.2299	0.2329	0.2401	0.2500	0.2554	0.2675

sparse constraints into the coefficient matrix V,exhibits a slight lower superiority over the Semi-GNMFSCu model in terms of enhancing robustness. This is attributed to the fact that the Semi-GNMFSCv model enforces sparse constraints on the coefficient matrix, resulting in the exclusion of numerous contaminated feature data and yielding a more transparent and sparser coefficient matrix. But compared to the Semi-GNMFSCu algorithm, this sparsity enhancement is slightly excessive, and it is possible that the appropriate sparsity has already been transmitted to the matrix V in the Semi-GNMFSCu algorithm. Note that the coefficient matrix serves as a proxy for the original matrix in subsequent data clustering operations. From the experimental results, it can be seen that an appropriate enhancement of V sparsity helps to comprehensively enhance the robustness of the model.

(4). Of course, although the overall performance of

the Semi-GNMFSCv algorithm is slightly unsatisfactory than Semi-GNMFSCu, we should also see that the Semi-GNMFSCv model with sparse constraints also performs well on a slightly smaller scale. It not only improves the robustness to noise interference and outliers to a certain extent, but also effectively reduces redundant information between features, thereby helping to extract more representative and discriminative features. Therefore, the Semi-GNMFSCV algorithm has also produced more accurate and reliable results in data clustering tasks, which is worth further application and research.

CONCLUSIONS AND FUTURE RESEARCH

In this paper, firstly, we have introduced the idea of Semi-NMF algorithm and l_1 sparse constraint together to graph based factorization method, and then we have established a new type of Semi-NMF model (Semi-GNMFSC). The aim is to relax the constraints to get better performance on the original matrix and factorized matrices. More importantly, we provide the multiplication update rules and the convergence theorems (with proofs), which has been also used for analyzing experiments. It should be emphasized that the combination of semi-NMF, GNMF, and sparse constraints is the characteristic of this paper.

So, as a conclusion, there are two questions that may bring more interesting work in the near future.

(1) In this paper, Euclidean distance is used to measure loss and define loss functions, but there are many ways to measure residuals on the real, such as the common β-divergence, so in the future research work, we can choose different loss and loss functions according to different real applications.

(2) The construction of manifold structure is the focus of manifold regularization algorithm, and the construction of graph will directly affect the performance of the algorithm. At present, a large number of graph construction methods have been proposed, and different structure-relation construction methods are suitable for different data sets. Therefore, how to combine manifold to establish models under NMF algorithm is still a fascinating problem.

DECLARATIONS

Acknowledgments

The authors would like to thank the anonymous referees for carefully reading the manuscript, giving valuable comments and suggestions to improve the results as well as the exposition of the paper. The first author is partially supported by the Natural Science Foundations of Tianjin City, China (Grant No. 18JCYBJC16300 and No. 22JCYBJC01470).

Conflict of Interest

The authors declare that there is no conflict of interests, we do not have any possible conflicts of interest.

Data Availability

The code generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

REFERENCES

1. Belhumeur PN, Hespanha JP, Kriegman DJ, et al. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. In: Computer Vision-ECCV'96.1996;43–58.
2. Belkin M, Niyogi P. Laplacian eigenmaps and spectral techniques for embedding and clustering. Advances in neural information processing systems. 2001;14.
3. Belkin M, Niyogi P, Sindhwani V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of machine learning research. 2006; 7.
4. Cai D, He X, Wu X, et al. non-negative matrix factorization on manifold. In: eighth IEEE international conference on data mining. 2008.
5. Cai D, He X, Wang X, et al. Locality preserving nonnegative matrix factorization. In: Twenty-first international joint conference on artificial intelligence.2009.
6. Cai D, He X, Han J, et al. Graph regularized nonnegative matrix factorization for data representation. IEEE transactions on pattern analysis and machine intelligence.2010;33:1548–1560.
7. Chen WS, Zhao Y, Pan B, et al. Supervised kernel nonnegative matrix factorization for face recognition. Neurocomputing. 2016; 205:165–181.
8. Chung FR. Spectral graph theory. American Mathematical Soc.1997.

