Structured Sparse Deep Nonnegative Matrix Factorization with $\ell_{2,1}$-norm and Graph Regularization for Clustering

Weifeng Yang
Yunnan Earthquake Agency
Kunming, China
ywf841673182@gmail.com

***Abstract*—Deep nonnegative matrix factorization (DNMF) is a powerful tool for feature extraction, and promoting the sparsity of the factor matrices in DNMF can improve both the quality of extracted features and model interpretability. However, most existing DNMF methods promote the sparsity of factor matrices by individually evaluating the importance of each element in factor matrices and pruning unimportant elements (e.g., via the ℓ_1-norm). This element-wise sparsity fails to capture the interactions and dependencies between different features and neglects the inherent structure of features, impairing the quality of extracted features and interpretability. To overcome this drawback, we propose a novel structured-sparse and graph-regularized deep nonnegative matrix factorization with $\ell_{2,1}$-norm regularization ($\ell_{2,1}$-SSGDNMF) method. Our method employs the $\ell_{2,1}$-norm to explicitly impose the group sparsity on factor matrices, which enables entire feature groups to be selected or discarded and effectively captures feature interactions and dependencies, thereby enhancing the feature extraction capability and interpretability. Additionally, $\ell_{2,1}$-SSGDNMF employs graph regularization to preserve the geometric structure information of data. To solve $\ell_{2,1}$-SSGDNMF, we propose an algorithm named the Alternating Proximal Linearized (APL) algorithm, which provides an efficient and practical convergent scheme for solving $\ell_{2,1}$-SSGDNMF. Furthermore, we prove that the sequence generated by our algorithm is globally convergent to a critical point and analyze the per-iteration complexity of our algorithm. Experimental results on six real-world benchmark datasets demonstrate that our method outperforms several state-of-the-art methods for clustering.**

***Index Terms*—Deep nonnegative matrix factorization, nonconvex nonsmooth optimization, group sparsity, graph, clustering**

I. Introduction

With the rapid advancement of data acquisition technology, the amount and complexity of data collected from real-world scenarios, such as EEG (Electroencephalography) data [1] and seismic data [2], have increased enormously. Therefore, utilizing effective methods to extract informative and compact representations from these enormous and complex data is an important research topic in machine learning, and this extraction process is known as feature extraction. Nonnegative matrix factorization (NMF) is a commonly used method for data redundancy reduction and feature extraction [3], [4]. Since the single-layer structure of NMF makes it difficult to extract underlying hierarchical features from data, to overcome this defect, deep nonnegative matrix factorization (DNMF) extends NMF to a multi-layer hierarchical structure, which is inspired by the deep learning principle that decomposes complex tasks into several parts and solves them progressively within a deep structure, enabling DNMF to extract underlying hierarchical features from data and provide an easily interpretable representation, thereby enhancing the feature extraction capability [5], [6]. Therefore, DNMF has emerged as a powerful tool for data analysis over the past decade.

Incorporating the prior knowledge of data into the model is an effective way to improve the quality of extracted features. For instance, since the data collected from real-world scenarios are often nonnegative, DNMF can incorporate this prior knowledge into the model by imposing nonnegativity constraints on the factor matrices, thereby extracting higher-quality features [7], [8]. Additionally, since the underlying geometric structure of data can also be considered as the prior data knowledge, some researchers have recently proposed the graph regularized DNMF methods that preserve geometric structure information of data by adding graph regularization into DNMF, for example, [9] introduced a DNMF method with graph regularization and L_p smoothing norm, [10] proposed a deep asymmetric NMF method with graph regularization.

Given the prior knowledge that the data inherently contain substantial redundant information and noise, forcing the factor matrices in deep matrix factorization to be sparse can effectively remove irrelevant features and redundant information in the data, thereby improving the quality of extracted features and interpretability [11], [12]. A straightforward and common scheme for achieving the sparsity of the solution is to employ element-wise sparse norm regularization, which individually evaluates the weight of each element in the solution and then sets unimportant elements to zero based on their weights,

The work was supported in part by the CEA Youth Key Project on Earthquake Information (No. CEAITNS202607), the Spark Program of Earthquake Sciences of China Earthquake Administration (XH25033YB), and Earthquake Science Technology Innovation Team Project of Yunnan Province (CXTD202507).

Corresponding author: Weifeng Yang

thereby promoting the sparsity of the solution (e.g., via the ℓ_1-norm). The sparsity achieved by this scheme is also known as element-wise sparsity. Consequently, several researchers have proposed sparse DNMF methods that use element-wise sparse norm regularization to promote the element-wise sparsity of factor matrices, for example, [13] and [14] respectively proposed sparse DNMF methods that use ℓ_1-norm regularization to promote the element-wise sparsity of factor matrices, [15] introduced a ℓ_1-norm regularized DNMF method that promotes the element-wise sparsity of a single factor matrix.

However, although this element-wise sparsity achieved by the element-wise sparsity regularization is straightforward and commonly used in deep matrix factorization methods, it neglects the prior data knowledge of both the interactions and dependencies between different features since it individually evaluates the importance of each element of factor matrices. Moreover, this element-wise sparsity neglects the inherent structure of features since it may zero out the contributions of the same feature across different dimensions individually, failing to completely remove or entirely preserve latent features [11], [16]. An additional limitation of most existing DNMF methods is that they either neglect the prior knowledge of the geometric structure information of data or consider the sparsity of a single factor matrix while overlooking the sparsity of other factor matrices. Therefore, the above drawbacks reduce the feature extraction capability and interpretability of such deep matrix factorization methods.

