Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 33(12): 7841–7865.
DOI: 10.3934/era.2025346
Received: 23 July 2025
Revised: 15 November 2025
Accepted: 05 December 2025
Published: 25 December 2025

Research article

An alternating minimization algorithm for sparse convolutive non-negative matrix factorization with ℓ_1-norm

Yijia Zhou*

Dalian Neusoft University of Information, Dalian 116086, China

* **Correspondence:** Email: zhouyijia@neusoft.edu.cn.

Abstract: Convolutive non-negative matrix factorization has been a dominant analytical technique for deriving interpretable insights from data in speech processing, image analysis, data mining, biomedicine, and other fields. In this paper, a sparse convolutive non-negative matrix factorization model was introduced by incorporating an ℓ_1 regularization on representation matrices. This enhancement not only preserved the inherent characteristics of convolutive non-negative matrix factorization, but also promoted sparse data representation, thereby facilitating more efficient data storage and analysis. An alternating minimization algorithm for the presented model was proposed by integrating the alternating direction method of multipliers with the accelerated iterative shrinkage-thresholding algorithm. In addition, a convergence result was presented that the convergence point of the algorithm necessarily constitutes a stable point of the problem. Experimental results showed that the proposed algorithm yielded sparser solutions for synthetic data designed to simulate sparse representation scenarios, and achieved practical applicability in speech dataset, validating its potential for real-world signal processing tasks.

Keywords: convolutive non-negative matrix factorization; alternating direction method of multipliers; convolutive basis; ℓ_1 regularization; sparse optimization

1. Introduction

In the era of global informatization, the scale of complex systems has exhibited exponential growth. Non-negative structured data, encompassing digital photographs, audio signals, and beyond, are inherently formulated as non-negative high-dimensional matrices, however, this representation underscores their physical interpretability in real-world applications. High-dimensional data matrices are often subjected to decomposition to fulfill critical tasks including dimensionality reduction and clustering. Independent component analysis (ICA) [1], principal component analysis (PCA) [2], and singular value decomposition (SVD) remain among the most widely used matrix decomposition techniques. However, these methods are often limited in practical applications by the lack of physical interpret-ability in negative elements, thus driving growing attention to non-negative matrix factorization (NMF) (see [3, 4], for example).

The typical formulation of the standard NMF given a non-negative data matrix $Z \in R_{\geq 0}^{F\times N}$ is,

$$\min_{Y,A} D(Z \mid YA) \quad \text{s.t.} \quad \mathbf{Y} \in \mathbb{R}_{\geq 0}^{F\times K}, \quad \mathbf{A} \in \mathbb{R}_{\geq 0}^{K\times N}, \tag{1.1}$$

where $D(\cdot \mid \cdot)$ is a divergence function that represents the distance between observations Z and YA. A comprehensive family of divergence functions is formally referred to as the β-divergence (see (1.2)),

$$D_\beta(\mu \mid \nu) = \begin{cases} \frac{\mu^\beta}{\beta(\beta-1)} + \frac{\nu^\beta}{\beta} - \frac{\mu\nu^{\beta-1}}{\beta-1}, & \beta \in \mathbb{R}\backslash\{0, 1\} \\ \mu \log \frac{\mu}{\nu} - \mu + \nu, & \beta = 1 \\ \frac{\mu}{\nu} - \log \frac{\mu}{\nu} - 1, & \beta = 0 \end{cases}, \tag{1.2}$$

to quantify the distance. The β-divergence function with $\beta = 0, \beta = 1$, and $\beta = 2$ correspond to Itakura-Saito divergence (ISD), generalized Kullback-Leiberler divergence (KLD) and Euclidean distance (ELD), respectively [5]. In this paper, we will only focus on the cost function based on Euclidean distance. Specifically, we propose an alternating minimization approach for solving NMF-related problems. It is worth noting that analogous algorithms can be derived for other models or cost functions.

In the prestigious journal Nature in 1999, scientists Lee and Seung proposed the NMF problem [6], which addresses objective functions based on Euclidean distance and Kullback-Leibler divergence [5]. The original work also provided updating rules and convergence proofs for the algorithm. As a matrix decomposition technique, NMF is renowned for its straightforward mathematical structure and effective factorization performance in various applications [7]. A high-dimensional data matrix is decomposed into two low-dimensional non-negative matrices, a process that enables the extraction of interpretable latent features from complex datasets. Importantly, the non-negative constraint in NMF induces inherent sparsity, meaning the factorized matrices contain numerous zero elements. This property enables NMF to not only represent the original data parsimoniously, but also reduce storage requirements, leveraging sparse representations for efficient data modeling [8]. However, this inherent sparsity often proves insufficient in certain practical applications. To address this, many studies focus on inducing an ℓ_0, ℓ_1, and other regularization terms and sparse constraints to enhance the sparsity of the factorized matrices [8–11]. Sparse representation refers to a theoretical framework for representing feature vectors over an overcomplete basis, wherein each feature vector is expressed as a linear combination of a minimal number of basis elements to ensure both representation accuracy and efficiency. This approach leverages the sparsity-inducing property of such combinations to facilitate dimensionality reduction and feature extraction in complex data modeling [12–17].

Traditional approaches often overlook the latent dependencies between speech frames and fail to accurately model their temporal dynamics. Convolutive NMF (CNMF) has emerged as a transformative extension of standard NMF by employing a set of time-shifted basis matrices $Y(t), t \in (0, T-1)$, addressing critical limitations in modeling sequential or spatially structured data [18]. For instance, CNMF effectively preserves inter-frame correlations in speech signals [19, 20]. This capability enables CNMF to be optimally applied to time-series problems, such as audio source separation [21] and neural sequence identification [22].

While the non-negativity constraint of CNMF inherently induces some level of sparsity, this is often insufficient for practical needs. Sparse CNMF (SCNMF) emerged as a refinement that integrates sparsity constraints into the CNMF framework which addresses this limitation. SCNMF introduces regularization terms (typically ℓ_1-norm as a convex approximation of the non-convex ℓ_0-norm) to enforce sparsity in the coefficient matrix A which promotes more interpretable, discrete activations of temporal-spectral bases [23, 24]. This sparsity constraint aligns with the physiological properties of human auditory processing and the statistical characteristics of natural signals, where meaningful structures tend to manifest as sparse activations of elementary components. In speech enhancement, for instance, SCNMF outperforms traditional methods like multi-band spectral subtraction and CNMF under non-stationary noise and low SNR conditions by leveraging sparse temporal-spectral patterns to better distinguish speech from noise [25, 26]. Recent studies highlight the value of advanced computational methods for addressing data sparsity and high-dimensionality, which is key to extending the proposed SCNMF framework. For instance, nonlinear modeling and wavelet transform integration can enhance performance on complex real-world data [27, 28], and feature extraction/representation strategies excel at handling high-dimensional information [29, 30]. Specifically, in biomedical and neuroimaging contexts, deep learning- and tensor factorization-based methods have achieved remarkable success in tasks like disease diagnosis [31] which underscores the potential of extending SCNMF to nonlinear settings or integrating it with complementary techniques for broader utility.

Over the past several decades, numerous algorithms for addressing NMF-related problems have been proposed and extensively applied across diverse fields, including machine learning, speech processing, and image analysis [32–35]. For instance, Lee and Seung [36] proposed a multiplicative update (MU) approach and applied it to image processing, while Lin [37] later analyzed the convergence of this method. This method exhibits two major limitations: first, it suffers from slow convergence and a pronounced tendency to converge to local minima; second, traditional NMF fails to model the temporal dependencies among matrix columns, thereby limiting its applicability to time-series data such as speech processing. The other popular algorithm for NMF is the alternating nonnegative least squares (ANLS) method [38]. Its core idea lies in iteratively solving non-negativity constrained least squares subproblems for each factor matrix while fixing the other factors. The standard ANLS solvers often suffer from slow convergence when the data dimension is high and less straightforward to extend to non-quadratic loss functions. Zhang [39] proposed another widely used alternating direction method for NMF and conducted a series of well-designed numerical experiments to compare its performance against the MU and ANLS algorithms. Their numerical results demonstrate that this alternating direction algorithm tends to yield higher-quality solutions while achieving faster computation speeds. In fact, this alternating direction method is more formally known as the alternating direction method of multipliers (ADMM) (see, e.g., [40, 41]). A core characteristic of ADMM is its ability to decompose constrained optimization problems into simpler subproblems, which it accomplishes by introducing auxiliary variables and leveraging dual Lagrange multipliers. This enables seamless incorporation of regularization terms (e.g., ℓ_1 sparsity, group sparsity, or spatial smoothness) without drastically increasing computational complexity. In addition, ADMM offers greater adaptability to NMF variants, such as robust NMF and CNMF [23, 24].

In summary, most existing algorithms for SCNMF frequently adopt a framework based on the multiplicative update approach, which employs a component-wise update strategy. This is attributed to the fact that the convolutive basis and sparsity constraints imposed by the model complicate the problem from a computational perspective. Furthermore, prior work on ADMM has only been explored for CNMF, sparse NMF, etc. In this paper, we focus primarily on addressing a type of SCNMF problem incorporating the ℓ_1-norm with respect to the Euclidean distance. We extend the ADMM framework to tackle this SCNMF problem by introducing a novel variable separation approach and efficient techniques for handling sparsity constraints, and subsequently analyze its convergence properties. The structure of this paper is as follows: Section 2 introduces related work on CNMF and ADMM. In Section 3, we propose a new algorithm for SCNMF with ℓ_1-norm regularization by extending the classic ADMM with acceleration techniques, and provide a simple convergence result. Section 4 presents several sets of computational results to show the performance of the proposed algorithm. Finally, Section 5 offers some concluding remarks.

2. Related works

In this section, we first outline the core preliminary models foundational to SCNMF, including NMF and CNMF. Next, we review the classic ADMM algorithm and then extend it to NMF and CNMF by appropriately introducing separable variables.

