

## Research Article

# Graph Sparse Nonnegative Matrix Factorization Algorithm Based on the Inertial Projection Neural Network

Xiangguang Dai <sup>1</sup>, Chuandong Li <sup>1</sup>, and Biqun Xiang<sup>2</sup>

<sup>1</sup>College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China

<sup>2</sup>College of Mobile Telecommunications, Chongqing University of Posts and Telecommunications, Chongqing 401520, China

Correspondence should be addressed to Chuandong Li; [cdli@swu.edu.cn](mailto:cdli@swu.edu.cn)

Received 3 May 2017; Accepted 13 December 2017; Published 19 March 2018

Academic Editor: Pietro De Lellis

Copyright © 2018 Xiangguang Dai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present a novel method, called graph sparse nonnegative matrix factorization, for dimensionality reduction. The affinity graph and sparse constraint are further taken into consideration in nonnegative matrix factorization and it is shown that the proposed matrix factorization method can respect the intrinsic graph structure and provide the sparse representation. Different from some existing traditional methods, the inertial neural network was developed, which can be used to optimize our proposed matrix factorization problem. By adopting one parameter in the neural network, the global optimal solution can be searched. Finally, simulations on numerical examples and clustering in real-world data illustrate the effectiveness and performance of the proposed method.

## 1. Introduction

Dimensionality reduction plays a fundamental role in image processing, and many researchers have been seeking effective methods to solve this problem. For a given image database, there are many distinct features, whereas the available features are far less enough. Thus, it is of great significance to find useful features with low-dimensionality to represent the original feature space. For this purpose, matrix factorization techniques have attracted great attention in recent decades [1–3], and many different methods have been developed by using different criteria. The most familiar methods include Singular Value Decomposition (SVD) [4], Principal Component Analysis (PCA) [5], and Vector Quantization (VQ) [6]. The main idea of matrix factorization methods is finding several matrices whose product approximates to the original matrix. In dimensionality reduction, the dimension of the decomposed matrices is smaller than that of the original matrix. This gives rise to a low-dimensional compact representation of the original data points, which can facilitate clustering or classification.

Among these matrix factorization methods, one of the most used methods is nonnegative matrix factorization (NMF) [3], which requires the decomposed matrices to

be nonnegative. The effect of the nonnegative constraint leads NMF to learn a part-based representation of high-dimensional data, and it is applied to so many areas such as signal processing [7], data mining [8, 9], and computer vision [10]. In general, NMF is shown to be effective for unsupervised learning problems, but not applicable to supervised learning problems. To overcome this problem, some researchers [11–13] have presented semi-supervised learning theory to achieve better performance in dimensionality reduction. In the light of locality preserving projection, a graph regularized nonnegative matrix factorization method (GNMF) has been proposed to impose the geometrical information on the data space. The geometrical structure is constructed by a nearest neighbor graph [11]. Based on the idea of label propagation, Liu et al. [13] imposed the label information constraint into nonnegative matrix factorization (CNMF). The idea of CNMF is that the neighboring data points with the same class are supposed to merge together in the low-dimensional representation space.

Motivated by previous researches in matrix factorization, in this paper, we propose a novel method, called graph sparse nonnegative matrix factorization (GSNMF), for dimensionality reduction, which can be used for semi-supervised learning problems. In GSNMF, a sparse constraint is imposed

on GNMF, and this leads matrix factorization to learn a sparse part-based representation of the original data space. The sparse constraint causes GSNMF to be a nonconvex nonsmooth problem, and traditional optimization algorithms can not be optimized directly. Recently, numerous neural networks have emerged as a powerful tool for optimization problems [14–27]. For some nonconvex problems, an inertial projection neural network (IPNN) [16] has been proposed to search different local optimal solutions by the inertial term. In [17], a shuffled frog leaping algorithm (SFLA) has been developed using the recurrent neural network. Based on the SFLA framework, the global optimal solution can be searched. Moreover, there are many optimization methods for nonconvex nonsmooth problems that use neural networks [22–27].

It is worth highlighting some advantages of our proposed method as follows:

- (i) Traditional algorithms for GNMF [11] and NMF [3] can easily trap into local optimum solution, and these algorithms are sensitive to initial values, while our proposed algorithm using inertial projection neural network can avoid these problems.
- (ii) Our proposed algorithm can be initialized by sparser matrices; however, GNMF and NMF may fail in this case.
- (iii) By adopting one parameter in the neural network, GSNMF has the better clustering effect than GNMF and NMF.

The rest of the paper is organized as follows. In Section 2, some related works to NMF are briefly reviewed; then we introduce GSNMF. Section 3 reviews the inertial projection neural network theory and provides a convergence proof to GSNMF. Section 4 presents numerical examples to demonstrate the validity of our proposed algorithm. Experiments on clustering are given in Section 5. Finally, we present some concluding marks and future work in Section 6.

## 2. Problem Formulation

To find the effective features of high dimensionality data, matrix factorization can be used to learn sets of features to represent data. Given a data matrix  $Y = [Y_1, \dots, Y_n] \in R^{m \times n}$  and an integer  $k$ , matrix factorization is to find two matrices  $A \in R^{m \times k}$  and  $S = [S_1, \dots, S_n] \in R^{k \times n}$  such that

$$Y \approx AS. \quad (1)$$

When  $k \ll \min(m, n)$ , the matrix factorization method can be regarded as a dimensionality reduction method. In image dimensionality reduction, each column of  $S$  is a basis vector to capture the original image data and each column of  $A$  is the representation with respect to the new basis. The most used method to measure the approximation is the Frobenius norm in the following form:

$$\min_{A, S} f(A, S) = \frac{1}{2} \|Y - AS\|_F^2$$

$$= \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^n \left( y_{ij} - \sum_{k=1}^r a_{ik} s_{kj} \right)^2. \quad (2)$$