To overcome these drawbacks, we incorporate the $\ell_{2,1}$-norm regularization and graph regularization into DNMF, thereby proposing a novel structured-sparse and graph-regularized deep nonnegative matrix factorization with $\ell_{2,1}$-norm regularization ($\ell_{2,1}$-SSGDNMF) method. Compared with other deep matrix factorization methods, our method directly employs the $\ell_{2,1}$-norm, a structured sparse norm regularization, to explicitly impose the row-wise sparsity (group sparsity) on factor matrices in DNMF. This group sparsity of factor matrices introduces the structured sparsity pattern into DNMF by promoting the simultaneous selection or removal of entire feature groups, thereby accounting for both the interactions and dependencies between features and enhancing both the feature extraction capability and interpretability. Furthermore, $\ell_{2,1}$-SSGDNMF also incorporates the graph regularization to preserve the geometric structure information of data. Although the nonconvex nature of DNMF and the nonsmooth nature of $\ell_{2,1}$-norm make optimizing $\ell_{2,1}$-SSGDNMF challenging, we propose an Alternating Proximal Linearized (APL) algorithm, which provides an efficient and practical convergence scheme to solve the $\ell_{2,1}$-SSGDNMF. We prove that our algorithm ensures the monotonic convergence of the objective function of $\ell_{2,1}$-SSGDNMF and establish the global convergence of our algorithm as well as analyze the per-iteration complexity. We evaluate our method on six real-world benchmark datasets. Experimental results on these datasets demonstrate that our method outperforms several of the state-of-the-art clustering methods.

A. Contributions

We summarize our main contributions as follows.

(1) We propose a novel structured-sparse and graph-regularized deep nonnegative matrix factorization with $\ell_{2,1}$-norm regularization ($\ell_{2,1}$-SSGDNMF) method. This method employs the $\ell_{2,1}$-norm to explicitly impose the row-wise sparsity (group sparsity) on factor matrices, and this group sparsity enables entire feature groups to be selected or discarded, thereby achieving the structural sparsity of DNMF and enhancing both the feature extraction capability and interpretability. Moreover, $\ell_{2,1}$-SSGDNMF introduces the graph Laplacian regularization to preserve the geometric structure information of data, thereby integrating manifold learning and group sparsity into DNMF.

(2) We propose an Alternating Proximal Linearized (APL) algorithm to solve $\ell_{2,1}$-SSGDNMF. The APL algorithm provides an efficient and practical convergent scheme for solving $\ell_{2,1}$-SSGDNMF. Additionally, we prove that the sequence generated by our algorithm is globally convergent to a first-order critical point and analyze the per-iteration complexity.

(3) We evaluate the proposed $\ell_{2,1}$-SSGDNMF method on six real-world benchmark datasets. Experimental results on these datasets demonstrate that our method outperforms several state-of-the-art methods for clustering. We also investigate the parameter sensitivity of $\ell_{2,1}$-SSGDNMF and the effectiveness and convergence of our proposed algorithm. These experimental results can be found in Section VI.

The rest of this paper is organized as follows. We first introduce some notations and basic knowledge in Section II and present a brief review of related works in Section III. Then we present our proposed $\ell_{2,1}$-SSGDNMF method in Section IV and our proposed algorithm for solving $\ell_{2,1}$-SSGDNMF in Section V. We report the experimental results in Section VI and conclude this paper in Section VII. The Supplementary Material provides complete proofs of all theorems.

II. Notations

In this paper, we define $\mathbb{R}$ as the set of all real numbers, $\mathbb{N}$ as the set of all natural numbers, and $\mathbb{Z}$ as the set of all integer numbers.

For the definition of $\ell_{2,1}$-norm, given a matrix $M \in \mathbb{R}^{m_1 \times R}$, let $m_{i_1 i_2}$ denote the (i_1, i_2)-th component of M, then the $\ell_{2,1}$-norm of M is defined as

$$\|M\|_{2,1} := \sum_{i_1=1}^{m_1} \sqrt{\sum_{i_2=1}^{R} m_{i_1 i_2}^2}.$$

We also define $L = D - W$ as the graph Laplacian matrix. For the weight matrix W, we use the p-nearest neighbor graph and the heat kernel weight to construct it. Specifically, given a matrix $X \in \mathbb{R}^{m_1 \times R}$, where R is the sample size, let $W = \{w_{ij},\ i, j = 1, 2.....R\}$ be the weight matrix with heat kernel weight, then w_{ij} can be calculated as follows.

$$w_{ij} = \begin{cases} e^{\frac{\|X_{:,i} - X_{:,j}\|_F^2}{-\sigma^2}} & \text{if } X_{:,j} \in \mathcal{N}_p(X_{:,i}) \text{ or } X_{:,i} \in \mathcal{N}_p(X_{:,j}), \\ 0 & \text{else,} \end{cases} \tag{1}$$

where $X_{:,i}$ is the i-th column of X, $\mathcal{N}_p(X_{:,i})$ is the set of p samples closest to $X_{:,i}$ in the graph, and σ^2 denotes the divergence. We set $\sigma^2 = 1$ for our method.

Additionally, the notations used in this paper are summarized in Table I, the specific definitions of all related concepts in this paper can be found in the Supplementary Material.

Table I: Summary of frequently-used notations

Notation	Definition
$\{x_i\}_{i=1}^N$	$x_1, x_2 ....., x_N$
x_i^k	the i-th block of $\{x_i\}_{i=1}^N$ within k-th outer loop
$x_{(N)}$	$\{x_i\}_{i=1}^N$
$x_{(N)} + y_{(N)}$	$\{x_i + y_i\}_{i=1}^N$
$[N]$	$\{i\}_{i=1}^N$
$L_{\nabla_{x_j} H}$	the Lipschitz constant of $\nabla_{x_j} H(\{x_i\}_{i=1}^N)$
$\|\cdot\|_p$	l_p-norm
$\lceil a \rceil$ $(a \in \mathbb{R})$	$\lceil a \rceil := \min\{n \in \mathbb{Z} \mid n \geq a\}$
$\mathrm{Tr}(A)$ (A is a matrix)	the trace of A
A^T (A is a matrix)	the transpose of matrix A
$\langle \cdot, \cdot \rangle$	inner product
$\mathbb{R}_+$	$\{x \| x \in \mathbb{R},\ x \geq 0\}$
$\mathbb{I}_S(X)$	$\mathbb{I}_S(X) = 0$ if $X \in S$, and $\mathbb{I}_S(X) = \infty$ otherwise
dom J	the domain of function J

III. Preliminaries

NMF [17] is a dimensionality reduction and feature extraction technique widely used in machine learning. Since NMF can be viewed as a single-layer learning process, it cannot discover the hierarchical features of data. To remedy this defect, [6] proposed the deep NMF (DNMF) method, which extends NMF to a multi-layer structure and provides an easily interpretable representation. Given a nonnegative matrix $X \in \mathbb{R}_+^{m_1 \times R}$, the DNMF method can be expressed as follows.

$$\begin{aligned} &\min_{\{A_i\}_{i=1}^N} \frac{1}{2}\|X - \prod_{i=1}^N A_i\|_F^2, \\ &s.t.\ A_i \geq 0,\ i \in [N], \end{aligned} \tag{2}$$

where each factor matrix $A_i \in \mathbb{R}_+^{m_i \times m_{i+1}}$, $m_{N+1} = R$.