2.1. The standard NMF and CNMF

NMF decomposes a high-dimensional non-negative matrix Z into two low-dimensional non-negative matrices Y and A, where Y denotes the basis matrix and A denotes the coefficient matrix. Due to the non-uniqueness of NMF decomposition and the non-negative constraints imposed on matrices Y and A, this decomposition is inherently approximate, meaning that matrix Z can only be decomposed into Y and A in an approximate manner. To quantify the error between Z and the reconstructed matrix YA, a cost function rooted in the Euclidean distance is introduced. The general form of standard NMF [6] can be expressed as:

$$\min \frac{1}{2}\|Z - YA\|_F^2 \quad \text{s.t. } Y \in R_{\geq 0}^{F\times K}, A \in R_{\geq 0}^{K\times N}, \tag{2.1}$$

where $\|\cdot\|_F^2$ is Frobenius norm of the matrix.

Convolutive non-negative matrix factorization extends traditional NMF to two-dimensional domains, where each basis vector acquires an additional dimensionality and the basis matrix is generalized into a tensor. Let T denote the number of time slices per basis. The reconstruction error of CNMF is measured using the Euclidean distance, and the general formulation of standard CNMF [34] is,

$$\min \frac{1}{2}\left\|Z - \sum_{t=0}^{T-1} Y(t) \overset{t\rightarrow}{A}\right\|_F^2 \quad \text{s.t. } Y(t) \in R_{\geq 0}^{F\times K}, \quad \forall t, \overset{t\rightarrow}{A} \in R_{\geq 0}^{K\times N}, \tag{2.2}$$

where $Y(t)$ and A represent the basis matrix and coefficient matrix. The operator $\overset{t\rightarrow}{(\cdot)}$ denotes a column-wise right shift, where each column is shifted right by t positions with zeros padded on the left.

2.2. The classic ADMM

The ADMM method [42, 43] is a distributed convex optimization approach designed to tackle complex problems. By decomposing the coordination process, ADMM breaks down large-scale global challenges into smaller, more tractable local sub-problems, enabling efficient computation and convergence to the optimal solution of the original problem. The classical ADMM is designed to address the following convex optimization problem involving decoupled variables x and y,

$$\min f(x) + g(y) \quad \text{s.t. } Bx + Cy = z. \tag{2.3}$$

Here, B, C, and z denote matrices and a vector of appropriate dimensions, while f and g are convex functions. The augmented Lagrangian function for problem (2.3) can be expressed as,

$$L(x, y, \mu) = f(x) + g(y) + \mu^T(Bx + Cy - z) + \frac{\rho}{2}\|Bx + Cy - z\|_2^2, \tag{2.4}$$

where ρ is a penalty parameter and μ is a Lagrangian multipliers vector. While the framework of the ADMM algorithm is analogous to that of the augmented Lagrangian method (ALM), ADMM updates x and y alternately at each iteration, whereas ALM optimizes over x and y jointly. The core idea of ADMM lies in variable splitting, which decomposes a single optimization variable into multiple auxiliary variables through the introduction of equality constraints. This technique enables the transformation of a complex, coupled optimization problem into simpler, decoupled sub-problems that can be solved iteratively.

2.3. ADMM for NMF and CNMF

To align with the ADMM framework for solving the NMF problem defined in (2.1), auxiliary variables W and H are introduced. Consequently, the original NMF problem can be reformulated as the following equivalent problem:

$$\min \frac{1}{2}\|Z - YA\|_F^2 \quad \text{s.t. } Y = W, A = H, \quad W \in R_{\geq 0}^{F\times K}, H \in R_{\geq 0}^{K\times N}. \tag{2.5}$$

The corresponding augmented Lagrange function of (2.5) is

$$\begin{aligned} L(Y, A, W, H, \Lambda, \Pi) = &\frac{1}{2}\|Z - YA\|_F^2 + \langle \Lambda, Y - W\rangle \\ &+ \frac{\alpha}{2}\|Y - W\|_F^2 + \langle \Pi, A - H\rangle + \frac{\beta}{2}\|A - H\|_F^2, \end{aligned} \tag{2.6}$$

where Λ and Π are the Lagrange Multipliers and α,β are penalty parameters. Although the objective function is non-convex and non-separable with respect to variables Y and A, ADMM can be extended to solve the NMF problem directly. Indeed, ADMM has demonstrated success in addressing numerous non-convex problems, making it a versatile approach for such challenges. For instance, ADMM-based NMF has demonstrated superior performance compared to multiplicative update methods, as documented in [40].

To solve the CNMF problem via ADMM, we introduce variables W and H, which aligns with the problem simplification strategy presented in [44]. Specifically, since matrix Y is generalized as a tensor with T time slices, we define a tensor $U \in R^{F\times K\times T}$ to decompose Z into T component tensors

$$Z = \sum_{t=0}^{T-1} U(t). \tag{2.7}$$

In each iteration, $U(t) \in R^{F\times K}$ is redefined to partition Z proportionally, with the constraint that the sum of all time-slices in U exactly equals Z. Consequently, standard CNMF can be reformulated as a composite optimization problem over a set of slices,

$$\begin{aligned} \min \quad & \frac{1}{2}\sum_{t=0}^{T-1} \|U(t) - Y(t) \overset{t\rightarrow}{A} \|_F^2 \\ \text{s.t.} \quad & Y(t) = W(t), A = H \\ & W(t) \in R_{\geq 0}^{F\times K}, H \in R_{\geq 0}^{K\times N}. \end{aligned} \tag{2.8}$$

With the introduction of the variable tensor U, the augmented Lagrangian function for problem (2.8) can be rewritten as,

$$\begin{aligned} & L(Y, A, W, H, \Lambda, \Pi) \\ = & \frac{1}{2}\sum_{t=0}^{T-1} \|U(t) - Y(t) \overset{t\rightarrow}{A} \|_F^2 + \sum_{t=0}^{T-1} \langle \Lambda(t), Y(t) - W(t)\rangle \\ + & \sum_{t=0}^{T-1} \frac{\alpha}{2}\|Y(t) - W(t)\|_F^2 + \langle \Pi, A - H\rangle + \frac{\beta}{2}\|A - H\|_F^2, \end{aligned} \tag{2.9}$$

where Λ and Π denote Lagrange multipliers, while α and β prepresent penalty parameters. Then, the ADMM for CNMF (2.8) is derived by successively minimizing the augmented Lagrangian function (2.9) with respect to Y, A and (W,H), one at a time while fixing others at their most recent values, and then updating the multipliers (Λ, Π) after each sweep of such alternating minimization.

3. An alternating minimization algorithm for SCNMF

This section presents the main work of this paper. We first introduce the SCNMF model with ℓ_1-norm regularization in Section 3.1. In Section 3.2, we propose an alternating minimization algorithm for the SCNMF problem based on the ADMM framework. Finally, in Section 3.3, we provide a convergence result under certain assumptions.

3.1. SCNMF with ℓ_1-norm regularization

Sparse optimization has emerged as a popular research field in the past few decades, driven by its remarkable success in applications such as compressive sensing, feature extraction, and image processing. The sparsity constraints imposed on the basis and coefficient tensors encourage a more interpretable representation by limiting the number of active elements, effectively extracting only the most relevant features. Meanwhile, the convolutive aspect allows for the modeling of local temporal or spatial dependencies, where basis components are convolved with the coefficient tensors instead of being multiplied element-wise. This combination enables SCNMF to capture both the sparse nature of the underlying data patterns and the sequential or structural relationships between data elements, making it a powerful tool for tasks like source separation, feature extraction, and data compression in complex, high-dimensional datasets. For sparse NMF, many studies focus on inducing sparsity by incorporating ℓ_0, ℓ_1, and other regularization terms into the NMF model. However, because CNMF involves far more complex matrix convolution operations, which makes the design and optimization of regularization terms more challenging, most studies on SCNMF have turned to ℓ_1-norm regularization as a practical solution to enforce sparsity (see [23–26] for example).

In this paper, we propose enforcing sparsity on the coefficient matrix by incorporating ℓ_1-norm regularization into the original objective function of CNMF, which aims to obtain a sparser representation. The problem formulated in (3.1) corresponds to sparse convolutive non-negative matrix factorization (SCNMF),

$$\begin{aligned} \min \quad & \frac{1}{2}\sum_{t=0}^{T-1} \|U(t) - Y(t) \overset{t\rightarrow}{A}\|_F^2 + \lambda\|A\|_1 \\ \text{s.t.} \quad & Y(t) = W(t), A = H \\ & W(t) \in R_{\geq 0}^{F\times K}, H \in R_{\geq 0}^{K\times N}, \end{aligned} \tag{3.1}$$

which constitutes the primary focus of this section.

3.2. An alternating minimization framework for SCNMF

Similar to (2.9), we can easily write the augmented Lagrangian of the SCNMF problem (3.1),

$$\begin{aligned} & L(Y, A, W, H, \Lambda, \Pi) \\ = & \frac{1}{2}\sum_{t=0}^{T-1} \|U(t) - Y(t) \overset{t\rightarrow}{A}\|_F^2 + \lambda\|A\|_1 + \sum_{t=0}^{T-1}\langle \Lambda(t), Y(t) - W(t)\rangle \\ + & \sum_{t=0}^{T-1} \frac{\alpha}{2}\|Y(t) - W(t)\|_F^2 + \langle \Pi, A - H\rangle + \frac{\beta}{2}\|A - H\|_F^2. \end{aligned} \tag{3.2}$$

Consequently, the SCNMF problem (3.1) is transformed into a straightforward nonnegative constrained optimization problem (3.3),

$$\begin{aligned} \min \quad & L(Y, A, W, H, \Lambda, \Pi) \\ \text{s.t.} \quad & W(t) \in R_{\geq 0}^{F\times K}, H \in R_{\geq 0}^{K\times N}. \end{aligned} \tag{3.3}$$

In accordance with the ADMM approach, one sequentially solves (3.3) with respect to each original variable ($Y, A, W,$ and H), followed by an update of the Lagrange multipliers (Λ, Π). The details of this process are elaborated as follows.