Different matrix factorization methods imposed different constraints on (2) that can solve different practical problems. At present, the most used matrix factorization method is nonnegative matrix factorization (NMF) [3] with nonnegative constraints on  $A$  and  $S$ . The classic algorithm is summarized as follows:

$$a_{ik} = a_{ik} \frac{(YS^T)_{ik}}{(ASS^T)_{ik}},$$

$$s_{kj} = s_{kj} \frac{(A^T Y)_{kj}}{(A^T AS^T)_{kj}}. \quad (3)$$

Recently, Cai et al. [11] proposed a graph regularized nonnegative matrix factorization method (GNMF) and incorporated the geometrical information into the data space. The goal of GNMF is to find effective basis vectors to represent the intrinsic structure. The research has presented the natural assumption that if two data points  $y_j$  and  $y_l$  from  $Y$  are close in the intrinsic geometry of the data distribution, new representations of two points  $s_j$  and  $s_l$  are also close to each other. For each data point  $y_j$ , we find its  $p$  nearest neighbors and put edges between  $y_j$  and its neighbors. Edges between each data points  $y_j$  can be considered as the weight matrix  $W$ . If nodes  $j$  and  $l$  are connected by an edge, then  $W_{jl} = 1$ .  $W$  can be described by

$$W_{jl} = \begin{cases} 1, & \text{if } y_j \text{ and } y_l \text{ have the same label,} \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

The low-dimensional representation of  $y_j$  with respect to the new basis is  $s_j$ . The Euclidean distance is used to measure the dissimilarity between  $s_j$  and  $s_l$  by

$$d(s_j, s_l) = \|s_j - s_l\|^2. \quad (5)$$

With the above analysis, the following term is used to measure the smoothness of the low-dimensional representation:

$$R = \frac{1}{2} \sum_{j,l=1}^n \|s_j - s_l\|^2 W_{jl} = \sum_{j=1}^n s_j^T s_j F_{jj} - \sum_{j,l=1}^n s_j^T s_l W_{jl}$$

$$= \text{tr}(SFS^T) - \text{tr}(SWS^T) = \text{tr}(SLS^T), \quad (6)$$

where  $\text{tr}(\cdot)$  denotes the trace of a matrix,  $F_{jj} = \sum_{l=1}^n W_{jl}$ , and  $L = F - W$ . Combining (6) and (2), the new objective function is defined by the Euclidean distance.

$$\min_{A \geq 0, S \geq 0} f(A, S) = \frac{1}{2} \|Y - AS\|_F^2 + \alpha \text{tr}(SLS^T). \quad (7)$$

The algorithm to solve (7) is presented as follows:

$$\begin{aligned} a_{ik} &= a_{ik} \frac{(YS^T)_{ik}}{(ASS^T)_{ik}}, \\ s_{kj} &= s_{kj} \frac{(A^T Y + SW^T)_{kj}}{(A^T AS^T + SF^T)_{kj}}. \end{aligned} \quad (8)$$

When  $\alpha = 0$  or  $W = 0$ , GNMF is equivalent to non-negative matrix factorization. In the representation of the image data, GNMF and NMF only consider the Euclidean structure of image space. However, recent researches have shown that human generated images may from a submanifold of the ambient Euclidean space [28, 29]. In general, the human generated images cannot uniformly fill up the high-dimensional Euclidean space. Therefore, the matrix factorization should respect the intrinsic manifold structure and learn the sparse basis to represent the image data. In the light of sparse coding [30], we impose the sparse constraint on (7) and the optimization problem is transformed into another form:

$$\begin{aligned} \min_{A \geq 0, S \geq 0} f(A, S) \\ = \frac{1}{2} \|Y - AS\|_F^2 + \alpha \text{tr}(SLS^T) + \beta \Phi(S). \end{aligned} \quad (9)$$

Because the optimization problem (9) is nonconvex, a block-coordinate update (BCD) [31] structure is proposed to optimize GSNMF. Given the initial  $A$  and  $S$ , BCD alternatively solves

$$\begin{aligned} \min_{S^*} f(S) &= \frac{1}{2} \|Y - AS\|_F^2 + \alpha \text{tr}(SLS^T) + \beta \Phi(S) \\ \text{s.t. } S &\geq 0, \end{aligned} \quad (10)$$

$$\begin{aligned} \min_{A^*} f(A) &= \frac{1}{2} \|Y^T - (S^*)^T A^T\|_F^2 \\ \text{s.t. } A &\geq 0 \end{aligned} \quad (11)$$

until convergence. Since (11) and (10) have a similar form, we only consider how to solve (10); then (11) can be solved accordingly. The problem (10) can be transformed into the following vector form:

$$\begin{aligned} \min_H f(H) \\ = \frac{1}{2} \sum_{i=1}^n \|Y_i - AH_i\|_2^2 + \frac{1}{2} \alpha \sum_{j,l=1}^n \|H_j - H_l\|_2^2 W_{jl} \\ + \beta \|H\|_1 \\ \text{s.t. } H \geq 0, \end{aligned} \quad (12)$$

where

$$H = \begin{bmatrix} H_1 \\ \vdots \\ H_n \end{bmatrix} = \begin{bmatrix} S_1 \\ \vdots \\ S_n \end{bmatrix} \in \mathbb{R}^{nk}, \quad H_i \in \mathbb{R}^k, \quad Y_i \in \mathbb{R}^m. \quad (13)$$

It is evident that  $L_1$ -norm is not differentiable. However, [32] has presented a method to solve it. Supposing  $H = U - V$ , problem (12) can be rewritten as follows:

$$\begin{aligned} \min_{U, V} f(U, V) \\ = \frac{1}{2} \sum_{i=1}^n \|Y_i - A(U_i - V_i)\|_2^2 \\ + \frac{1}{2} \alpha \sum_{j,l=1}^n \|(U_j - V_j) - (U_l - V_l)\|_2^2 W_{jl} \\ + \beta \|U - V\|_1 \\ \text{s.t. } U - V \geq 0, \end{aligned} \quad (14)$$

where  $U_i = (H_i)_+ \in \mathbb{R}^k$ ,  $V_i = (-H_i)_+ \in \mathbb{R}^k$ ,  $(H_i)_+ = \max(H_i, 0)$ ,  $i = 1, \dots, n$ ,  $1_{nk} = [1, \dots, 1]^T$ , and  $U$  and  $V$  are, respectively, defined as follows:

$$\begin{aligned} U &= \begin{bmatrix} U_1 \\ \vdots \\ U_n \end{bmatrix} \in \mathbb{R}^{nk}, \\ V &= \begin{bmatrix} V_1 \\ \vdots \\ V_n \end{bmatrix} \in \mathbb{R}^{nk}. \end{aligned} \quad (15)$$