Adding the graph Laplacian regularization into the model is an effective way to improve the quality of extracted features since graph Laplacian regularization can preserve the geometric structure information of data (e.g., the distance or similarity between images). Therefore, [18] proposed the graph regularized DNMF (GDNMF) method to improve the quality of features extracted by DNMF, which can be expressed as follows.

$$\begin{aligned} &\min_{\{A_i\}_{i=1}^N} \frac{1}{2}\|X - \prod_{i=1}^N A_i\|_F^2 + \frac{\alpha}{2}\mathrm{Tr}((A_N)^T L A_N), \\ &s.t.\ A_i \geq 0,\ i \in [N], \end{aligned} \tag{3}$$

where $\alpha \in \mathbb{R}_+$, L is the graph Laplacian matrix.

Similarly, enforcing the solution to be sparse can remove redundant features and retain relevant features with respect to the underlying essential structure of data, improving the quality of extracted features [11]. A straightforward and common scheme for achieving the sparsity of the solution is to use element-wise sparse norm regularization (e.g., ℓ_1-norm regularization) to individually evaluate the weights of each element in the solution and zero out unimportant elements based on their weights. Therefore, the ℓ_1-norm regularized sparse DNMF (ℓ_1-SDNMF) method [14] has been proposed, which employs ℓ_1-norm regularization to promote the element-wise sparsity of factor matrices. ℓ_1-SDNMF can be expressed as follows.

$$\begin{aligned} &\min_{\{A_i\}_{i=1}^N} \frac{1}{2}\|X - \prod_{i=1}^N A_i\|_F^2 + \sum_{i=1}^N \lambda_i \|A_i\|_1, \\ &s.t.\ A_i \geq 0,\ i \in [N], \end{aligned} \tag{4}$$

where $\lambda_i \in \mathbb{R}_+$. Considering that both the element-wise sparsity of the solution and graph Laplacian regularization can enhance feature extraction capability, [15] proposed a DNMF method that combines graph regularization with ℓ_1-norm regularization applied solely to a single factor matrix. However, this method is limited to promoting the element-wise sparsity of a single factor matrix without considering the sparsity of other factor matrices.

More fundamentally, as outlined in the Introduction, most existing deep matrix factorization methods (including all the DNMF methods mentioned in this section) promote the element-wise sparsity of factor matrices via corresponding element-wise sparse regularizations. Although this element-wise sparsity is straightforward and commonly used, it only individually evaluates the importance of each element of factor matrices and zeros out unimportant ones, neglecting both the interactions and dependencies between different features. Additionally, this element-wise sparsity may zero out the contributions of the same feature across different dimensions individually, thus failing to completely remove or entirely preserve latent features and leading to the destruction of the inherent structure of features [11], [16]. Therefore, the above drawbacks reduce the feature extraction capability and interpretability of such deep matrix factorization methods.

IV. The Proposed $\ell_{2,1}$-SSGDNMF Method

To overcome the aforementioned drawbacks, we integrate the $\ell_{2,1}$-norm into DNMF to promote the simultaneous selection or removal of entire feature groups while incorporating graph Laplacian regularization to preserve the geometric structure information of data, thereby integrating manifold learning and group sparsity into DNMF and enhancing both the feature extraction capability and interpretability. This leads to our $\ell_{2,1}$-SSGDNMF method as follows.

$$\begin{aligned} &\min_{\{A_i\}_{i=1}^N} \frac{1}{2}\|X - \prod_{i=1}^N A_i\|_F^2 + \frac{\alpha}{2}\mathrm{Tr}((A_N)^T L A_N) + \sum_{i=1}^N \lambda_i \|A_i\|_{2,1}, \\ &s.t.\ A_i \geq 0,\ i \in [N], \end{aligned} \tag{5}$$

where $A_i \in \mathbb{R}_+^{m_i \times m_{i+1}}$, $\lambda_i \in \mathbb{R}_+$, $\alpha \in \mathbb{R}_+$, $L = D - W$ and D is the diagonal matrix, W is the weight matrix, and $d_{ii} = \sum_{j=1}^R w_{ij}$ (d_{ii} is the (i,i)-th component of D, and w_{ij} is the (i,j)-th component of W). We use the heat kernel weight to construct the weight matrix W, as detailed in Section

II and Eq. (1). We consider the rows of $\prod_{i=1}^{N-1} A_i$ of Eq. (5) as cluster centroids and R as the number of samples, then the rows of A_N can be interpreted as cluster indicators learned with parts-based features, that is, A_N can be viewed as the low-dimensional representation of X. Thus only the feature weight matrices are subject to sparsity norm regularization, i.e., we always set $\lambda_N \equiv 0$ in Eq. (5).

Remark 1. *(i) If we simply introduce ℓ_1-norm regularization into DNMF to promote the element-wise sparsity of factor matrices and add the graph Laplacian regularization into DNMF, we can obtain the ℓ_1-SGDNMF method, which can be expressed as follows.*

$$\min_{\{A_i\}_{i=1}^N} \frac{1}{2}\|X - \prod_{i=1}^{N} A_i\|_F^2 + \frac{\alpha}{2}\operatorname{Tr}((A_N)^T L A_N) + \sum_{i=1}^{N} \lambda_i \|A_i\|_1, \quad s.t.\ A_i \geq 0,\ i \in [N], \tag{6}$$

where $A_i \in \mathbb{R}_+^{m_i \times m_{i+1}}$, $\lambda_i \in \mathbb{R}_+$ $(\forall i \in [N])$, the definition of L is the same as that of L in Eq. (5). Similarly, we also prove that ℓ_1-SGDNMF can be solved by our proposed algorithm, and that the monotonic convergence and global convergence of our algorithm for solving ℓ_1-SGDNMF are still guaranteed (See Remark 2 of Supplementary Material).