3.2.1. Update Y

Upon each update of $Y(t)$, it suffices to optimize the Lagrange formula L in (13) with respect to each $Y(t)$, $\forall t \in [0, T-1]$. Accordingly, the remaining $Y(\tau)$ terms and all other variables A, W, and

H are treated as constants. Consequently, the subproblem of updating $Y(t)$ simplifies to the following unconstrained convex optimization problem,

$$\min_{Y(t)} \frac{1}{2}\|U(t) - Y(t) \overset{t\rightarrow}{A}\|_F^2 + \langle \Lambda(t), Y(t) - W(t)\rangle + \frac{\alpha}{2}\|Y(t) - W(t)\|_F^2. \tag{3.4}$$

Then, a closed-form solution of the $Y(t)$ subproblem can be derived as

$$Y(t) = P_{Y(t)} \left(Q_{Y(t)}\right)^{-1}, \tag{3.5}$$

with $P_{Y(t)}$ and $Q_{Y(t)}$ defined as follows,

$$P_{Y(t)} = U(t)(\overset{t\rightarrow}{A})^T + \alpha W(t) - \Lambda(t), \tag{3.6}$$

$$Q_{Y(t)} = \overset{t\rightarrow}{A} (\overset{t\rightarrow}{A})^T + \alpha I, \tag{3.7}$$

where I denotes a $K \times K$-dimensional identity matrix.

3.2.2. Update A

While keeping the other variables fixed at their most recent values, the minimization of (3.2) with respect to A can be reformulated as,

$$\min_{A} \frac{1}{2}\sum_{t=0}^{T-1}\|U(t) - Y(t) \overset{t\rightarrow}{A}\|_F^2 + \lambda\|A\|_1 + \langle \Pi, A - H\rangle + \frac{\beta}{2}\|A - H\|_F^2. \tag{3.8}$$

Given that the ℓ_1-norm does not admit a direct gradient, the unconstrained convex optimization problem for A is solved via the accelerated proximal gradient method [45]. Now, we denote the objective of (3.8) as $F(A) = f(A) + g(A)$, where

$$f(A) = \frac{1}{2}\sum_{t=0}^{T-1}\|U(t) - Y(t) \overset{t\rightarrow}{A}\|_F^2, \tag{3.9}$$

and

$$g(A) = \lambda\|A\|_1 + \langle \Pi, A - H\rangle + \frac{\beta}{2}\|A - H\|_F^2. \tag{3.10}$$

It is evident that g is a continuous, nonsmooth convex function, while f is a continuous, smooth convex function. In fact, the Lipschitz continuity of $f(A)$ can also be demonstrated through the following derivation,

$$\begin{aligned}
&\|\nabla f(A_1) - \nabla f(A_2)\|_F \\
=& \left\|\sum_{t=0}^{T-1}(-Y(t))^T\left(\overset{t\leftarrow}{U}(t) - Y(t)A_1\right) - \sum_{t=0}^{T-1}(-Y(t))^T\left(\overset{t\leftarrow}{U}(t) - Y(t)A_2\right)\right\|_F \\
=& \left\|\sum_{t=0}^{T-1}(Y(t))^T Y(t)A_1 - \sum_{t=0}^{T-1}(Y(t))^T Y(t)A_2\right\|_F \\
\leq& \lambda_{\max}\left(\sum_{t=0}^{T-1}(Y(t))^T Y(t)\right)\|A_1 - A_2\|_F,
\end{aligned} \tag{3.11}$$

where $\lambda_{\max}\left(\sum_{t=0}^{T-1}(Y(t))^T Y(t)\right)$ denotes the maximum eigenvalue of the matrix $\sum_{t=0}^{T-1}(Y(t))^T Y(t)$. Denote

$$Q_\eta(A,X) = f(X) + \langle A-X, \nabla f(X)\rangle + \frac{\eta}{2}\|A-X\|_F^2 + g(A), \tag{3.12}$$

where $\eta \geq \lambda_{\max}\left(\sum_{t=0}^{T-1}(Y(t))^T Y(t)\right)$. Therefore, the objective $F(A)$ can be approximated by $Q_\eta(A,X)$, which is derived from a second-order approximation of $f(A)$ at X. Furthermore, we can establish the following lemma, which describes the relationship between $F(A)$ and $Q_\eta(A,X)$.

Lemma 1. *For any $\eta \geq \lambda_{\max}\left(\sum_{t=0}^{T-1}(Y(t))^T Y(t)\right)$, and A, X, we have $F(A) \leq Q_\eta(A,X)$.*

Proof.

$$\begin{aligned}
&Q_\eta(A,X) - F(A)\\
&= f(X) - f(A) + \langle A-X, \nabla f(X)\rangle + \frac{\eta}{2}\|A-X\|_F^2\\
&\geq \frac{\eta}{2}\|A-X\|_F^2 - \frac{\lambda_{\max}\left(\sum_{t=0}^{T-1}\left((Y(t))^T Y(t)\right)\right)}{2}\|A-X\|_F^2\\
&\geq 0.
\end{aligned} \tag{3.13}$$

Regarding the optimization of matrix A, minimizing the function $Q_\eta(A,X)$ appears to be more computationally feasible than minimizing $F(A)$. This problem can be effectively addressed using the iterative shrinkage-thresholding algorithm (ISTA) [45], which leverages the proximal operator to handle ℓ_1-norm regularization. Specifically, the A-subproblem (3.8) can be approximated by minimizing $Q_\eta(A,X)$ through the following steps. First, we reformulate the objective function into the standard form of a proximal mapping problem for the ℓ_1-norm, enabling efficient iterative solution via soft-thresholding operations.

$$\begin{aligned}
P_\eta(X) &= \arg\min_A\ Q_\eta(A,X)\\
&= \arg\min_A\ \left\langle A-X, \sum_{t=0}^{T-1}\left[-(Y(t))^T(\overset{\leftarrow t}{U}(t) - Y(t)X)\right]\right\rangle\\
&\quad + \frac{\eta}{2}\|A-X\|_F^2 + \langle \Pi, A-H\rangle + \frac{\beta}{2}\|A-H\|_F^2 + \lambda\|A\|_1\\
&= \arg\min_A\ \frac{\eta}{2}\left\|A - X - \frac{1}{\eta}\sum_{t=0}^{T-1}(Y(t))^T(\overset{\leftarrow t}{U}(t) - Y(t)X)\right\|_F^2\\
&\quad + \frac{\beta}{2}\left\|A - H + \frac{1}{\beta}\Pi\right\|_F^2 + \lambda\|A\|_1\\
&= \arg\min_A\ \frac{\beta+\eta}{2}\|A - h(X)\|_F^2 + \lambda\|A\|_1,
\end{aligned} \tag{3.14}$$

where $h(X) = \frac{1}{\beta+\eta}\left[\eta X + \sum_{t=0}^{T-1}(Y(t))^T \overleftarrow{U}(t) - \sum_{t=0}^{T-1}(Y(t))^T Y(t)X + \beta H - \Pi\right]$. Second, we employ the fast iterative shrinkage-thresholding algorithm (FISTA) to tackle the subsequent problem (3.14):

$$\begin{aligned} \bar{A}_k &= S_{\lambda(\beta+\eta)^{-1}}\left(h\left(A_k\right)\right), \\ t_{k+1} &= \frac{1+\sqrt{1+4t_k^2}}{2}, \\ A_{k+1} &= \bar{A}_k + \left(\frac{t_k-1}{t_{k+1}}\right)\left(\bar{A}_k - \bar{A}_{k-1}\right), \end{aligned} \tag{3.15}$$

where $S_\alpha(\cdot) : R^{m\times n} \to R^{m\times n}$ denotes the contraction operator acting element-wise on a matrix, which is explicitly defined as:

$$S_\alpha(X) = \text{sign}(X) \otimes (|X| - \alpha)_+, \tag{3.16}$$

where the symbol $\otimes$ denotes the Hadamard product (also commonly referred to as the element-wise product) of two matrices. This accelerated optimization technique enhances the convergence rate by incorporating a momentum term, making it particularly suitable for efficiently solving large-scale sparse recovery problems with ℓ_1-norm regularization. By leveraging FISTA, we can expedite the minimization process while maintaining the theoretical guarantees of the iterative shrinkage-thresholding framework.

3.2.3. Update $W(t)$ and H

Since the Lagrangian function (3.2) is separable with respect to variables W and H, the subproblems of (3.3) concerning W and H can be solved simultaneously as shown in (3.17),

$$\begin{aligned} &\min_{W(t)\geq 0} \quad \langle \Lambda(t), Y(t) - W(t)\rangle + \frac{\alpha}{2}\|Y(t) - W(t)\|_F^2, \\ &\min_{H\geq 0} \quad \langle \Pi, A - H\rangle + \frac{\beta}{2}\|A - H\|_F^2. \end{aligned} \tag{3.17}$$

Actually, both the $W(t)$ subproblem and the H subproblem reduce to simple projections onto nonnegative structures. Their update formulas are listed below,

$$\begin{aligned} W(t) &= \max\left(Y(t) + \frac{1}{\alpha}\Lambda(t), 0\right), \\ H &= \max\left(A + \frac{1}{\beta}\Pi, 0\right). \end{aligned} \tag{3.18}$$

3.2.4. Update Λ and Π

The Lagrange multipliers are updated using the classical form of the augmented Lagrangian method,

$$\begin{aligned} &\Lambda(t) \leftarrow \Lambda(t) + \alpha(Y(t) - W(t)), \\ &\Pi \leftarrow \Pi + \beta(A - H). \end{aligned} \tag{3.19}$$

Next, we outline the alternating minimization algorithmic framework for solving SCNMF in Algorithm 1. It is easy to observe from Algorithm 1 that the main computations involve updating matrices Y and A in each iteration. Since the inverse matrices involved in updating $Y(t)$ are both of size $K \times K$, the corresponding linear systems are relatively computationally inexpensive when K is small (i.e., $K \ll N$). In this scenario, the dominant computational tasks for each T consist of the matrix multiplications required to compute P_Y and $Y(t)$, which together account for approximately $O(FNK + FK^2 + K^3)$ arithmetic operations (i.e., scalar additions and multiplications). When updating matrix A, the dominant computational task in each iteration consists solely of matrix multiplications, accounting for approximately $O(TFNK + NK^2)$ arithmetic operations. Therefore, we conclude that the total computational complexity of the proposed Algorithm 1 is on the order of $O(TFNK + TFK^2 + TK^3 + NK^2)$.