According to the BCD structure, (14) can be separated into two subproblems. Given the initial  $U$  and  $V$ , one alternatively solves

$$\begin{aligned} \min_U f(U^*) \\ = \frac{1}{2} \sum_{i=1}^n \|Y_i + AV_i - AU_i\|_2^2 \\ + \frac{1}{2} \alpha \sum_{j,l=1}^n \|(U_j - U_l) - (V_j - V_l)\|_2^2 W_{jl} \\ + \beta \|U\|_1 \end{aligned} \quad (16)$$

$$\text{s.t. } U \geq V,$$

$$\begin{aligned} \min_V f(V^*) \\ = \frac{1}{2} \sum_{i=1}^n \|Y_i - AU_i - (-AV_i)\|_2^2 \\ + \frac{1}{2} \alpha \sum_{j,l=1}^n \|(V_j - V_l) - (U_j - U_l)\|_2^2 W_{jl} \\ + \beta \|V\|_1 \\ \text{s.t. } 0 \leq V \leq U \end{aligned} \quad (17)$$

until convergence. Since (16) and (17) have a similar form, we only consider how to solve (16); then (17) can be solved accordingly. Equation (16) can be transformed into the following convex quadratic program (CQP):

$$\begin{aligned} \min_U \quad & f(U) = \frac{1}{2}U^T C U + D U \\ \text{s.t.} \quad & U \in \Omega, \end{aligned} \quad (18)$$

where

$$E = \text{diag}(A^T A, \dots, A^T A) \in R^{nk \times nk},$$

$$F_{ij} = \text{diag}(W_{ij}, \dots, W_{ij}) \in R^{n \times n},$$

$$F = \begin{pmatrix} F_{11} & F_{12} & \cdots & F_{1k} \\ F_{21} & F_{22} & \cdots & F_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ F_{k1} & F_{k2} & \cdots & F_{kk} \end{pmatrix}, \quad (19)$$

$$G = \text{diag}(F_{11}, F_{12}, \dots, F_{kk})$$

$$C = E - \alpha F + \alpha G$$

$$\begin{aligned} D = \beta 1_{nk}^T &- (Y_1^T A + V_1^T A^T A, \dots, Y_n^T A + V_n^T A^T A) \\ &- 2\alpha (G V^T - F V^T) \end{aligned}$$

$$\Omega = \{U \mid U \geq V\}.$$

According to the above analysis, problem (11) can be also transformed into a convex quadratic program problem. For saving space, we do not provide the derivation process. In the following section, we will introduce IPNN to optimize (18).

### 3. Neural Network Model and Analysis

**3.1. Inertial Projection Neural Network.** To solve problem (18), we establish the following neural network using IPNN [16]:

$$\begin{aligned} \frac{dU}{dt} &= X, \\ \frac{dX}{dt} &= -\lambda X + P_\Omega(U - \nabla f(U)) - U, \\ U(t_0) &= U_0 \in \Omega, \end{aligned} \quad (20)$$

where  $\nabla f(U) = C U + D^T$ . Now, we are ready to show the convergence and optimality of (20). For any  $U \in R^{2nk}$  and  $X \in R^{2nk}$ , we set  $Q_1 = (U_1^T, X_1^T)^T$ ,  $Q_2 = (U_2^T, X_2^T)^T$ , and  $R(Q_i) = (-\lambda X_i + P_\Omega(U_i - \nabla f(U_i)) - U_i)$ ; then we will present the following theorems.

**Theorem 1.** *For any initial point  $Q_0 \in \Omega$  with the initial condition  $Q(0) = Q_0$ , there exists unique continuous solution  $Q(t) \in \Omega$ , where  $t \in [t_0, +\infty)$ .*

*Proof.* Note that  $\nabla f(U) = C U + D^T$  is Lipschitz continuous. Let  $L$  be the Lipschitz constant. Thus, for any  $Q_1$  and  $Q_2 \in \Omega$ , we obtain

$$\begin{aligned} &\|R(Q_1) - R(Q_2)\| \\ &\leq \|X_1 - X_2\| + \lambda \|X_1 - X_2\| + \|U_1 - U_2\| \\ &\quad + \|P_\Omega(U_1 - \nabla f(U_1)) - P_\Omega(U_2 - \nabla f(U_2))\| \\ &\leq (2 + \lambda)(Q_1 - Q_2) + \|(I - C)(U_1 - U_2)\| \\ &\leq (2 + \lambda + \|I - C\|)(Q_1 - Q_2), \end{aligned} \quad (21)$$

where  $I$  and  $L$  are unit matrix and Lipschitz constant, respectively. Therefore,  $R(Q)$  is Lipschitz continuous on  $\Omega$ . There exists unique solution  $Q(t)$  with initial condition  $Q(t_0)$  by the local existence theorem of solution to ordinary differential equations.  $\square$

**Theorem 2.** *Define  $N(U) = U - P_\Omega(U - F(U))$ , if the following two conditions hold.*

(1) *For any  $U_1$  and  $U_2 \in R^{nk}$ ,*

$$\begin{aligned} &\langle N(U_1) - N(U_2), U_1 - U_2 \rangle \\ &\geq \frac{\cos \theta}{(1 + \|I - C\|)} \|N(U_1) - N(U_2)\|, \end{aligned} \quad (22)$$

where  $\theta$  is the angle between  $N(U_1) - N(U_2)$  and  $U_1 - U_2$ .

(2)  $\lambda > \sqrt{(1 + \|I - C\|) / \cos \theta}$ . Then, the solution of model (20) converges to optimal solution set of (18).