(ii) To test the effectiveness of $\ell_{2,1}$-norm in $\ell_{2,1}$-SSGDNMF, ℓ_1-SGDNMF will also serve as a baseline method and be compared with $\ell_{2,1}$-SSGDNMF in the experiment section later.

V. The Proposed APL Algorithm

In this section, we present the technical details of the proposed Alternating Proximal Linearized (APL) algorithm for solving the $\ell_{2,1}$-SSGDNMF. We also prove the monotonic convergence and global convergence of the APL algorithm and analyze the per-iteration complexity.

A. Optimization for $\ell_{2,1}$-SSGDNMF

For Eq. (5), we first rewrite it into the following form.

$$\min_{A_{(N)}} J(A_{(N)}) = H(A_{(N)}) + \sum_{i=1}^{N} F_i(A_i), \tag{7}$$

where $H(A_{(N)}) = \frac{1}{2}\|X - \prod_{i=1}^{N} A_i\|_F^2 + \frac{\alpha}{2}\operatorname{Tr}((A_N)^T L A_N)$, $F_i(A_i) = \lambda_i \|A_i\|_{2,1} + \mathbb{I}_{A_i \geq 0}(A_i)$. Then $\nabla_{A_i} H(\{A_i\}_{i=1}^N)$ is

$$\nabla_{A_i} H(A_{(N)}) = \begin{cases} (B_i)^T(B_i A_i C_i - X)(C_i)^T + \alpha L A_i, & \text{if } i = N, \\ (B_i)^T(B_i A_i C_i - X)(C_i)^T, & \text{else}, \end{cases} \tag{8}$$

where $B_i = \prod_{j=1}^{i-1} A_j$, $C_i = \prod_{j=i+1}^{N} A_j$. Therefore, from Eq. (8), we have the following theorem.

Theorem 1. *The Lipschitz constant of $\nabla_{A_i} H(\{A_i\}_{i=1}^N)$ can be presented as follows.*

$$L_{\nabla_{A_i} H} = \|(\prod_{j=1}^{i-1} A_j)^T \prod_{j=1}^{i-1} A_j\|_F \| \prod_{j=i+1}^{N} A_j (\prod_{j=i+1}^{N} A_j)^T\|_F.$$

The specific proof process of Theorem 1 can be found in the Supplementary Material.

Obviously, Eq. (7) is a nonconvex and nonsmooth problem. To this end, we employ the proximal linearization operator [19], [20] to solve Eq. (7). Specifically, during the iteration of our algorithm, the proximal linearization operator for solving Eq. (7) can be defined as

$$\begin{aligned} A_i^{k+1} &\in prox_{\sigma_i^k F_i}(U_i^k) \\ &= \arg\min_A \left\{ \frac{1}{2\sigma_i^k}\|A - U_i^k\|_F^2 + F_i(A) \right\} \\ &= \arg\min_A \left\{ \frac{1}{2\sigma_i^k}\|A - U_i^k\|_F^2 + \lambda_i \|A\|_{2,1} : A \geq 0 \right\}, \end{aligned} \tag{9}$$

where $\sigma_i^k \in (0, \frac{1}{L_{\nabla_{A_i^k} H}})$, $U_i^k \in \mathbb{R}^{m_i \times m_{i+1}}$. As a result, by using Eq. (8), Theorem 1 and Eq. (9) as the foundation, we propose the APL algorithm for directly solving the $\ell_{2,1}$-SSGDNMF (Algorithm 1).

Algorithm 1: APL: Alternating Proximal Linearized algorithm for solving $\ell_{2,1}$-SSGDNMF

Input: $\{A_i^1\}_{i=1}^N = \{A_i^0\}_{i=1}^N \in \operatorname{dom} J$, $\beta_1 = 0$, $k = 1$, $t_1 = 1$, $\gamma = 1.01$, $\rho_t > 0$.

Output: $\{A_i^k\}_{i=1}^N$.

repeat
- $\beta_k = \frac{t_k - 1}{t_{k+1}}$, $Y_{(N)}^k = A_{(N)}^k + \beta_k(A_{(N)}^k - A_{(N)}^{k-1})$.
- **for** $i = 1$ *to* N **do**
  - $h_i(Y_i^k) = H(\{A_j^{k+1}\}_{j=1}^{i-1}, Y_i^k, \{A_j^k\}_{j=i+1}^N)$,
  - $\sigma_i^k = \frac{1}{\gamma L_{\nabla_{A_i^k} H}}$, $U_i^k = Y_i^k - \sigma_i^k \nabla_{A_i} h_i(Y_i^k)$,
  - $A_i^{k+1} \in prox_{\sigma_i^k F_i}(U_i^k)$,
- **end**
- **if** $J(A_{(N)}^{k+1}) > J(A_{(N)}^k) - \rho_t \|A_{(N)}^{k+1} - Y_{(N)}^k\|_F^2$ **then**
  - $\beta_k = 0$, re-update $A_{(N)}^{k+1}$ through step 2 to 7.
- **end**
- $t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$, $k = k + 1$.

until $\frac{\|A_{(N)}^k - A_{(N)}^{k-1}\|_F}{\|A_{(N)}^{k-1}\|_F} < \epsilon$;

B. Convergence analysis

To simplify the proof in this section, we define the sequence generated by Algorithm 1 as $z^k = A_{(N)}^k$, $c^k = Y_{(N)}^k$. The specific proof process of this section is provided in the Supplementary Material, which is available at our code link https://github.com/Weifeng-Yang/SSGDNMF.

1) Monotonic convergence: We first prove the monotonic convergence of the Algorithm 1.

Theorem 2. *Let $\{z^k\}_{k \in \mathbb{N}}$ be a sequence generated by Algorithm 1.*
(i) $J(A_{(N)}^{k+1})$ is nonincreasing and in particular

$$J(A_{(N)}^{k+1}) \leq J(A_{(N)}^k) - \rho \|z^{k+1} - c^k\|_F^2,$$

where ρ is a positive constant.
(ii) We have $\lim_{k\to\infty} \|z^{k+1} - c^k\|_F = 0$.