Algorithm 1: An alternating minimization algorithm for SCNMF with ℓ_1-norm

Input: $Z \in R^{F\times N}$, integers $T > 0$, *maxiter* > 0.
Output: $Y \in R^{F\times K\times T}$ and $A \in R^{K\times N}$.

Set $\alpha, \beta > 0$.
Set W, H, Λ, Π to zero matrices of appropriate sizes, and A to a random matrix.
Initialize: $\sum_{t=0}^{T-1} U(t) = Z$.
for $k = 1 : maxiter$ **do**
  for $t = 0 : T-1$ **do**
    $P_Y = U(t)(\overset{t\rightarrow}{A})^T + \alpha W(t) - \Lambda(t)$,
    $Q_Y = \overset{t\rightarrow}{A}(\overset{t\rightarrow}{A})^T + \alpha I$,
    $Y(t) = P_Y (Q_Y)^{-1}$.
  end
  Set $\eta \geq \lambda_{\max}\left(\sum_{t=0}^{T-1}(Y(t))^T Y(t)\right)$,
  $hA = \frac{1}{\beta+\eta}\left[\eta A + \sum_{t=0}^{T-1}(Y(t))^T \overset{\leftarrow t}{U}(t) - \sum_{t=0}^{T-1}(Y(t))^T Y(t) A + \beta H - \Pi\right]$,
  $\bar{A}_+ = S_{\lambda(\beta+\eta)^{-1}}(hA)$,
  $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$,
  $A \leftarrow \bar{A}_+ + \left(\frac{t_k - 1}{t_{k+1}}\right)\left(\bar{A}_+ - \bar{A}\right)$.
  for $t = 0 : T-1$ **do**
    $W(t) = \max\left(Y(t) + \frac{1}{\alpha}\Lambda(t), 0\right)$,
    $\Lambda(t) \leftarrow \Lambda(t) + \alpha(Y(t) - W(t))$.
  end
  $H = \max\left(A + \frac{1}{\beta}\Pi, 0\right)$,
  $\Pi \leftarrow \Pi + \beta(A - H)$.
  if *stopping criterion (3.20) is met* **then**
    output Y, A, and exit.
  end
end

To ensure practical convergence, the following stopping criterion is employed in Algorithm 1: for a predefined tolerance threshold *tol*, the iterative process terminates when

$$\min\left\{\frac{|f_k - f_{k+1}|}{|f_k|}, \max\left(\frac{\|Y_k - Y_{k+1}\|_F}{\|Y_k\|_F}, \frac{\|A_k - A_{k+1}\|_F}{\|A_k\|_F}\right)\right\} \le tol, \tag{3.20}$$

where $f_k = \|Z - \sum_{t=0}^{T-1} Y_k(t) \overset{t\rightarrow}{A_k} \|_F$.

3.3. Convergence analysis

Global convergence is guaranteed when the classical ADMM is applied to two-block convex programs in the form of (2.3). However, to the best of our knowledge, there exists no general global convergence theory for non-convex programming or convex systems involving three or more blocks. It is important to note that problem (3.1) is non-convex, and its updates involve three blocks with respect to Y, A and (W, H). Given these limitations, we derive a convergence analysis for the proposed alternating minimization framework, which holds under specific assumptions.

A tuple (Y, A, W, H) meets the KKT conditions corresponding to problem (3.1) provided that there exist Λ and Π satisfying the following conditions:

$$\begin{aligned}
(U(t) - Y(t) \overset{t\rightarrow}{A})(- \overset{t\rightarrow}{A})^T + \Lambda(t) &= 0, && (3.21a)\\
\sum_{t=0}^{T-1}(-Y(t))^T(\overset{t\leftarrow}{U}(t) - Y(t)A) + \Pi + \lambda v &= 0, && (3.21b)\\
Y(t) - W(t) &= 0, && (3.21c)\\
A - H &= 0, && (3.21d)\\
\Lambda(t) \le 0 \le W(t), \quad \Lambda(t) \odot W(t) &= 0, && (3.21e)\\
\Pi \le 0 \le H, \quad \Pi \odot H &= 0, && (3.21f)
\end{aligned}$$

where $v \in \partial\|A\|_1$, and $\odot$ denotes element multiplication. To simplify the presentation, all variables in problem (3.1) are grouped as a single entity, denoted by:

$$V := (Y, A, W, H). \tag{3.22}$$

Theorem 1. *Let* $\{(V_k, \Lambda_k, \Pi_k)\}$ *denote the sequence generated by the alternating minimization Algorithm 1. If* $\{(\Lambda_k, \Pi_k)\}$ *is a bounded sequence of Lagrange multipliers [46] satisfying*

$$\sum_{k=0}^{\infty}\left(\|\Lambda_{k+1} - \Lambda_k\|_F^2 + \|\Pi_{k+1} - \Pi_k\|_F^2\right) < \infty, \tag{3.23}$$

then every accumulation point of $\{V_k\}$ *satisfies the* KKT *conditions for problem (3.1).*

Proof. First and foremost, we establish that the sequences satisfy the following convergence properties:

$$\|V_{k+1} - V_k\| \to 0, \|(\Lambda_{k+1}, \Pi_{k+1}) - (\Lambda_k, \Pi_k)\| \to 0. \tag{3.24}$$

To substantiate this claim, we first note that the observation $L(V, \Lambda, \Pi)$ is bounded from below. This can be inferred from:

$$\begin{aligned} L(V, \Lambda, \Pi) = & \frac{1}{2} \sum_{t=0}^{T-1} \| U(t) - Y(t) \overset{t\rightarrow}{A} \|_F^2 + \lambda \|A\|_1 \\ & + \sum_{i=0}^{T-1} \frac{\alpha}{2} \left\| Y(t) - W(t) + \frac{\Lambda(t)}{\alpha} \right\|_F^2 - \frac{1}{2\alpha} \sum_{i=0}^{T-1} \|\Lambda(t)\|_F^2 \\ & + \frac{\beta}{2} \left\| A - H + \frac{\Pi}{\beta} \right\|_F^2 - \frac{1}{2\beta} \|\Pi\|_F^2, \end{aligned} \tag{3.25}$$

and $\{(\Lambda, \Pi)\}$ is bounded. Furthermore, the augmented Lagrangian L is strongly convex with respect to each variable $Y,\ A,\ W,\ H$. Specifically, for any $Y(t)$ and $\Delta Y(t)$, the following inequality holds

$$L(Y(t) + \Delta Y(t)) - L(Y(t)) \geq \langle \partial_{Y(t)} L(Y(t)), \Delta Y(t) \rangle + \alpha \|\Delta Y(t)\|_F^2. \tag{3.26}$$

Additionally, since $Y^*(t)$ minimizes L with respect to Y, the following inequality holds for any feasible $\Delta Y^*(t)$,

$$\langle \partial_{Y(t)} L\left(Y^*(t)\right), \Delta Y^*(t) \rangle \geq 0. \tag{3.27}$$

Accordingly, we have the following inequality:

$$L\left(Y_k(t)\right) - L\left(Y_{k+1}(t)\right) \geq \alpha \left\|Y_k(t) - Y_{k+1}(t)\right\|_F^2 . \tag{3.28}$$

By the same reasoning, analogous results holds for A, W and H,

$$\begin{aligned} L\left(A_k\right) - L\left(A_{k+1}\right) &\geq \beta \left\|A_k - A_{k+1}\right\|_F^2 , \\ L\left(W_k(t)\right) - L\left(W_{k+1}(t)\right) &\geq \alpha \left\|X_k(t) - X_{k+1}(t)\right\|_F^2 , \\ L\left(H_k\right) - L\left(H_{k+1}\right) &\geq \beta \left\|H_k - H_{k+1}\right\|_F^2 . \end{aligned} \tag{3.29}$$

Let $c := \min\{\alpha, \beta\}$, the difference in the augmented Lagrangian values between two consecutive iterations can be readily bounded,

$$\begin{aligned} & L\left(V_k, \Lambda_k, \Pi_k\right) - L\left(V_{k+1}, \Lambda_{k+1}, \Pi_{k+1}\right) \\ & = L\left(V_k, \Lambda_k, \Pi_k\right) - L\left(V_{k+1}, \Lambda_k, \Pi_k\right) \\ & + L\left(V_{k+1}, \Lambda_k, \Pi_k\right) - L\left(V_{k+1}, \Lambda_{k+1}, \Pi_{k+1}\right) \\ & \geq c \left\|V_k - V_{k+1}\right\|_F^2 - \frac{1}{c} \left[\left\|\Lambda_k - \Lambda_{k+1}\right\|_F^2 + \left\|\Pi_k - \Pi_{k+1}\right\|_F^2 \right]. \end{aligned} \tag{3.30}$$

As a result of the boundedness of both $\{(\Lambda, \Pi)\}$ and $L(V, \Lambda, \Pi)$, it follows that

$$\sum_{t=0}^{T-1} c \left\|V_k - V_{k+1}\right\|_F^2 < \infty. \tag{3.31}$$

According to the update formulas (3.18) and (3.19), we can derive

$$Y_k(t) + \frac{\Lambda_k(t)}{\alpha} = \max\left(0, Y_k(t) + \frac{\Lambda_k(t)}{\alpha}\right) + \min\left(0, Y_k(t) + \frac{\Lambda_k(t)}{\alpha}\right) = W_k(t) + \frac{\Lambda_{k+1}(t)}{\alpha}. \tag{3.32}$$

Given the convergence of the Lagrange multipliers Λ and Π, we directly obtain

$$Y_k(t) - W_k(t) \to 0, \tag{3.33}$$

and in the same way, it follows that

$$A_k - H_k \to 0. \tag{3.34}$$

At this point, the equality constraints within the KKT systems (3.21c) and (3.21d) are derived.