*Proof.* Considering  $U^* \in \Omega$  and the Lyapunov function  $V(t) = (1/2)\|U(t) - U^*\|^2 = (1/2)(U(t) - U^*)^T (U(t) - U^*)$ , we obtain

$$\begin{aligned} &\dot{V}(t) + \lambda \dot{V}(t) \\ &= X^T X + (U - U^*)^T \dot{X} + \lambda (U(t) - U^*)^T X \\ &= \|X\|^2 + \langle U(t) - U^*, \dot{X} + \lambda X \rangle \\ &= \|X\|^2 + \langle U(t) - U^*, P_\Omega(U - \nabla f(U) - U) \rangle \\ &= \|X\|^2 + \langle U(t) - U^*, -N(U) \rangle. \end{aligned} \quad (23)$$

Since  $N(U^*) = 0$  and Condition (1) holds, (23) can be rewritten as

$$\begin{aligned} \|X\|^2 &= \dot{V}(t) + \lambda \dot{V}(t) + \langle U(t) - U^*, N(U) \rangle \\ &= \dot{V}(t) + \lambda \dot{V}(t) \\ &\quad + \langle U(t) - U^*, N(U) - N(U^*) \rangle \end{aligned}$$

$$\begin{aligned}
&\geq \dot{V}(t) + \lambda \dot{V}(t) + \frac{\cos \theta}{(1 + \|I - C\|)} \|N(U)\|^2 \\
&= \dot{V}(t) + \lambda \dot{V}(t) + \frac{\cos \theta}{(1 + \|I - C\|)} \|\dot{X} + \lambda X\|^2 \\
&= \dot{V}(t) + \lambda \dot{V}(t) + \frac{\cos \theta}{(1 + \|I - C\|)} \|\dot{X}\|^2 \\
&\quad + \frac{\lambda \cos \theta}{(1 + \|I - C\|)} \frac{d\|X\|^2}{dt} \\
&\quad + \frac{\lambda^2 \cos \theta}{(1 + \|I - C\|)} \|X\|^2.
\end{aligned} \tag{24}$$

Then, (24) can be transformed into

$$\begin{aligned}
&\dot{V}(t) + \lambda \dot{V}(t) + \frac{\cos \theta}{(1 + \|I - C\|)} \|\dot{X}\|^2 \\
&\quad + \frac{\lambda \cos \theta}{(1 + \|I - C\|)} \frac{d\|X\|^2}{dt} \\
&\quad + \left( \frac{\lambda^2 \cos \theta}{(1 + \|I - C\|)} - 1 \right) \|X\|^2 \leq 0,
\end{aligned} \tag{25}$$

which indicates that the function  $Z(t) = \dot{V}(t) + \lambda V(t) + (\lambda \cos \theta / (1 + \|I - C\|)) \|X\|^2 + (\cos \theta / (1 + \|I - C\|)) \int_0^t \|\dot{X}(s)\|^2 ds + (\lambda^2 \cos \theta / (1 + \|I - C\|) - 1) \int_0^t \|X(s)\|^2 ds$  is monotone nonincreasing. Thus, for any  $t > 0$ ,

$$\begin{aligned}
Z(t) &\leq Z_0 = \dot{V}(0) + \lambda V(0) + \frac{\cos \theta}{(1 + \|I - C\|)} \|X(0)\|^2 \\
&= (U(0) - U^*)^T X(0) + \frac{\lambda}{2} \|U(0) - U^*\|^2 \\
&\quad + \frac{\lambda \cos \theta}{(1 + \|I - C\|)} \|X(0)\|^2.
\end{aligned} \tag{26}$$

Since  $\lambda > \sqrt{(1 + \|I - C\|) / \cos \theta}$ , we obtain  $(\lambda^2 \cos \theta / (1 + \|I - C\|) - 1) > 0$ . Further

$$\dot{V}(t) + \lambda V(t) \leq Z_0. \tag{27}$$

By multiplying the inequality (27) into  $e^{\lambda t}$ , we obtain  $\dot{V}(t)e^{\lambda t} + \lambda V(t)e^{\lambda t} \leq Z_0 e^{\lambda t}$  which implies that

$$\frac{d(V(t)e^{\lambda t})}{dt} \leq Z_0 e^{\lambda t}. \tag{28}$$

Integrating (28) from 0 to  $t$ , it is obtained that  $V(t) \leq (V(0) - 1/\lambda Z_0)e^{(-\lambda t)} + 1/\lambda Z_0$ . Therefore, the trajectory of model (20) is bounded.

Since  $V(t)$  is bounded and inequality (26) holds, we obtain

$$\dot{V}(t) + \frac{\cos \theta}{(1 + \|I - C\|)} \|X(t)\| \leq Z_0. \tag{29}$$

Thus, (29) can be rewritten as

$$\langle U(t) - U^*, X(t) \rangle + \frac{\cos \theta}{(1 + \|I - C\|)} \|X(t)\|^2 \leq Z_0. \tag{30}$$

Since  $\|U(t) - U^*\|$  is bounded, (30) indicates that  $\|X(t)\|^2$  is also bounded. From (25) and (30), one obtains  $\int_0^{+\infty} \|X(s)\|^2 ds < +\infty$  and  $\int_0^{+\infty} \|\dot{X}(s)\| ds < +\infty$ .

Assuming  $\int_0^{+\infty} \|X(s)\|^2 ds = Q < +\infty$ , and since  $\int_0^{+\infty} \|\dot{X}(s)\|^2 ds < +\infty$ , it implies that there exists  $P > 0$ , such that  $\|\dot{X}(t)\| < P$  for any  $t \in (0, +\infty)$ . Therefore, one obtains

$$\int_0^{+\infty} \|X(s)\|^2 \|\dot{X}(s)\| ds \leq P \int_0^{+\infty} \|X(s)\|^2 ds < +\infty, \tag{31}$$

and it is easy to obtain that

$$\begin{aligned}
&\int_0^{+\infty} \|X(s)\|^2 \|\dot{X}(s)\| ds \\
&\leq \frac{1}{3} \lim_{t \rightarrow +\infty} [\|X(t)\|^3 - \|X(0)\|^3] < +\infty.
\end{aligned} \tag{32}$$

According to the theory of Calculus, the value of  $\lim_{t \rightarrow +\infty} \|X(t)\|^3$  exists. Therefore, the value of  $\lim_{t \rightarrow +\infty} \|X(t)\|$  also exists. Since  $\int_0^{+\infty} \|X(s)\|^2 ds < +\infty$ , we obtain  $\lim_{t \rightarrow +\infty} \|X(t)\| = 0$ . Since  $V(t)$  is bounded, we have  $\lim_{t \rightarrow +\infty} \dot{X}(t) = 0$ .