2) Global convergence: Let z' be the derived set of $\{z^k\}_{k\in\mathbb{N}}$, we have the following theorem.

Theorem 3. *(i) Let $\{z^k\}_{k\in\mathbb{N}}$ be a sequence generated by Algorithm 1, then z' is a nonempty compact set.*
(ii) J is constant on z'.
(iii) $z' \subseteq crit\ J$, where $crit\ J$ is the critical point set of J, i.e., $0 \in \partial J(z^*)$ $(J(z^*) = \lim_{k\to\infty} J(z^k))$.

Next, we prove that the Algorithm 1 has global convergence.

Theorem 4. *The sequence $\{z^k\}_{k\in\mathbb{N}}$ generated by Algorithm 1 converges, i.e., $\lim_{k\to\infty} \|z^{k+p} - z^k\|_F = 0$ $(\forall p \in \mathbb{N})$.*

C. Computational complexity analysis

We analyze the computational complexity of Algorithm 1.

Theorem 5. *Assuming that the factor matrices $A_i \in \mathbb{R}^{m_i\times m_{i+1}}$ and original matrix $X \in \mathbb{R}_+^{m_1\times R}$ (where $m_{N+1} = R$), then the time complexity of Algorithm 1 in each iteration is approximately estimated as $\mathcal{O}(\sum_{i=1}^{N-2}(R\sum_{j=i+1}^{N-1} m_j m_{j+1} + m_1) + \sum_{i=3}^{N}(\sum_{j=1}^{i-2} m_j m_{j+1}) + \sum_{i=1}^{N}(m_i^2 m_{i+1} + m_i m_{i+1}^2 + m_1 m_i R + m_i m_{i+1} R))$.*

VI. Experiments

In this section, we perform the experiments on six real-world datasets (Orlraws10P[1], Pixraw10P[1], COIL20[2], ORL[1], PNW[3], OrganSMNIST[4]). Since the scales of the PNW dataset are too large, we only randomly select 10 samples from each class of the PNW dataset. For all the datasets, we normalize the value of each feature to the range of 0 to 1. The details of these datasets are summarized in Table II. All the experiments are implemented in this configuration: Intel(R) Core(TM) i7-10850H CPU @ 2.70GHz, RAM 16.0 GB, Matlab 2021b. Our code is available at at our code link https://github.com/Weifeng-Yang/SSGDNMF.

Table II: Detailed statistics of the datasets.

Dataset	Sample size	Feature size	Classes
Orlraws10P	100	10304	10
Pixraw10P	100	10000	10
COIL20	1440	1024	20
ORL	400	1024	40
PNW	50	45000	5
OrganSMNIST	1100	784	11

A. Comparative methods

We first propose two transformation forms of the $\ell_{2,1}$-SSGDNMF to test the effectiveness of the $\ell_{2,1}$-norm regularization in $\ell_{2,1}$-SSGDNMF.

1) ℓ_1-SGDNMF: Sparse and graph-regularized DNMF with ℓ_1-norm regularization method, i.e., Eq. (6).
2) $\ell_{2,1}$-SSDNMF: This method corresponds exactly to $\ell_{2,1}$-SSGDNMF with $\alpha = 0$.

For the parameter settings of these transformation forms of $\ell_{2,1}$-SSGDNMF, we set λ_i for $\ell_{2,1}$-SSDNMF to be the same as in $\ell_{2,1}$-SSGDNMF. We also set both α and λ_i for ℓ_1-SGDNMF to be the same as in $\ell_{2,1}$-SSGDNMF. Additionally, we set the number of latent classes as R and set $\epsilon = 1e-6$ for our algorithm in this experiment.

Furthermore, we compare $\ell_{2,1}$-SSGDNMF and its transformation forms with the following methods for clustering.

1) $\ell_{2,1}$-NMF [21]: Nonnegative matrix factorization (NMF) with the $\ell_{2,1}$-norm. This method incorporates $\ell_{2,1}$-norm into NMF to promote group sparsity of factor matrices.
2) $\ell_{2,1}$-GSNMF [22]: Graph Regularized Sparse $\ell_{2,1}$ semi-nonnegative matrix factorization. This method introduces graph regularization into $\ell_{2,1}$-NMF.
3) GDNMF [18]: Graph regularized DNMF, i.e., Eq. (3).
4) ℓ_1-SDNMF [14]: ℓ_1 regularized sparse DNMF (ℓ_1-SDNMF) method, i.e., Eq. (4).
5) DHGLpSNMF [9]: Deep hypergraph regularized L_p smooth semi-nonnegative matrix factorization method.

All parameters for the compared methods are set to the optimal values recommended in their respective papers. All the methods mentioned above use the K-means method to cluster the extracted low-dimensional representation. To mitigate the issue of local convergence, each K-means run is repeated 10 times with different initial points, and the average result of these 10 times is then taken as the result of that K-means run. The elements of the initial points for all methods are generated by the standard uniform distribution. We adopt three widely used metrics *accuracy* (ACC), *adjusted rand index* (ARI) and *normalized mutual information* (NMI) to quantitatively describe the clustering performance [23], [24]. The larger the values of these metrics, the better the clustering performance.

B. Experiment 1: Clustering performance evaluations

We evaluate the clustering performance of all methods on six real-world datasets described in Table II. We set $N = 3$, $m_2 = 100$, $m_3 = \lceil \frac{R}{4} \rceil$, and $\alpha = 1$ as well as $\lambda_i = 1$ $(\forall i \in [N-1])$ across all datasets. Table III shows the clustering performance results (average and standard deviation) for the methods over 10 independent runs with different initial points. From Table III, we have the following observations.

1) ℓ_1-SGDNMF outperforms other methods except $\ell_{2,1}$-SSGDNMF and $\ell_{2,1}$-SSDNMF on these datasets. This shows that combining the sparsity of factor matrices with graph regularization into DNMF can indeed improve the quality of extracted features and interpretability.
2) $\ell_{2,1}$-SSDNMF outperforms both $\ell_{2,1}$-NMF and ℓ_1-SDNMF on most datasets, and $\ell_{2,1}$-SSGDNMF achieves the best clustering performance on these datasets. These results fully demonstrate the effectiveness of our proposed method and show that utilizing the $\ell_{2,1}$-norm to

[1] https://jundongl.github.io/scikit-feature/datasets.html
[2] http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html
[3] https://github.com/niyiyu/PNW-ML
[4] https://github.com/MedMNIST/MedMNIST

Table III: Average and standard deviation results by all methods on datasets. The best performance is highlighted in bold.