Since $Y_{k+1}(t)$ is the minimizer of the $Y(t)$-subproblem (3.4), the first-order optimality condition given below is satisfied,

$$Y_{k+1}(t)\left[\overset{t\rightarrow}{A_k}\,(\overset{t\rightarrow}{A_k})^T + \alpha I\right] - U(t)(\overset{t\rightarrow}{A_k})^T - \alpha W_k(t) + \Lambda_k(t) = 0. \tag{3.35}$$

By adding and subtracting the term $Y_k(t)[\overset{t\rightarrow}{A_k}\,(\overset{t\rightarrow}{A_k})^T + \alpha I]$ and rearranging the resulting expression, we can obtain

$$\begin{aligned}
&(Y_{k+1}(t) - Y_k(t))\left[\overset{t\rightarrow}{A_k}\,(\overset{t\rightarrow}{A_k})^T + \alpha I\right] \\
&= (U(t) - Y_k(t)\,\overset{t\rightarrow}{A_k})(\overset{t\rightarrow}{A_k})^T + \alpha(W_k(t) - Y_k(t)) - \Lambda_k(t) \\
&= 0.
\end{aligned} \tag{3.36}$$

According to (3.33) and the convergence property of Y_k, the following holds:

$$(U(t) - Y_k(t)\,\overset{t\rightarrow}{A_k})(\overset{t\rightarrow}{A_k})^T - \Lambda_k(t) \to 0. \tag{3.37}$$

Evidently, the first equation (3.21a) in the KKT condition is satisfied at any accumulation point.

Next, we turn our attention to the update rule for A in Algorithm 1 and the reformulated form (3.14). For simplicity, we denote

$$A_{k+1} = P_\eta(A) = \arg\min_A Q_\eta(A, A_k). \tag{3.38}$$

Thus, by invoking the first-order necessary condition, we obtain

$$\sum_{t=0}^{T-1}(-Y_{k+1}(t))^T(\overset{\leftarrow t}{U}(t) - Y_{k+1}(t)A_{k+1}) + \Pi_k + \eta(A_{k+1} - A_k) + \beta(A_{k+1} - H_k) + \lambda v_{k+1} = 0, v_{k+1} \in \partial\|A_{k+1}\|_1. \tag{3.39}$$

Given the convergence properties of A_k, Π_k, together with Eq (3.34), we derive

$$\sum_{t=0}^{T-1}(-Y_{k+1}(t))^T(\overset{\leftarrow t}{U}(t) - Y_{k+1}(t)A_{k+1}) + \Pi_{k+1} + \lambda v_{k+1} \to 0. \tag{3.40}$$

Clearly, the KKT conditions (3.21b) holds as well.

The nonnegativity of $W(t)$ and H is guaranteed by the algorithm's construction. Thus, we only need to verify the non-positivity of $\Lambda(t)$ and Π, as well as the complementarity between $W(t)$ and $\Lambda(t)$, and between H and Π. We now examine the update formulas (3.18) of $W(t)$ and H:

$$W(t) = \max\left(Y(t) + \frac{1}{\alpha}\Lambda(t), 0\right), H = \max\left(A + \frac{1}{\beta}\Pi, 0\right).$$

Obviously, if $H_{ij} = A_{ij} = 0$, this implies $\max\left(\frac{1}{\beta}\Pi_{ij}, 0\right) = 0$, which in turn yields $\Pi_{ij} \le 0$. Conversely, if $H_{ij} = A_{ij} > 0$, we can drive $\Pi_{ij} = 0$. This establishes both the non-positivity of Π and the complementarity between H and Π. Due to their identical structure, the same reasoning applies to $W(t)$ and $\Lambda(t)$, thereby confirming the non-positivity of $\Lambda(t)$ and the complementarity between $W(t)$ and $\Lambda(t)$.

This completes the proof.

The following corollary is an immediate consequence.

Corollary 1. *Whenever the sequence $\{(V_k, \Lambda_k, \Pi_k)\}_k^\infty$ converges, its limit is a KKT point.*

While the above simple result is far from satisfactory, it nonetheless provides some assurance regarding the behavior of the ADMM-like algorithm when applied to the non-convex SCNMF problem. Further theoretical investigations in this direction are certainly warranted.

4. Numerical experiments

In this section, we conduct numerical experiments to evaluate the performance of the proposed SCNMF algorithm. All numerical experiments were run under Matlab version R2023b on a PC with an Intel Core i9 processor at 3.3 GHz with 32 GB RAM.

4.1. Synthetic dataset

In this section, we specify the synthetic dataset for each test instance as well as the basic default settings of the algorithm. First, we randomly generate a non-negative matrix with dimensions $F = 200$ and $K = 50$ as the oracle tensor base Y^*, and a non-negative oracle coefficient matrix A^* with dimensions $K = 50$ and $N = 1000$. Following the method for randomly synthesizing datasets described in [44], we select three cases for the value of T ($T = 2, 5$, or 10), and then construct the data matrix Z with dimensions $F = 200$ and $N = 1000$ as

$$Z = \sum_{t=0}^{T-1} Y^*(t) \overset{t\rightarrow}{A^*} . \tag{4.1}$$

Unless otherwise specified, we set the maximum number of iterations as $maxiter = 1000$ and the tolerance as $tol = 1e-6$.

4.2. Validation of the proposed algorithm's convergence

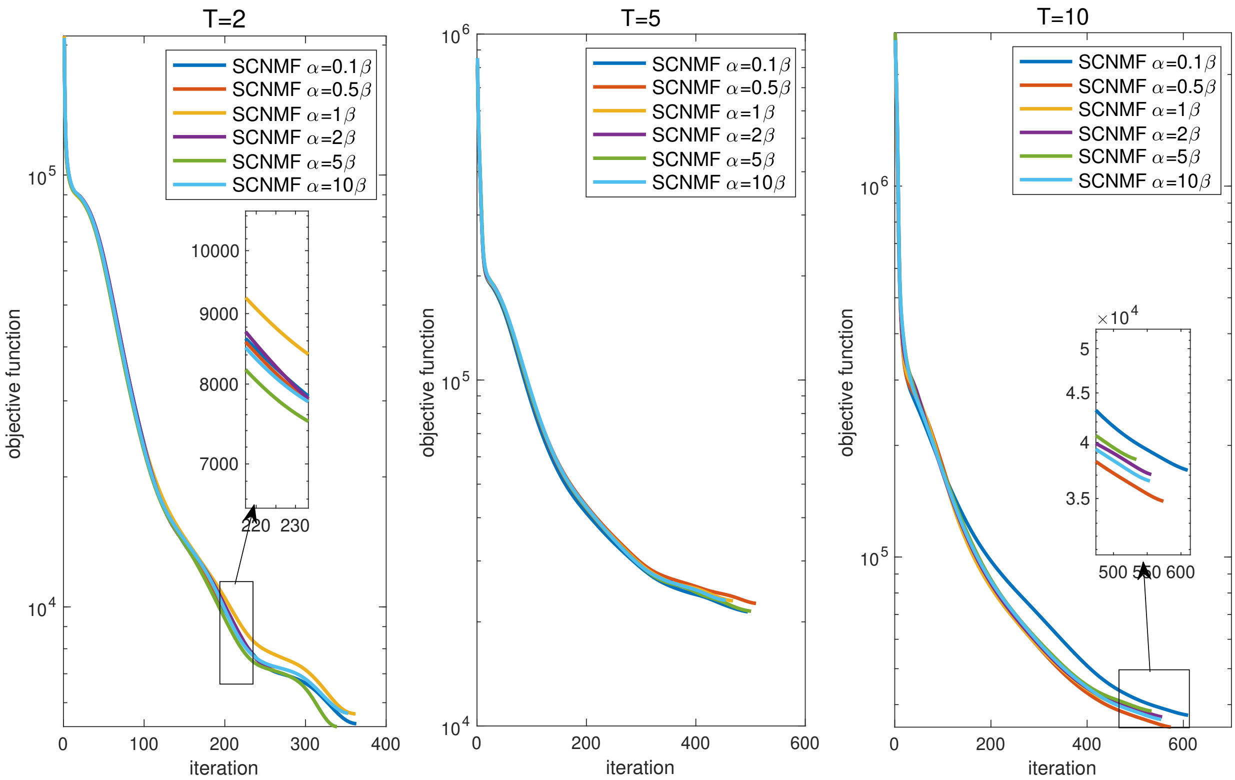
To demonstrate the effectiveness of our Algorithm 1, we conduct a series of tests using synthetic data to examine the algorithm's behavior under different values of penalty parameters (α, β), regularizer λ and different settings of T (where $T = 2, 5, 10$). In this section, we first fix the ℓ_1 regularizer $\lambda = 1$ and discuss the impact of different ratios of α to β as well as individual values of $\alpha(\beta)$. We then explore the performance of the proposed algorithm with varying λ.

4.2.1. Test for different ratios of α to β

We fix $\beta = 1$ and test on 6 pairs of initial penalty parameter values with different ratio k

$$\alpha = k * \beta, \quad k = 0.1, 0.5, 1, 2, 5, 10.$$

The history of objective values of SCNMF problem, considering different lengths T of the convolutive bases, is plotted in Figure 1. It should be evident from Figure 1 that the algorithm 1 converges for all 6 pairs of penalty parameters when $T = 2, 5$ and 10. Moreover, the proposed algorithm is not sensitive to the ratio of these penalty parameters. For instance, the curves of objective values almost coincide before approximately 200 iterations and then begin to diverge slightly. Nevertheless, the iterations terminate with objective values of a similar magnitude. This observation represents a major advantage of robustness to penalty parameters with different ratio scales. In practice, we typically set $\alpha = \beta$ (or $\alpha = 2\beta$), as a tradeoff within an appropriate range.