It follows from (20) that

$$\lim_{t \rightarrow +\infty} (\dot{X} + U - P_\Omega(U - \nabla f(U))) = 0. \tag{33}$$

If  $\lim_{t \rightarrow +\infty} \dot{X}(t) = 0$ , then

$$\lim_{t \rightarrow +\infty} (U - P_\Omega(U - \nabla f(U))) = 0, \tag{34}$$

which implies that  $\lim_{t \rightarrow +\infty} U(t) = U^*$ . In the following, we will prove  $\lim_{t \rightarrow +\infty} \dot{X}(t) = 0$ .

Defining  $r_\varepsilon(t) = (1/\varepsilon)(X(t + \varepsilon) - X(t))$ , and substituting  $r_\varepsilon(t)$  into (20), one obtains

$$\dot{r}_\varepsilon(t) + \lambda r_\varepsilon(t) = -\frac{1}{\varepsilon} N(U(t + \varepsilon)) + \frac{1}{\varepsilon} N(U(t)). \tag{35}$$

Since Condition (1) holds, we have

$$\begin{aligned}
&\|N(U(t + \varepsilon)) - N(U(t))\| \\
&\quad \cdot \|N(U(t + \varepsilon)) - N(U(t))\| \\
&\leq \frac{\varepsilon(1 + \|I - C\|)}{\cos \theta} \|N(U(t + \varepsilon)) - N(U(t))\| \\
&\quad \cdot \left\| \frac{U(t + \varepsilon) - U(t)}{\varepsilon} \right\|.
\end{aligned} \tag{36}$$

Hence, we get

$$\begin{aligned}
&\|N(U(t + \varepsilon)) - N(U(t))\| \\
&\leq \frac{\varepsilon(1 + \|I - C\|)}{\cos \theta} \sup_{s \in [t, t + \varepsilon]} \|X(s)\|.
\end{aligned} \tag{37}$$

**Input:**  $Y, A, V, W, \alpha, \beta$   
**Output:**  $D, E, F, G$

- (1) Calculate the number of rows  $k$  and columns  $n$  in the matrix  $V$
- (2) Calculate  $E$  with  $E \leftarrow A^T A$
- (3) Calculate  $F$  with  $F \leftarrow \alpha W$
- (4) Calculate the diagonal matrix  $G$  with  $G_{ii} \leftarrow \alpha \sum_{j=1}^n W_{ij}$
- (5) Calculate  $D$  with  $D \leftarrow \beta 1_{n \times k} - (Y^T A + V^T A^T A) - 2\alpha(GV^T - FV^T)$

ALGORITHM 1: Parameters.

Integrating (35), it gives  $\lim_{t \rightarrow +\infty} \sup \|r_\varepsilon(t)\| = 0$ . Since  $\|\dot{X}(t)\| \leq \sup \|r_\varepsilon(t)\|$ , we obtain  $\lim_{t \rightarrow +\infty} \dot{X}(t) = 0$ . Thus, the solution of system (20) converges to the optimal set  $\Omega$ . The proof is completed.  $\square$

*Remark 3.* In Theorem 2, Condition (1) should be satisfied. In the following, we discuss the existence of parameter  $\cos \theta / (1 + \|I - C\|)$ . For any  $U_1$  and  $U_2$ , we have

$$\begin{aligned} \|N(U_1) - N(U_2)\| &= \|U_1 - P_\Omega(U_1 - \nabla f(U_1)) - U_2 \\ &\quad + P_\Omega(U_2 - \nabla f(U_2))\| \leq \|U_1 - U_2\| \\ &\quad + \|P_\Omega(U_1 - \nabla f(U_1)) - P_\Omega(U_2 - \nabla f(U_2))\| \leq \|U_1 \\ &\quad - U_2\| + \|(I - C)(U_1 - U_2)\| \leq (1 + \|I - C\|) \|U_1 \\ &\quad - U_2\|. \end{aligned} \quad (38)$$

Therefore,  $N(U)$  is also Lipschitz continuous and

$$\begin{aligned} \|N(U_1) - N(U_2)\|^2 & \\ &\leq (1 + \|I - C\|) \|N(U_1) - N(U_2)\| \|U_1 - U_2\|. \end{aligned} \quad (39)$$

Thus, we get

$$\begin{aligned} \|N(U_1) - N(U_2)\|^2 & \\ &\leq \frac{(1 + \|I - C\|)}{\cos \theta} \langle N(U_1) - N(U_2), U_1 - U_2 \rangle, \end{aligned} \quad (40)$$

where  $\theta$  is the angle between  $N(U_1) - N(U_2)$  and  $U_1 - U_2$ . It is easy to obtain

$$\begin{aligned} \langle N(U_1) - N(U_2), U_1 - U_2 \rangle & \\ &\geq \frac{\cos \theta}{(1 + \|I - C\|)} \|N(U_1) - N(U_2)\|. \end{aligned} \quad (41)$$

**3.2. Algorithms.** Based on the above analysis, we summarize Algorithms 1, 2, and 3 to optimize GSNMF. Firstly, the parameters  $D, E, F$ , and  $G$  can be derived by Algorithm 1. Secondly, Algorithm 2 applies IPNN to optimize the CQP problem. Thirdly, the optimization problem (9) is divided into two CQP problems which are solved alternatively by Algorithm 3. In the following, we analyse the time complexity of our proposed algorithms. The main cost of our proposed algorithms is spent on the calculation of the gradient  $\nabla f$  in

**Input:**  $U, D, E, F, G, h, \lambda, \Omega, K_1$   
**Output:**  $U^*$

**Initialization:**  $X, U, i \leftarrow 0$

**repeat**

- (1)  $\nabla f \leftarrow EU - UF + UG + D^T$
- (2)  $U^{i+1} \leftarrow X * h + U$
- (3)  $X^{i+1} \leftarrow (-\lambda X + P_\Omega(U - \nabla f) - U) * h + X$
- (4)  $U = U^{i+1}$
- (5)  $X = X^{i+1}$
- (6)  $i \leftarrow i + 1$

**until**  $i \leq K_1$   
 $U^* = U$

ALGORITHM 2: IPNN.