Method \ Dataset	Orlraws10P			Pixraw10P		
	ACC	ARI	NMI	ACC	ARI	NMI
$\ell_{2,1}$-NMF	0.506 ± 0.028	0.287 ± 0.043	0.540 ± 0.034	0.580 ± 0.021	0.404 ± 0.030	0.649 ± 0.023
$\ell_{2,1}$-GSNMF	0.535 ± 0.009	0.391 ± 0.003	0.502 ± 0.007	0.659 ± 0.030	0.515 ± 0.038	0.732 ± 0.025
GDNMF	0.651 ± 0.010	0.478 ± 0.017	0.712 ± 0.021	0.665 ± 0.029	0.522 ± 0.037	0.747 ± 0.025
ℓ_1-SDNMF	0.683 ± 0.045	0.539 ± 0.074	0.750 ± 0.053	0.701 ± 0.054	0.598 ± 0.062	0.758 ± 0.033
DHGLpSNMF	0.689 ± 0.004	0.549 ± 0.004	0.754 ± 0.007	0.680 ± 0.010	0.543 ± 0.009	0.751 ± 0.016
ℓ_1-SGDNMF	0.709 ± 0.030	0.586 ± 0.035	0.769 ± 0.019	0.721 ± 0.016	0.609 ± 0.026	0.800 ± 0.016
$\ell_{2,1}$-SSDNMF	0.698 ± 0.022	0.567 ± 0.020	0.764 ± 0.010	0.731 ± 0.015	0.605 ± 0.026	0.796 ± 0.015
$\ell_{2,1}$-SSGDNMF	**0.721 ± 0.010**	**0.605 ± 0.014**	**0.793 ± 0.007**	**0.755 ± 0.027**	**0.651 ± 0.033**	**0.854 ± 0.019**

Method \ Dataset	COIL20			ORL		
	ACC	ARI	NMI	ACC	ARI	NMI
$\ell_{2,1}$-NMF	0.589 ± 0.025	0.501 ± 0.017	0.718 ± 0.014	0.503 ± 0.022	0.347 ± 0.025	0.716 ± 0.014
$\ell_{2,1}$-GSNMF	0.609 ± 0.020	0.539 ± 0.020	0.744 ± 0.011	0.542 ± 0.015	0.353 ± 0.025	0.734 ± 0.012
GDNMF	0.524 ± 0.019	0.413 ± 0.020	0.635 ± 0.016	0.418 ± 0.021	0.313 ± 0.024	0.633 ± 0.017
ℓ_1-SDNMF	0.519 ± 0.018	0.428 ± 0.014	0.642 ± 0.014	0.468 ± 0.027	0.321 ± 0.029	0.693 ± 0.017
DHGLpSNMF	0.595 ± 0.011	0.468 ± 0.015	0.688 ± 0.008	0.528 ± 0.004	0.341 ± 0.001	0.677 ± 0.007
ℓ_1-SGDNMF	0.583 ± 0.014	0.452 ± 0.016	0.676 ± 0.013	0.549 ± 0.061	0.378 ± 0.104	0.742 ± 0.062
$\ell_{2,1}$-SSDNMF	0.617 ± 0.011	0.546 ± 0.016	0.735 ± 0.016	0.555 ± 0.012	0.404 ± 0.014	0.759 ± 0.007
$\ell_{2,1}$-SSGDNMF	**0.624 ± 0.010**	**0.553 ± 0.005**	**0.746 ± 0.005**	**0.589 ± 0.010**	**0.452 ± 0.012**	**0.786 ± 0.004**

Method \ Dataset	PNW			OrganSMNIST		
	ACC	ARI	NMI	ACC	ARI	NMI
$\ell_{2,1}$-NMF	0.342 ± 0.034	0.012 ± 0.037	0.135 ± 0.028	0.278 ± 0.008	0.113 ± 0.005	0.270 ± 0.006
$\ell_{2,1}$-GSNMF	0.352 ± 0.033	0.013 ± 0.032	0.139 ± 0.039	0.295 ± 0.008	0.118 ± 0.004	0.263 ± 0.006
GDNMF	0.421 ± 0.011	0.104 ± 0.011	0.290 ± 0.017	0.249 ± 0.006	0.080 ± 0.005	0.186 ± 0.009
ℓ_1-SDNMF	0.424 ± 0.012	0.114 ± 0.009	0.321 ± 0.025	0.229 ± 0.031	0.074 ± 0.027	0.202 ± 0.047
DHGLpSNMF	0.448 ± 0.015	0.121 ± 0.012	0.295 ± 0.016	0.293 ± 0.004	0.101 ± 0.005	0.261 ± 0.001
ℓ_1-SGDNMF	0.451 ± 0.008	0.123 ± 0.006	0.281 ± 0.012	0.299 ± 0.009	0.119 ± 0.007	0.272 ± 0.012
$\ell_{2,1}$-SSDNMF	0.428 ± 0.017	0.118 ± 0.007	**0.330 ± 0.019**	0.240 ± 0.003	0.075 ± 0.015	0.227 ± 0.012
$\ell_{2,1}$-SSGDNMF	**0.467 ± 0.012**	**0.141 ± 0.008**	0.308 ± 0.003	**0.333 ± 0.003**	**0.131 ± 0.002**	**0.316 ± 0.006**

promote the group sparsity of the factor matrices in DNMF can indeed enhance feature extraction capability and interpretability.