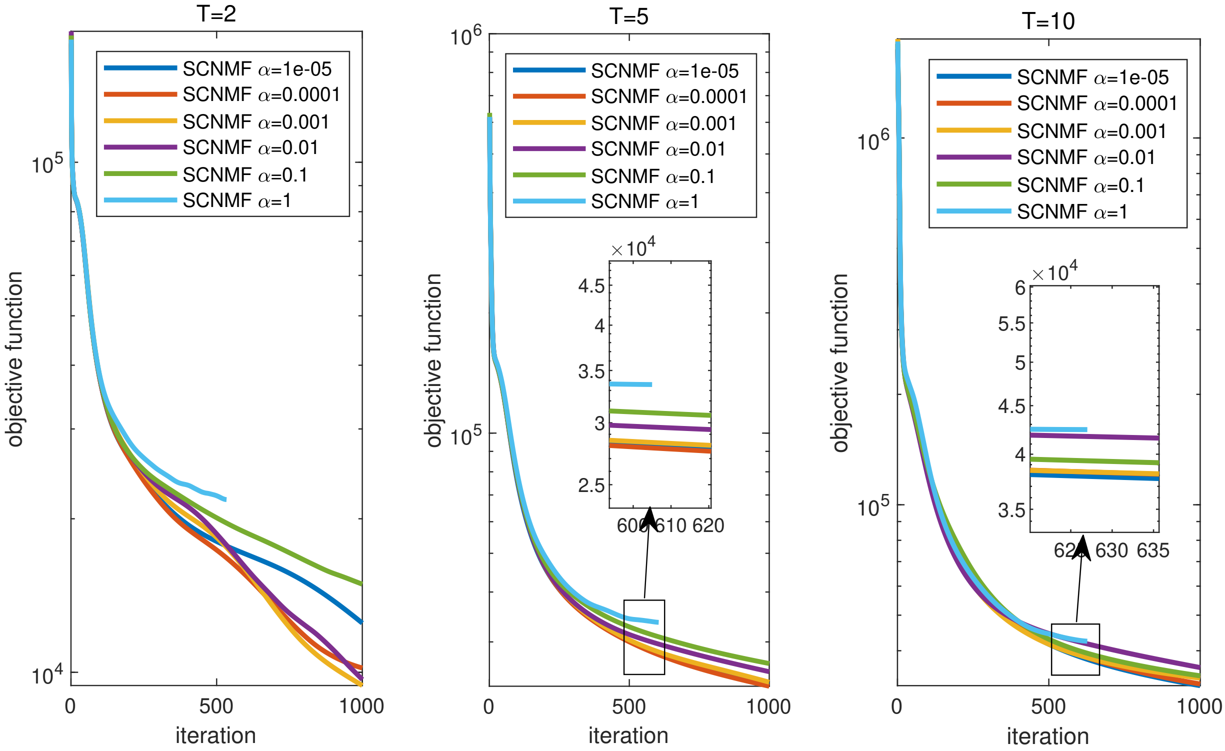
(a) (b) (c)

Figure 1. Convergence history of objective values produced by Algorithm 1 with different ratios of penalty parameters when $T = 2, 5$ and 10.

4.2.2. Test for different initial $\alpha(\beta)$

Here, we initialize $\alpha(= 2\beta)$ to the default value ranging from 1e-5 to 1, respectively, and evaluate the algorithm's performance. Over many randomized runs with different random starting points, we present a set of typical convergence history results in Figure 2. As shown by the convergence history curves in Figure 2, the proposed algorithm can always converge effectively within a certain range of penalty parameters. In addition, for $T = 2, 5$ and 10, the behavior of our algorithm appears consistent with respect to different settings of the parameter α. However, as observed from Figure 2, the curves of objective function values generally appear sequentially from top to bottom as α varies from 1 to

1e-5. Actually, α and β denote the weights of the term $\|Y - W\|_F^2$ and the term $\|A - H\|_F^2$, respectively, in the augmented Lagrangian function (3.2). These weights penalize the equation constraints $Y = W$ and $A = H$, respectively. Specifically, a larger α assigns a larger weight to the penalty term, which in turn results in a slightly larger objective value and causes the algorithm to terminate earlier (e.g., when $\alpha = 1$ in Figure 3(a)). However, for $T = 5$ and $T = 10$, the proposed algorithm does not exhibit significant differences across different magnitude scales of α.



(a) (b) (c)

Figure 2. Convergence history of objective values produced by Algorithm 1 with different penalty parameters when $T = 2, 5$ and 10.

4.2.3. Test for different λ

In this experiment, we utilize the proposed algorithm to SCNMF with varying values of the regularizer λ. Specifically, we fix $\alpha = \beta = 0.001$ and adjust λ within the range of 0.01 to 10. The reconstruction error across iterations is presented in Figure 3. From Figure 3, it can be observed that the output curves exhibit qualitatively similar behavior for $T = 2, 5$ and 10. That is, the algorithm terminates earlier for larger values of λ, which requires fewer iterations to meet the same tolerance. Conversely, it is found that $\lambda < 1$ tends to result in similar reconstruction errors. Specifically, as λ decreases below 1, across numerous randomized runs, the number of instances where the algorithm attains an indistinct objective value increases significantly. Additionally, it should be noted that the reconstruction error tends to decrease as the regularizer λ becomes smaller. Indeed, this observation is theoretically consistent, as a stronger regularization term in the SCNMF model diminishes the weight assigned to the fidelity term.

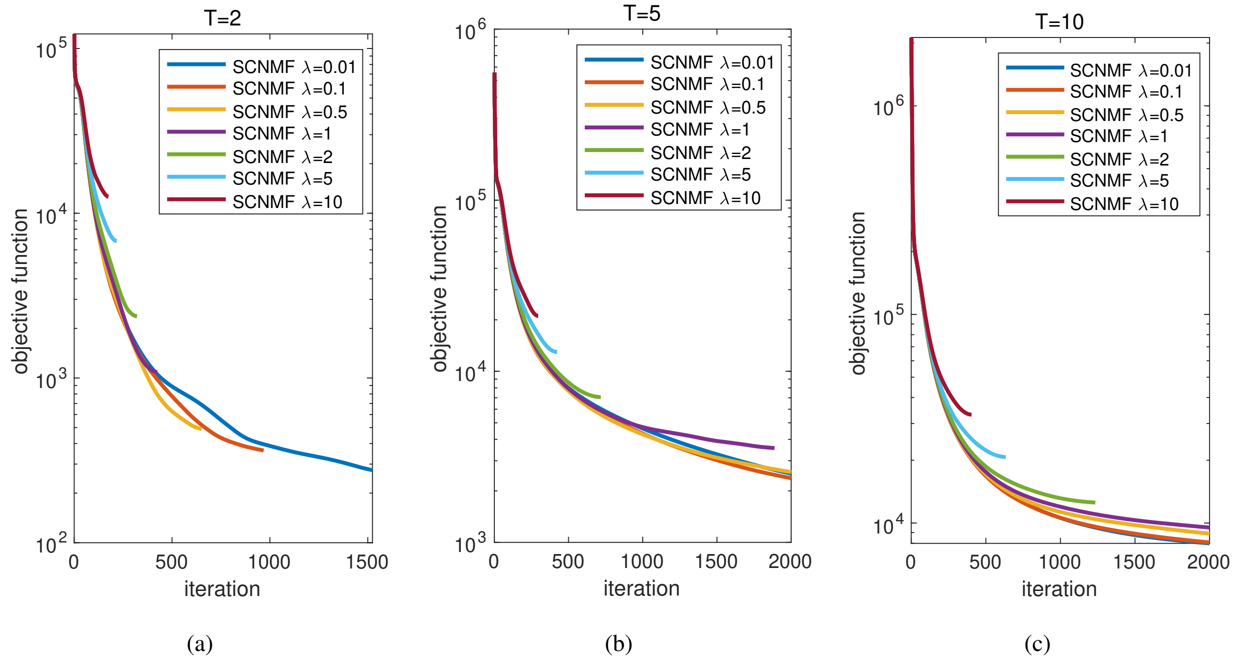


Figure 3. Convergence history of objective values produced by Algorithm 1 with different λ when $T = 2, 5$ and 10.

4.3. Comparison of sparse solutions between SCNMF and CNMF

Early studies have indicated that algorithms based on ADMM tend to generate sparse solutions, a property that is desirable in numerous applications [40, 44]. In this section, we employ the ADMM algorithm to solve CNMF and SCNMF respectively, and compare the sparsity of their solutions under different penalty parameters (α, β), regularizer λ, and temporal slices T. Since the CNMF problem (2.8) lacks ℓ_1 regularization of A and shares the same constraints as the SCNMF problem (3.1), the ADMM algorithm for CNMF adopts the same framework as Algorithm 1, except for the update formula of A. We now focus on the A-subproblem within the ADMM framework for the CNMF problem,

$$\min_{A} \frac{1}{2} \sum_{t=0}^{T-1} \|U(t) - Y(t) \overset{t\to}{A} \|_F^2 + \langle \Pi, A - H \rangle + \frac{\beta}{2} \|A - H\|_F^2. \tag{4.2}$$

More straightforwardly, a closed-form update for A can be derived as

$$A = (Q_A)^{-1} P_A, \tag{4.3}$$

where,

$$P_A = \sum_{t=0}^{T-1} (Y(t))^T \overset{\leftarrow t}{U}(t) + \beta H - \Pi, \tag{4.4}$$

$$Q_A = \sum_{t=0}^{T-1} (Y(t))^T Y(t) + \beta I. \tag{4.5}$$

In this test, we select three different settings of penalty parameters: $\alpha = \beta = 0.001, 0.002$ and 0.005, along with different regularizer values $\lambda = 0.01, 0.1, 0.5, 1, 2, 5$ and 10. Each experiment is repeated ten times, and the average percentage of zero entries in H are calculated. The results are presented in Table 1.

Table 1. The proportion of zero values in matrix H obtained by both CNMF and SCNMF.