TABLE 1: Time complexity of GSNMF, GNMF, and NMF.

Solver	Time complexity
GSNMF	$O(mnk) + K1 \times O(mk^2 + nk^2 + kn^2)$
GNMF	$O(mnk + mk^2 + nk^2 + kn^2)$
NMF	$O(mnk + mk^2 + nk^2)$

Algorithm 2. To optimize subproblem (10), the operation in Algorithm 2 is the matrix product  $EU - UF + UG$ , which takes  $O(nk^2 + n^2k)$ . With the cost  $O(mnk)$  for calculating  $D^T$ , the complexity of using Algorithm 2 to optimize the subproblem (10) is

$$O(mnk) + K1 \times O(nk^2 + n^2k). \quad (42)$$

Similarly, the complexity of using Algorithm 2 to optimize the subproblem (11) is

$$O(mnk) + K1 \times O(mk^2). \quad (43)$$

Hence, the overall cost to solve GSNMF is

$$O(mnk) + K1 \times O(mk^2 + nk^2 + kn^2). \quad (44)$$

We summarize the time complexity of one iteration round of GSNMF, GNMF, and NMF in Table 1. At each iteration, there are two  $O(mnk)$  operations, same as NMF and GNMF.

## 4. Numerical Examples

In this section, we exhibit the global searching ability of GSNMF. By adjusting the inertial term  $\lambda$  in the neural

**Input:**  $Y \in R^{m \times n}$ ,  $A \in R^{m \times k}$ ,  $S \in R^{k \times n}$ ,  $K_2$   
**Output:**  $A^*$ ,  $S^*$   
**Initialization:**  $W$  satisfies (6),  $\Omega_1 = \{U \mid U \geq V\}$ ,  $\Omega_2 = \{V \mid 0 \leq V \leq U\}$ ,  $\Omega_3 = \{U \mid U \geq 0\}$ ,  $i \leftarrow 0$   
**repeat**  
(1)  $U = \max(S, 0)$ ,  $V = \max(S, 0)$   
(2)  $[D, E, F, G] \leftarrow \text{PARAMETER}(Y, A, V, W, \alpha, \beta)$   
(3)  $U \leftarrow \text{IPNN}(U, D, E, F, G, h, \lambda, \Omega_1, K_1)$   
(4)  $[D, E, F, G] \leftarrow \text{PARAMETER}(Y, -A, -U, W, \alpha, \beta)$   
(5)  $V \leftarrow \text{IPNN}(V, D, E, F, G, h, \lambda, \Omega_2, K_1)$   
(6)  $[D, E, F, G] \leftarrow \text{PARAMETER}(Y^T, S^T, 0, 0, 0, 0)$   
(7)  $A \leftarrow \text{IPNN}(A^T, D, E, F, G, h, \lambda, \Omega_3, K_1)$   
(8)  $A \leftarrow A^T$ ,  $i \leftarrow i + 1$   
**until**  $i \leq K_2$   
 $A^* = A$ ,  $S^* = S$

ALGORITHM 3: GSNMF.

TABLE 2:  $Y$ .

4.345	1.79	2.92	0.852	6.036	6.929	2.803	2.445
0.513	1.164	8.54	8.645	0.879	12.309	4.731	1.326
4.277	3.665	5.619	2.1	6.048	6.164	1.077	4.167
5.658	5.788	0.887	4.599	7.653	2.342	7.534	7.104
2.282	0.3	12.657	3.248	4.33	1.508	1.819	1.26
1.856	7.669	2.113	4.084	7.616	0.287	2.865	1.329
0.589	6.886	2.388	0.446	7.759	2.456	0.486	7.189
0.214	16.156	3.684	1.973	0.25	3.841	2.825	1.482

TABLE 3:  $A$ .

0.797	0.394	0.001	1.029	0.182
0.3	0.633	0.28	1.14	0.727
0.81	0.078	1.589	1.764	1.152
0.062	0.408	0.265	0.78	0.716
0.073	0.389	0.005	1.768	0.166
1.921	1.45	1.474	0.673	0.016
0.346	1.896	0.284	1.135	0.087
1.849	0.058	0.445	1.574	1.745

network, different local optimal solutions can be searched. Let  $m = 8$ ,  $n = 8$ ,  $k = 5$ ,  $1 \leq \lambda \leq 10$ ,  $\alpha = 0$ ,  $\beta = 0$ ,  $K_1 = 5$ ,  $K_2 = 500$ , and  $h = 0.2$ . In order to ensure the validity of this experiment, we provide the initial  $Y$ ,  $A$ , and  $S$  in Tables 2, 3, and 4, respectively. Table 5 shows the comparison between GSNMF and NMF. To investigate whether GSNMF can converge, the convergence curve is depicted in Figure 1 with  $\lambda = 1$ .

## 5. Application in Image Clustering

**5.1. Databases.** To examine the clustering performance of GSNMF, we present the experiment in two databases including IRIS and COIL20. Their details are presented in the following (see also Table 6).