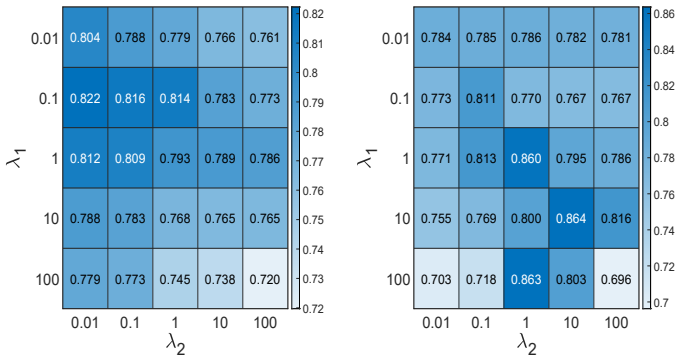
C. Experiment 2: Parameter sensitivity analysis

Next, we analyze the impact of various parameters in $\ell_{2,1}$-SSGDNMF on clustering performance. Specifically, we report the clustering performance under the following three parameter settings on the orlraws10P and Pixraw10P datasets.

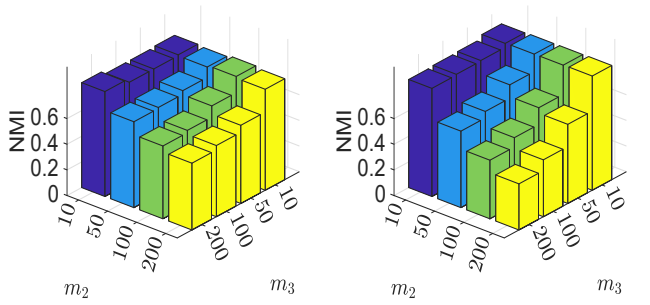
1) Fix $\alpha = 1$, $N = 3$, $m_2 = 100$, $m_3 = \lceil \frac{R}{4} \rceil$ and then alternately vary one of the sparsity level parameters λ_i ($\forall i \in [N-1]$), and record the clustering performance under all possible parameter combinations, as shown in Figure 1(a).
2) Fix $N = 3$, $\lambda_i = 1$ ($i \in [N-1]$) and $\alpha = 1$, then alternately vary one of the rank m_{i+1} ($\forall i \in [N-1]$), and record the clustering performance under all possible parameter combinations, as shown in Figure 1(b).
3) Fix $N = 3$, $m_2 = 100$, $m_3 = \lceil \frac{R}{4} \rceil$ and $\lambda_i = 1$ ($i \in [N-1]$), then vary the graph regularization parameter α and record the clustering performance, as shown in Figure 1(c).

For the sake of space, we only illustrate results on the orlraws10P and Pixraw10P datasets. From Figure 1, we have the following observations.

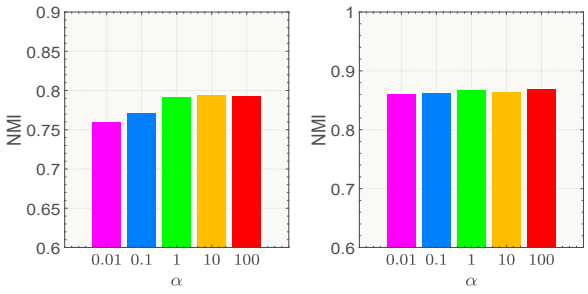
1) For the sparsity level parameters, the clustering performance of $\ell_{2,1}$-SSGDNMF on orlraws10P dataset initially improves and then declines as λ_1 increases while it shows a negative correlation with λ_2. Besides, the clustering performance of $\ell_{2,1}$-SSGDNMF on the Pixraw10P dataset initially improves and then declines as λ_2 increases while it is relatively sensitive to λ_1.
2) For the rank, the clustering performance of $\ell_{2,1}$-SSGDNMF on the orlraws10P dataset is relatively stable for small values of m_2 or m_3, but shows a clear negative correlation with both m_2 and m_3 as they increase further. Besides, the clustering performance of $\ell_{2,1}$-SSGDNMF on the Pixraw10P dataset always shows a negative correlation with both m_2 and m_3.
3) For the graph regularization parameter, the clustering performance of $\ell_{2,1}$-SSGDNMF on the orlraws10P dataset tends to stabilize when $\alpha \geq 1$, but the clustering performance of $\ell_{2,1}$-SSGDNMF on the Pixraw10P dataset is relatively insensitive to α.



(a) Clustering NMI with respect to the parameters λ_1 and λ_2 on the orlraws10P (left) and Pixraw10P (right) datasets, respectively.



(b) Clustering NMI with respect to the parameters m_2 and m_3 on the orlraws10P (left) and Pixraw10P (right) datasets, respectively.



(c) Clustering NMI with respect to the parameter α on the orlraws10P (left) and Pixraw10P (right) datasets, respectively.

Figure 1: Parameter sensitivity analysis of $\ell_{2,1}$-SSGDNMF on the orlraws10P and Pixraw10P datasets.

D. Experiment 3: Convergence analysis

We also evaluate the convergence performance of our proposed APL algorithm (Algorithm 1) for solving both the ℓ_1-SGDNMF and $\ell_{2,1}$-SSGDNMF. Figure 2 plots how the objective function value changes with respect to time for the APL algorithm on several datasets. From Figure 2, we observe that the objective function value monotonically decreases to a stationary point, thereby validating the effectiveness and convergence of the APL algorithm for solving both the ℓ_1-SGDNMF and $\ell_{2,1}$-SSGDNMF.

VII. Conclusion

In this paper, we proposed a novel structured-sparse and graph-regularized deep nonnegative matrix factorization with

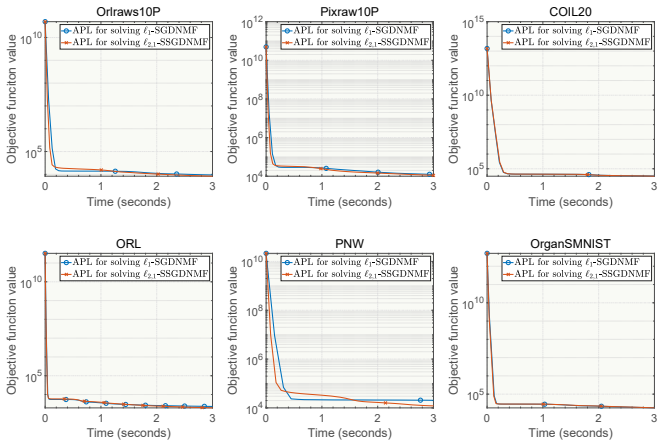


Figure 2: Convergence curves of the APL algorithm for solving both the ℓ_1-SGDNMF and $\ell_{2,1}$-SSGDNMF on six datasets.