T	α	CNMF	SCNMF						
			$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 0.5$	$\lambda = 1$	$\lambda = 2$	$\lambda = 5$	$\lambda = 10$
2	0.001	7.02	7.64	8.88	10.38	14.94	17.07	23.52	40.20
	0.002	7.30	7.69	8.05	10.25	14.59	16.47	22.71	38.90
	0.005	7.38	7.56	8.32	9.83	14.31	15.99	22.22	34.24
5	0.001	7.79	8.29	9.06	12.28	13.71	15.27	21.92	28.98
	0.002	8.10	8.19	9.19	11.51	14.03	15.30	22.00	28.91
	0.005	7.78	8.80	9.72	11.20	13.01	14.83	21.70	28.36
10	0.001	8.45	10.62	12.10	12.69	13.93	14.48	19.16	24.24
	0.002	7.67	9.84	9.93	13.24	14.30	14.58	18.44	22.87
	0.005	6.56	8.27	8.99	12.63	12.98	13.75	16.37	21.18

From Table 1, it can be observed that the percentage of zero values in the results generated by SCNMF is consistently higher than that in CNMF. This finding confirms that our proposed algorithm is indeed capable of achieving a significant sparsity effect. In addition, we find that a larger value of λ leads to increased sparsity in H, especially when $\lambda > 1$. In contrast, the penalty parameters (α, β) do not have a distinct impact on sparsity. For varying convolution slices T, it is apparent that an increase in the number of slices T leads to a slight reduction in sparsity for SCNMF when $\lambda > 1$.

Table 2. The proportion of zero values in tensor W obtained by both CNMF and SCNMF.

T	α	CNMF	SCNMF						
			$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 0.5$	$\lambda = 1$	$\lambda = 2$	$\lambda = 5$	$\lambda = 10$
2	0.001	3.22	3.19	4.71	5.05	5.42	4.10	5.95	4.89
	0.002	3.41	4.72	4.64	4.53	4.55	4.25	6.91	5.37
	0.005	4.35	4.49	4.31	5.34	3.34	4.47	5.59	4.83
5	0.001	11.21	11.92	10.34	10.81	15.99	11.92	9.52	9.03
	0.002	9.35	9.19	10.12	9.55	14.12	9.77	9.83	8.89
	0.005	6.72	6.89	7.47	10.20	13.02	8.59	9.72	8.51
10	0.001	17.19	17.08	17.53	17.15	17.38	17.82	17.43	13.44
	0.002	16.15	16.11	16.82	17.13	18.37	18.08	15.69	14.86
	0.005	14.81	13.78	14.02	15.35	16.01	16.31	12.76	10.03

In fact, similar to all NMF-based models, both CNMF and SCNMF are also capable of inducing sparsity in the basis matrix. We computed the average proportion of zero values in the tensor W output by CNMF and SCNMF, respectively. These results are presented in Table 2.

It should be noted that the sparsity in W does not appear to differ significantly between CNMF and SCNMF, even when varying penalty parameters and regularizers. In construct, it is interesting to observe that the percentage of zeros in W decreases slightly as λ increases (for instance, when $\lambda = 10$).

4.4. Experiments on real-world speech dataset

In this section, we evaluate the proposed algorithm on a real-world speech dataset. Following the experimental setup in [44], 50 sentences are randomly selected from the TIMIT database [47] (sampling rate: 16 kHz) to formulate the training database. A Hamming window of 512 samples (i.e., 32 ms) and a frame shift of 128 points (i.e., 8 ms) are used to compute the short-time Fourier transform (STFT) magnitude. From this process, a non-negative matrix of size $257 \times 22{,}395$ is obtained. Following the settings, 40 convolutive bases of length $T = 10$ are learned from the observation.

The performance of the proposed algorithm is compared against several recently proposed ADMM algorithms from [44] and the majorization-minimization algorithm from [20], including the heuristic approach from [33] and two MM algorithms. Since [20] claims that the sequential MM algorithm (MM2) yields the best overall trade-off between computation time and performance, only MM2 is adopted in this experiment. All algorithms adopt the same initialization for Y and A, thus sharing the same initial objective function value. The results of objective value with respect to time consumption are shown in Figure 4.

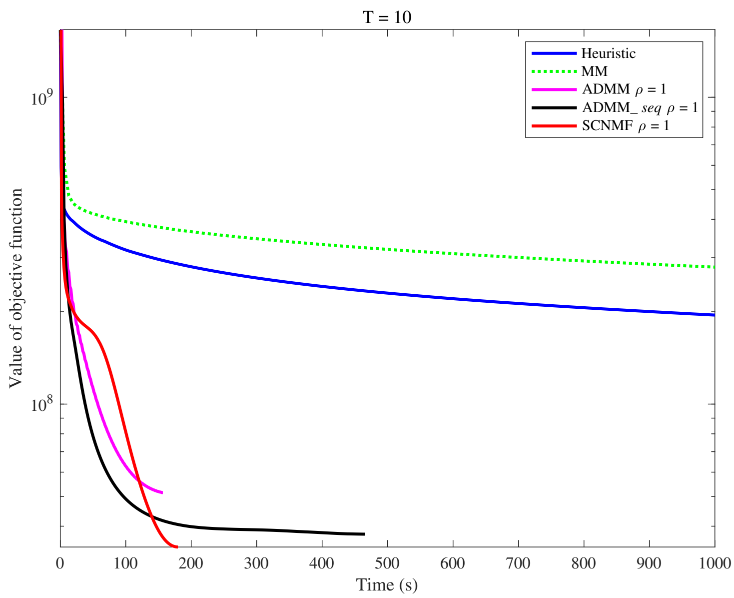


Figure 4. Comparisons of objective values in terms of time consumption: proposed algorithm (SCNMF) vs. baseline methods (Heuristic [33], MM [20], ADMM [44]) for a speech dataset.

As shown in the experimental results, both the proposed SCNMF algorithm and the two ADMM algorithms (from [44]) successfully terminate within the default tolerance for the relative change in the objective value. By contrast, the heuristic method and the MM algorithm exhibit a slower decreasing rate and fail to converge to local optimal points even after 1000 seconds. For comparison, the two ADMM algorithms decrease faster than the proposed algorithm, and ADMM_*seq* achieves the lowest objective values in the early initial stages. Subsequently, the proposed algorithm converges abruptly (after 100 seconds) and ultimately attains the best objective value among all tested algorithms. In terms of running time, the ADMM and SCNMF require much less time than the others; however, the ADMM is not as good as SCNMF in terms of fidelity. This experiment demonstrates that the proposed algorithm outperforms the baseline methods in terms of running time and accuracy.

5. Conclusions

This paper addresses the sparse feature extraction challenge in CNMF tasks by proposing an SCNMF model integrated with ℓ_1-norm regularization. To efficiently solve the model, we design an alternating minimization algorithm based on the ADMM framework and FISTA, leveraging ADMM's strength in handling complex constrained optimization and FISTA's efficiency in solving ℓ_1-regularized subproblems.

Theoretically, under mild technical assumptions, we establish rigorous convergence results for the proposed algorithm, filling the gap in theoretical analysis of ADMM-based methods for non-convex SCNMF problems and providing explicit guarantees for the algorithm's reliability. Computationally, the algorithm decomposes complex optimization problems into tractable subproblems solvable via closed-form or efficient iterative methods. Numerical experiments on synthetic and real-world datasets demonstrate its robust convergence with reasonable penalty parameters (avoiding precise tuning), significantly sparser solutions than standard CNMF, and excellent performance in signal fidelity and computational efficiency for large-scale speech data.

Future work will focus on enhancing the algorithm via dynamic parameter tuning to improve adaptability to diverse datasets and extend its application to complex speech processing tasks (e.g., speech enhancement and separation), aiming to advance the practical application of SCNMF in speech processing and related domains.

Use of AI tools declaration

The author declares they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by the Liaoning Provincial Joint Program of Science and Technology Plan (Grant No. 2025-MSLH-103), the Basic Scientific Research Projects of Colleges and Universities in Liaoning Province (Grant No. LJ212413631010) and Dalian Science and Technology Star Project (Grant No. 2021RQ064).

Conflict of interest

The author states no conflict of interest.

References

1. M. S. Bartlett, J. R. Movellan, T. J. Sejnowski, Face recognition by independent component analysis, *IEEE Trans. Neural Netw.*, **13** (2002), 1450–1464. https://doi.org/10.1109/TNN.2002.804287
2. K. Guo, X. Xu, F. Qiu, J. Chen, A novel incremental weighted PCA algorithm for visual tracking, in *2013 IEEE International Conference on Image Processing*, IEEE, (2013), 3914–3918. https://doi.org/10.1109/ICIP.2013.6738806
3. J. Gan, T. Liu, L. Li, J. Zhang, Nonnegative matrix factorization: A survey, *Comput. J.*, **64** (2021), 1080–1092. https://doi.org/10.1093/comjnl/bxab103
4. F. Dargahi, S. Babaie-Kafaki, Z. Aminifard, M. Hajian-Berenjestanaki, Solving an augmented nonnegative matrix factorization model by modified scaled nonmonotone memoryless BFGS methods devised based on the ellipsoid vector norm, *Math. Methods Appl. Sci.*, **48** (2025), 9088–9097. https://doi.org/10.1002/mma.10781
5. V. Y. F. Tan, C. Fevotte, Automatic relevance determination in non-negative matrix factorization with the beta-divergence, *IEEE Trans. Pattern Anal. Mach. Intell.*, **35** (2012), 1592–1605. https://doi.org/10.1109/TPAMI.2012.240
6. D. Lee, H. Seung, Learning the parts of objects by non-negative matrix factorization, *Nature*, **401** (1999), 788–791. https://doi.org/10.1038/44565
7. E. Gonzalez, Y. Zhang, Accelerating the lee-seung algorithm for nonnegative matrix factorization, *Dept. Comput. Appl. Math., Rice Univ., Houston, TX, Tech. Rep. TR-05-02*, **2005** (2005), 1–13.
8. P. O. Hoyer, Non-negative sparse coding, in *Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing*, IEEE, (2002), 557–565. https://doi.org/10.1109/NNSP.2002.1030067
9. P. O. Hoyer, Non-negative matrix factorization with sparseness constraints, preprint, arXiv:cs/0408058.
10. R. Peharz, F. Pernkopf, Sparse nonnegative matrix factorization with ℓ_0-constraints, *Neurocomputing*, **80** (2012), 38–46. https://doi.org/10.1016/j.neucom.2011.09.024
11. J. Yang, Y. Zhang, Alternating direction algorithms for ℓ_1-problems in compressive sensing, *SIAM J. Sci. Comput.*, **33** (2011), 250–278. https://doi.org/10.1137/090777761
12. M. Gong, X. Jiang, H. Li, K. C. Tan, Multiobjective sparse non-negative matrix factorization, *IEEE Trans. Cybern.*, **49** (2019), 2941–2954. https://doi.org/10.1109/TCYB.2018.2834898
13. Q. M. Pham, D. Lachmund, D. N. Hao, Convergence of proximal algorithms with stepsize controls for non-linear inverse problems and application to sparse non-negative matrix factorization, *Numer. Algor.*, **85** (2020), 1255–1279. https://doi.org/10.1007/s11075-019-00864-x