9. Ding C, Li T, Peng W, et al. Orthogonal nonnegative matrix t-factorizations for clustering. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. 2006;126–135.
10. Ding C, Li T, Jordan MI. Convex and semi-nonnegative matrix factorizations. IEEE transactions on pattern analysis and machine intelligence.2008;32:45–55.
11. Draper BA, Baek K, Bartlett MS, et al. Recognizing faces with pca and ica. Computer vision and image understanding. 2003; 91:115–137.
12. Eggert J, Korner E. Sparse. coding and nmf. In: 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541). IEEE. 2004;2529–2533.
13. Gao Y, Church G. Improving molecular cancer class discovery through sparse non-negative matrix factorization. Bioinformatics. 2005; 21:3970–3975.
14. Guan N, Tao D, Luo Z, et al. Nenmf: An optimal gradient method for nonnegative matrix factorization. IEEE Transactions on Signal Processing.2012;60:2882–2898.
15. Guo W. Sparse dual graph-regularized deep nonnegative matrix factorization for image clustering. IEEE Access. 2021;9:39926–39938.
16. Hoyer PO. non-negative sparse coding. In: Proceedings of the 12th IEEE workshop on neural networks for signal processing. 2002;IEEE. 557–565.
17. Hoyer PO. Non-negative matrix factorization with sparseness constraints. Journal of machine learning research.2004;5.
18. Huang S, Wang H, Li T, et al. Robust graph regularized nonnegative matrix factorization for clustering. Data Mining and Knowledge Discovery. 2018;32:483–503.
19. Hyvarinen A, Oja E. Independent component analysis: algorithms and applications. Neural networks. 2000;13:411–430.
20. Ioshisaqui AS, Attux R, Luna I. Analysis of occupational profiles in the Brazilian workforce based on non-negative matrix factorization. Big Data Research. 2022;29:100333.
21. Jain AK, Dubes RC. Algorithms for clustering data. Prentice-Hall Inc.1988.
22. Jolliffe IT, Cadima J. Principal component analysis: a review and recent developments. Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences. 2016; 374:20150202.
23. Kevin WW, Bhiksha R, Paris S, et al. Speech denoising using nonnegative matrix factorization with priors. 2008 IEEE International Conference on Acoustics,
Speech and Signal Processing. 2008;4029–4032.
24. Lee D, Seung HS. Algorithms for non-negative matrix factorization. Advances in neural information processing systems. 2000;13.
25. Lee DD, Seung HS. Learning the parts of objects by non-negative matrix factorization. Nature. 1999;401:788–791.
26. Li R, Yang G, Wang K, et al. Robust ecg biometrics using gnmf and sparse representation. Pattern Recognition Letters.2020;129:70–76.
27. Liu F, Yang X, Guan N, et al. Online graph regularized non-negative matrix factorization for large-scale datasets. Neurocomputing. 2016;204:162–171.
28. Liu H, Wu Z, Li X, et al. Constrained nonnegative matrix factorization for image representation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2011; 34:1299–1311.
29. Yu J, Plataniotis KN, Venetsanopoulos AN. Face recognition using lda-based algorithms. IEEE Transactions on Neural networks. 2003;14:195–200.
30. Lu X, Wu H, Yuan Y, et al. Manifold regularized sparse nmf for hyperspectral unmixing. IEEE Transactions on Geoscience and Remote Sensing. 2012; 51:2815–2826.
31. Lyu S, Wang X. On algorithms for sparse multi-factor nmf. Advances in Neural Information Processing Systems 2013;26.
32. Mackiewicz A, Ratajczak W. Principal components analysis (pca). Computers & Geosciences. 1993;19(3):303–342.
33. Martinez A, Benavente R. The art face database: Cvc technical report. 1998;24.
34. Mohammadiha N, Leijon A. Nonnegative matrix factorization using projected gradient algorithms with sparseness constraints. In: 2009 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT). IEEE. 2009; 418–423.
35. Nene SA, Nayar SK, Murase H, et al. Columbia object image library. 1996.
36. Paatero P, Tapper U. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. Environmetrics. 1994; 5:111–126.
37. Peng C, Zhang Z, Chen C, et al. Two-dimensional semi-nonnegative matrix factorization for clustering. Information Sciences. 2022;590:106–141.
38. Roweis ST, Saul LK. Nonlinear dimensionality reduction by locally linear embedding. Science. 2000; 290:2323–2326.
39. Samaria FS, Harter AC. Parameterisation of a stochastic model for human face identification. In: Proceedings of 1994 IEEE workshop on applications of computer vision. IEEE. 1994;138–142.
40. Sim T, Baker S, Bsat M. The cmu pose, illumination and expression database of human faces. Carnegie Mellon University Technical Report CMU-RI-TR-OI-02. 2001.
41. Tutte WT. Graph theory. Cambridge university press. 2001;24.
42. Wang J, Wang J, Ke Q, et al. Fast approximate k-means via cluster closures. 2012 IEEE Conference on Computer Vision and Pattern Recognition. 2015.
43.Wright SJ. Numerical optimization. 2006.
44. Xie Z, Mu Z. Ear recognition using lle and idlle algorithm. In. International conference on pattern recognition. IEEE.2008;19;1–4.
45. Yu H, Yang J. A direct lda algorithm for high-dimensional data—with application to face recognition. Pattern recognition. 2001; 34:2067–2070.
46. Zeng K, Yu J, Li C, et al. Image clustering by hyper-graph regularized non negative matrix factorization. Neurocomputing.2014;138:209–217.
47. Zhao Y, Wang C, Pei J, et al. Nonlinear loose coupled non-negative matrix factorization for low-resolution image recognition. Neurocomputing. 2021; 443:183–198.
48. Zhou J, Qian Y, Minchao. Multitask sparse nonnegative matrix factorization for joint spectral-spatial hyperspectral imagery denoising. IEEE Transactions on Geoscience and Remote Sensing. 2015; 53:2621–2639.