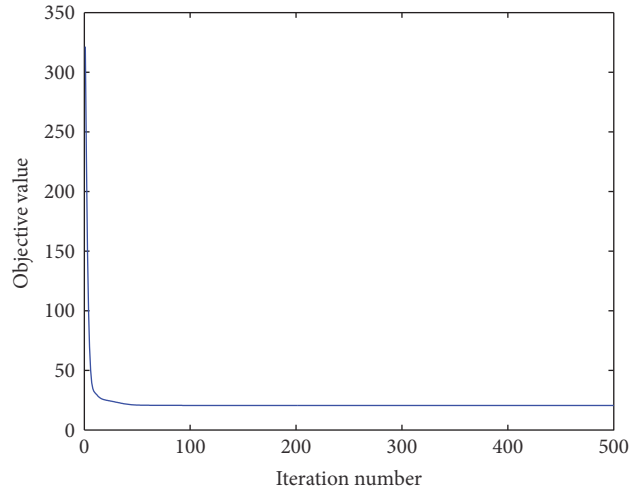
(1) *IRIS*. It includes 150 instances with 4 features. There are 3 classes including Versicolour, Setosa, and Virginca. Each class includes 50 instances.

(2) *COIL20*. This data set is an image library which contains 1440 instances with  $16 \times 16$  gray scale features. There are 20 different classes and each class contains 72 instances.

**5.2. Compared Methods.** We present the clustering performance on two databases using GSNMF and GNMF. There are two metrics including accuracy and normalized mutual information [33] to evaluate the clustering performance. To reveal the effect of the sparse constraint, different cardinalities (the number of zero entries) of the initial  $S$  are considered to evaluate the clustering performance. Because NMF is a

TABLE 4: S.

0.071	0.242	0.413	1.101	1.345	0.967	0.923	0.533
1.319	1.305	0.329	0.99	0.309	1.173	0.734	1.309
1.377	0.796	0.913	0.377	0.504	1.322	0.378	0.682
0.403	0.608	0.224	0.163	0.612	0.162	0.724	0.849
0.929	1.402	0.773	1.727	0.039	0.187	0.446	0.132

FIGURE 1: Iteration number versus objective value for GSNMF when  $\lambda = 1$ .TABLE 5: Comparisons between GSNMF using different  $\lambda$  and NMF.

	GNMF	NMF
Objective values	$\lambda = 1, 20.5172$	20.5172
	$\lambda = 3, 20.7534$	
	$\lambda = 6, 21.6804$	

TABLE 6: Two data sets.

Dataset	size ( $n$ )	Dimensionality ( $m$ )	Classes ( $k$ )
IRIS	150	4	3
COIL20	1440	256	20

nonconvex problem, different initial  $A$  and  $S$  may lead to different local optimal solutions. For the fair comparison, we try 10 different initial  $A$  and  $S$  and report the average results. We compare different methods in two cases. Firstly,  $A$  and  $S$  are randomly generated between 0 and 1. Secondly,  $A$  and  $S$  are randomly generated between  $-1$  and 1.

**5.3. Compared Results.** Tables 7 and 8 present the clustering results on IRIS in two cases. The parameters for IRIS are  $h = 0.02$ ,  $K_1 = 3$ ,  $K_2 = 200$ ,  $\alpha = 1$ ,  $\beta = 1e - 4$ , and  $0 \leq \lambda \leq 10$ . The clustering performance on COIL20 is shown in Tables 9 and 10 in two cases. The parameters for COIL20 are  $h = 0.02$ ,

$K_1 = 20$ ,  $K_2 = 200$ ,  $\alpha = 1$ ,  $\beta = 1e - 4$ , and  $0 \leq \lambda \leq 10$ . These tables reveal some interesting points:

- (i) When the sparse constraint is imposed on GNMF, the clustering performance of GSNMF is better than NMF and GNMF.
- (ii) When the initial  $A$  and  $S$  have some negative entries, NMF and GNMF fail to cluster. However, GSNMF is not affected in this case.

**5.4. Parameters Selection.** Our GSNMF has one essential parameter: the inertia term  $\lambda$ . Figures 2 and 3 depict the average performance of GSNMF with different  $\lambda$ .

**5.5. Convergence Study.** According to the BCD and IPNN theory, the method for optimizing GSNMF is proved to be convergent. Here we investigate whether this method can converge to a stationary point. Figure 4 depicts the convergence curves of GSNMF on two data sets. For each figure, the  $x$ -axis is the iteration number and the  $y$ -axis denotes the objective value.

## 6. Conclusion and Future Work

We propose a dimensionality reduction method, which can be solved by the inertial projection neural network. According to the experiments, three advantages are presented. Firstly, different local solutions can be achieved with different



TABLE 7: Clustering performance on IRIS in case 1.

Cardinality (%)	Accuracy (%)			Normalized mutual information (%)		
	SGNMF	GNMF	NMF	SGNMF	GNMF	NMF
0	<b>98.27</b>	91.60	83.87	<b>93.09</b>	80.66	70.58
10	<b>98.47</b>	77.93	69.47	<b>93.97</b>	55.70	48.48
20	<b>98.27</b>	71.93	68.67	<b>93.09</b>	42.23	38.86
30	<b>98.33</b>	61.93	62.87	<b>93.39</b>	28.90	28.82
40	<b>98.27</b>	56.93	55.80	<b>93.15</b>	18.89	17.59
50	<b>98.33</b>	53.40	53.27	<b>93.39</b>	13.35	12.96

TABLE 8: Clustering performance on IRIS in case 2.

Cardinality (%)	Accuracy (%)			Normalized mutual information (%)		
	SGNMF	GNMF	NMF	SGNMF	GNMF	NMF
0	<b>98.60</b>	34.00	34.00	<b>94.56</b>	1.035	1.035
10	<b>98.13</b>	34.00	34.00	<b>93.39</b>	1.035	1.035
20	<b>98.20</b>	34.00	34.00	<b>93.02</b>	1.035	1.035
30	<b>98.00</b>	34.00	34.00	<b>92.27</b>	1.035	1.035
40	<b>98.13</b>	34.00	34.00	<b>92.92</b>	1.035	1.035
50	<b>98.00</b>	34.00	34.00	<b>92.33</b>	1.035	1.035

TABLE 9: Clustering performance on COIL20 in case 1.