$\ell_{2,1}$-norm regularization ($\ell_{2,1}$-SSGDNMF) method. Unlike existing DNMF methods that use element-wise sparse norm regularization to promote the element-wise sparsity of factor matrices, our method employs the $\ell_{2,1}$-norm to select or discard entire feature groups, thereby explicitly imposing the group sparsity on factor matrices and enhancing both the feature extraction capability and interpretability. Moreover, $\ell_{2,1}$-SSGDNMF incorporates graph regularization to preserve the geometric structure information of data, thereby integrating manifold learning and group sparsity into DNMF. Although the $\ell_{2,1}$-SSGDNMF is a nonconvex and nonsmooth problem, we proposed the APL algorithm to solve $\ell_{2,1}$-SSGDNMF. Furthermore, we established the global convergence of our proposed algorithm and analyzed the per-iteration complexity. Experimental results on six real-world benchmark datasets demonstrated that our method outperforms several state-of-the-art methods for clustering. Additionally, we investigated the parameter sensitivity of $\ell_{2,1}$-SSGDNMF. We also presented the convergence behavior of the APL algorithm for solving both ℓ_1-SGDNMF and $\ell_{2,1}$-SSGDNMF, validating the effectiveness and convergence of our proposed algorithm.

References

[1] W. Yang and W. Min, "Globally convergent accelerated algorithms for multilinear sparse logistic regression with ℓ_0-constraints," in *Int. Conf. Intell. Comput.* Springer, 2024, pp. 88–99.

[2] S. Lv and Y. Peng, "Dtpp:an efficient depthwise separable tcn for seismic phase picking," *Artif. Intell. Geosci.*, vol. 7, no. 1, p. 100189, 2026.

[3] Z. Liu, F. Zhu, H. Xiong, X. Chen, D. Pelusi, and A. V. Vasilakos, "Graph regularized discriminative nonnegative matrix factorization," *Eng. Appl. Artif. Intel.*, vol. 139, p. 109629, 2025.

[4] Z. He, Y. Lin, Z. Lin, and C. Wang, "Multi-label feature selection via similarity constraints with non-negative matrix factorization," *Knowl-based. Syst.*, vol. 297, p. 111948, 2024.

[5] Z. Dou, N. Peng, W. Hou, X. Xie, and X. Ma, "Learning multi-level topology representation for multi-view clustering with deep non-negative matrix factorization," *Neural Networks*, vol. 182, p. 106856, 2025.

[6] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. W. Schuller, "A deep matrix factorization method for learning attribute representations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 3, pp. 417–429, 2016.

[7] D. Wang, T. Li, P. Deng, F. Zhang, W. Huang, P. Zhang, and J. Liu, "A generalized deep learning clustering algorithm based on non-negative matrix factorization," *Acm. T. Knowl. Discov. D.*, vol. 17, no. 7, pp. 1–20, 2023.

[8] K. Luong, R. Nayak, T. Balasubramaniam, and M. A. Bashar, "Multi-layer manifold learning for deep non-negative matrix factorization-based multi-view clustering," *Pattern. Recogn.*, vol. 131, p. 108815, 2022.

[9] S. Li, L. Wang, and M. Bai, "Deep hypergraph regularized l_p smooth semi-nonnegative matrix factorization for hierarchical clustering analysis," *J. King Saud Univ., Comp. Info. Sci.*, vol. 37, no. 9, p. 303, 2025.

[10] A. Hajiveiseh, S. A. Seyedi, and F. A. Tab, "Deep asymmetric nonnegative matrix factorization for graph clustering," *Pattern. Recogn.*, vol. 148, p. 110179, 2024.

[11] X. Li, Y. Wang, and R. Ruiz, "A survey on sparse learning models for feature selection," *IEEE. T. Cybernetics.*, vol. 52, no. 3, pp. 1642–1660, 2020.

[12] W.-S. Chen, Q. Zeng, and B. Pan, "A survey of deep nonnegative matrix factorization," *Neurocomputing*, vol. 491, pp. 305–320, 2022.

[13] H. Fang, A. Li, H. Xu, and T. Wang, "Sparsity-constrained deep nonnegative matrix factorization for hyperspectral unmixing," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 7, pp. 1105–1109, 2018.

[14] Z. Guo and S. Zhang, "Sparse deep nonnegative matrix factorization," *Big. Data. Min. Anal.*, vol. 3, no. 1, pp. 13–28, 2019.

[15] W. Guo, "Sparse dual graph-regularized deep nonnegative matrix factorization for image clustering," *IEEE Access*, vol. 9, pp. 39 926–39 938, 2021.

[16] H. Yan, J. Yang, and J. Yang, "Robust joint feature weights learning framework," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 5, pp. 1327–1339, 2016.

[17] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[18] X. Zeng, S. Qu, and Z. Wu, "Graph regularized deep semi-nonnegative matrix factorization for clustering," in *Eighth International Conference on Digital Image Processing (ICDIP 2016)*, vol. 10033. SPIE, 2016, pp. 1126–1130.

[19] W. Yang and W. Min, "An accelerated block proximal framework with adaptive momentum for nonconvex and nonsmooth optimization," *arXiv preprint arXiv:2308.12126*, 2023.

[20] W. Yang, T. Xu, and W. Min, "Sparse nonnegative cp decomposition with graph regularization and ℓ_0-constraints for clustering," *Chinese. J. Electron.*, vol. 34, no. 5, pp. 1641–1651, 2025.

[21] X. Jin, J. Miao, Q. Wang, G. Geng, and K. Huang, "Sparse matrix factorization with $l_{2,1}$ norm for matrix completion," *Pattern. Recogn.*, vol. 127, p. 108655, 2022.

[22] A. Rhodes, B. Jiang, and J. Jiang, "Graph regularized sparse $l_{2,1}$ semi-nonnegative matrix factorization for data reduction," *Numer. Linear Algebra Appl.*, vol. 32, no. 1, p. e2598, 2025.

[23] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, 2005.

[24] Y. Weifeng and M. Wenwen, "Graph regularized sparse nonnegative tucker decomposition with ℓ_0-constraints for unsupervised learning," *Chinese. J. Electron*, 2025.