14. M. S. Islam, Y. Zhu, M. I. Hossain, R. Ullah, Z. Ye, Supervised single channel dual domains speech enhancement using sparse non-negative matrix factorization, *Digital Signal Process.*, **100** (2020), 102697. https://doi.org/10.1016/j.dsp.2020.102697

15. P. Deng, T. Li, H. Wang, D. Wang, S. Horng, R. Liu, Graph regularized sparse non-negative matrix factorization for clustering, *IEEE Trans. Comput. Soc. Syst.*, **10** (2023), 910–921. https://doi.org/10.1109/TCSS.2022.3154030

16. W. Min, T. Xu, X. Wan, T. H. Chang, Structured sparse non-negative matrix factorization with $\ell_{2,0}$–norm, *IEEE Trans. Knowl. Data Eng.*, **35** (2023), 8584–8595. https://doi.org/10.1109/TKDE.2022.3206881

17. W. Wang, J. Meng, H. Li, J. Fan, Non-negative matrix factorization for overlapping community detection in directed weighted networks with sparse constraints, *Chaos*, **33** (2023), 053111. https://doi.org/10.1063/5.0152280

18. A. Degleris, N. Gillis, A provably correct and robust algorithm for convolutive nonnegative matrix factorization, *IEEE Trans. Signal Process.*, **68** (2020), 2499–2512. https://doi.org/10.1109/TSP.2020.2984163

19. Y. Li, X. Zhang, M. Sun, Y. Hu, L. Li, Online convolutive non-negative bases learning for speech enhancement, *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.*, **E99.A** (2016), 1609–1613. https://doi.org/10.1587/transfun.E99.A.1609

20. D. Fagot, H. Wendt, C. Fevotte, P. Smaragdis, Majorization-minimization algorithms for convolutive NMF with the beta-divergence, in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, (2019), 8202–8206. https://doi.org/10.1109/ICASSP.2019.8683837

21. M. Sun, Y. Li, J. F. Gemmeke, X. Zhang, Speech enhancement under low SNR conditions via noise estimation using sparse and low-rank NMF with Kullback-Leibler divergence, *IEEE/ACM Trans. Audio Speech Lang. Process.*, **23** (2015), 1233–1242. https://doi.org/10.1109/TASLP.2015.2427520

22. E. Mackevicius, A. Bahle, A. Williams, S. Gu, N. Denisenko, M. Goldman, et al., Unsupervised discovery of temporal sequences in high-dimensional datasets with applications to neuroscience, *eLife*, **8** (2019), e38471. https://doi.org/10.7554/eLife.38471

23. P. D. O'Grady, B. A. Pearlmutter, Discovering speech phones using convolutive non-negative matrix factorization with a sparseness constraint, *Neurocomputing*, **72** (2008), 88–101. https://doi.org/10.1016/j.neucom.2008.01.033

24. P. D. O'Grady, B. A. Pearlmutter, Convolutive non-negative matrix factorization with a sparseness constraint, in *16th IEEE Signal Processing Society Workshop on Machine Learning for Signal Processing*, IEEE, (2006), 427–432. https://doi.org/10.1109/MLSP.2006.275588

25. L. Zhang, C. Jia, X. Zhang, G. Min, L. Zeng, Speech enhancement based on convolutive nonnegative matrix factorization with sparseness constraints (in Chinese), *J. Data Acquis. Process.*, **29** (2014), 259–264. https://doi.org/10.16337/j.1004-9037.2014.02.016

26. C. Lu, M. Tian, J. Zhou, H. Wang, L. Tao, A single-channel speech enhancement approach using convolutive non-negative matrix factorization with $L_{1/2}$ sparse constraint (in Chinese), *Acta Acust.*, **42** (2017), 377–384. https://doi.org/10.15949/j.cnki.0371-0025.2017.03.016

27. F. Wang, H. Ke, H. Ma, Y. Tang, Deep wavelet temporal-frequency attention for nonlinear fMRI factorization in ASD, *Pattern Recogn.*, **165** (2025), 111543. https://doi.org/10.1016/j.patcog.2025.111543
28. F. Wang, H. Ke, C. Cai, Deep wavelet self-attention non-negative tensor factorization for non-linear analysis and classification of fMRI data, *Appl. Soft Comput.*, **182** (2025), 113522. https://doi.org/10.1016/j.asoc.2025.113522
29. F. Wang, H. Ke, Y. Tang, Fusion of generative adversarial networks and non-negative tensor decomposition for depression fMRI data analysis, *Inf. Process. Manage.*, **62** (2025), 103961. https://doi.org/10.1016/j.ipm.2024.103961
30. H. Ke, F. Wang, H. Bi, H. Ma, G. Wang, B. Yin, Unsupervised deep frequency-channel attention factorization to non-linear feature extraction: A case study of identification and functional connectivity interpretation of Parkinson's disease, *Expert Syst. Appl.*, **243** (2024), 122853. https://doi.org/10.1016/j.eswa.2023.122853
31. H. Ke, D. Chen, Q. Yao, Y. Tang, J. Wu, J. Monaghan, et al., Deep factor learning for accurate brain neuroimaging data analysis on discrimination for structural MRI and functional MRI, *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **21** (2024), 582–595. https://doi.org/10.1109/TCBB.2023.3252577
32. X. Fu, K. Huang, N. D. Sidiropoulos, W. K. Ma, Nonnegative matrix factorization for signal and data analytics identifiability algorithms and application, *IEEE Signal Process. Mag.*, **36** (2019), 59–80. https://doi.org/10.1109/MSP.2018.2877582
33. P. Smaragdis, Convolutive speech bases and their application to supervised speech separation, *IEEE Trans. Audio Speech Lang. Process.*, **15** (2007), 1–12. https://doi.org/10.1109/TASL.2006.876726
34. P. Smaragdis, Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs, in *Independent Component Analysis and Blind Signal Separation*, Springer, (2004), 494–499. https://doi.org/10.1007/978-3-540-30110-3_63
35. Y. Mai, L. Lan, N. Guan, X. Zhang, Z. Luo, Transductive convolutive non-negative matrix factorization for speech separation, in *2015 4th International Conference on Computer Science and Network Technology (ICCSNT)*, IEEE, (2015), 1400–1404. https://doi.org/10.1109/ICCSNT.2015.7490990
36. D. D. Lee, H. S. Seung, Algorithms for non-negative matrix factorization, in *Proceedings of the 14th International Conference on Neural Information Processing Systems*, MIT Press, (2001), 535–541.
37. C. J. Lin, On the convergence of multiplicative update algorithms for nonnegative matrix factorization, *IEEE Trans. Neural Networks*, **18** (2007), 1589–1596. https://doi.org/10.1109/TNN.2007.895831
38. P. Paatero, U. Tapper, Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values, *Environmetrics*, **5** (1994), 111–126. https://doi.org/10.1002/env.3170050203
39. Y. Zhang, An alternating direction algorithm for nonnegative matrix factorization, in *Technical Report TR10-03, CAAM, Rice University*, (2010), 1–14.

40. D. L. Sun, C. Fevotte, Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence, in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, (2014), 6201–6205. https://doi.org/10.1109/ICASSP.2014.6854796
41. W. Deng, W. Yin, On the global and linear convergence of the generalized alternating direction method of multipliers, *J. Sci. Comput.*, **66** (2016), 889–916. https://doi.org/10.1007/s10915-015-0048-x
42. R. Glowinski, A. Marroco, Sur lapproximation, par elements finis dordre un, et la resolution, par penalisation-dualite dune classe de problemes de dirichlet non lineaires, *Rev. Fr. Autom. Inf. Rech. Oper. Anal. Numer.*, **9** (1975), 41–76.
43. D. Gabay, B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximation, *Comput. Math. Appl.*, **2** (1976), 17–40. https://doi.org/10.1016/0898-1221(76)90003-1
44. Y. Li, R. Wang, Y. Fang, M. Sun, Z. Luo, Alternating direction method of multipliers for convolutive non-negative matrix factorization, *IEEE Trans. Cybern.*, **53** (2023), 7735–7748. https://doi.org/10.1109/TCYB.2022.3204723
45. T. Xie, H. Zhang, R. Liu, H. Xiao, Accelerated sparse non-negative matrix factorization for unsupervised feature learning, *Pattern Recognit. Lett.*, **156** (2022), 46–52. https://doi.org/10.1016/j.patrec.2022.01.020
46. Y. Y. Xu, W. T. Yin, Z. W. Wen, Y. Zhang, An alternating direction algorithm for matrix completion with nonnegative factors, *Front. Math. China*, **7** (2012), 365–384. https://doi.org/10.1007/s11464-012-0194-5
47. J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, et al., TIMIT acoustic-phonetic continuous speech corpus LDC93S1, *Ling. Data Consortium*, 1993. https://doi.org/10.35111/17gk-bn40