Cardinality (%)	Accuracy (%)			Normalized mutual information (%)		
	SGNMF	GNMF	NMF	SGNMF	GNMF	NMF
0	68.92	<b>72.92</b>	64.54	77.52	<b>84.88</b>	74.18
10	<b>68.24</b>	63.37	57.63	<b>77.05</b>	71.78	65.49
20	<b>68.15</b>	53.40	48.39	<b>77.31</b>	61.90	56.26
30	<b>68.38</b>	45.81	43.91	<b>77.39</b>	54.34	51.60
40	<b>68.06</b>	41.03	38.28	<b>77.31</b>	49.45	46.26
50	<b>68.67</b>	35.25	32.64	<b>77.61</b>	42.92	41.42

TABLE 10: Clustering performance on COIL20 in case 2.

Cardinality (%)	Accuracy (%)			Normalized mutual information (%)		
	SGNMF	GNMF	NMF	SGNMF	GNMF	NMF
0	<b>68.22</b>	5.07	5.07	<b>77.05</b>	1.38	1.035
10	<b>68.16</b>	5.07	5.07	<b>77.36</b>	1.38	1.035
20	<b>68.78</b>	5.07	5.07	<b>77.11</b>	1.38	1.035
30	<b>68.12</b>	5.07	5.07	<b>77.08</b>	1.38	1.035
40	<b>68.38</b>	5.07	5.07	<b>77.33</b>	1.38	1.035
50	<b>68.25</b>	5.07	5.07	<b>77.67</b>	1.38	1.035

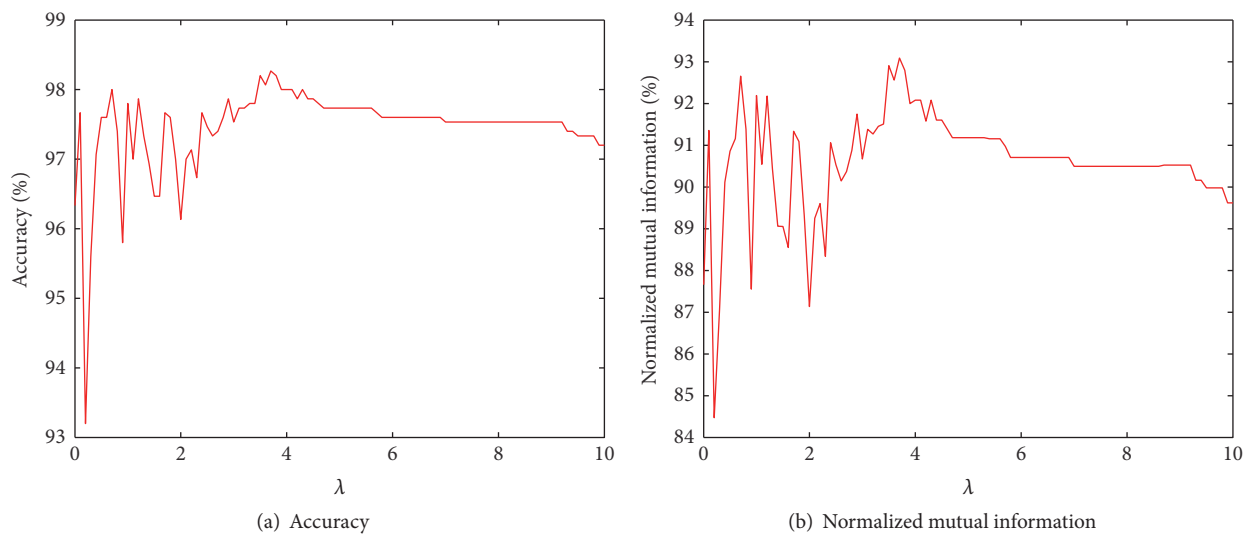


FIGURE 2: The performance of GSNMF versus different  $\lambda$  on IRIS data set.

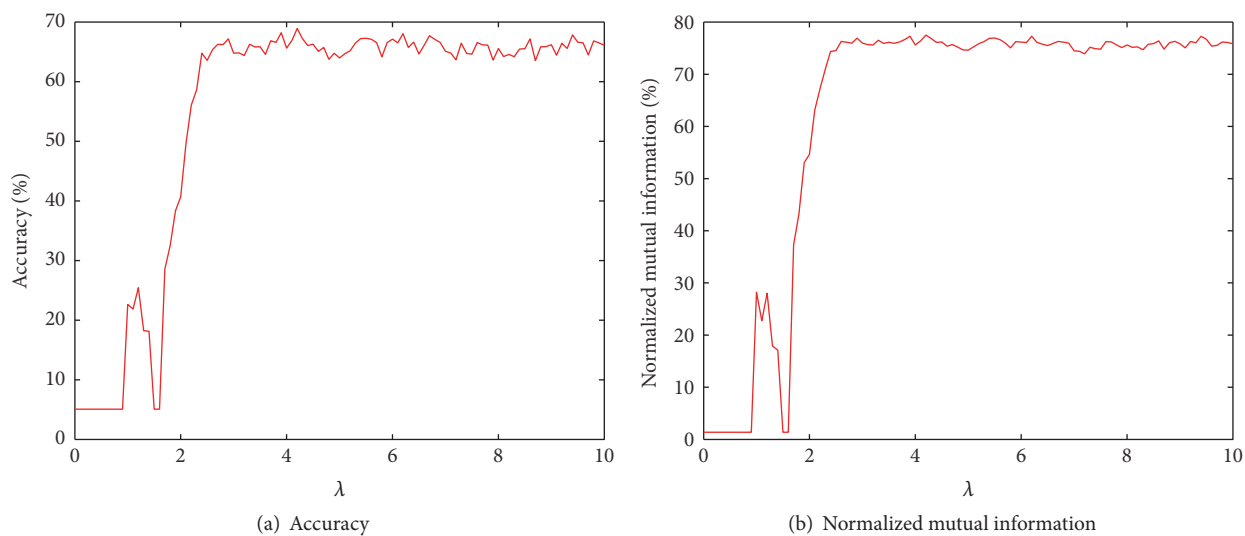


FIGURE 3: The performance of GSNMF versus different  $\lambda$  on COIL20 data set.

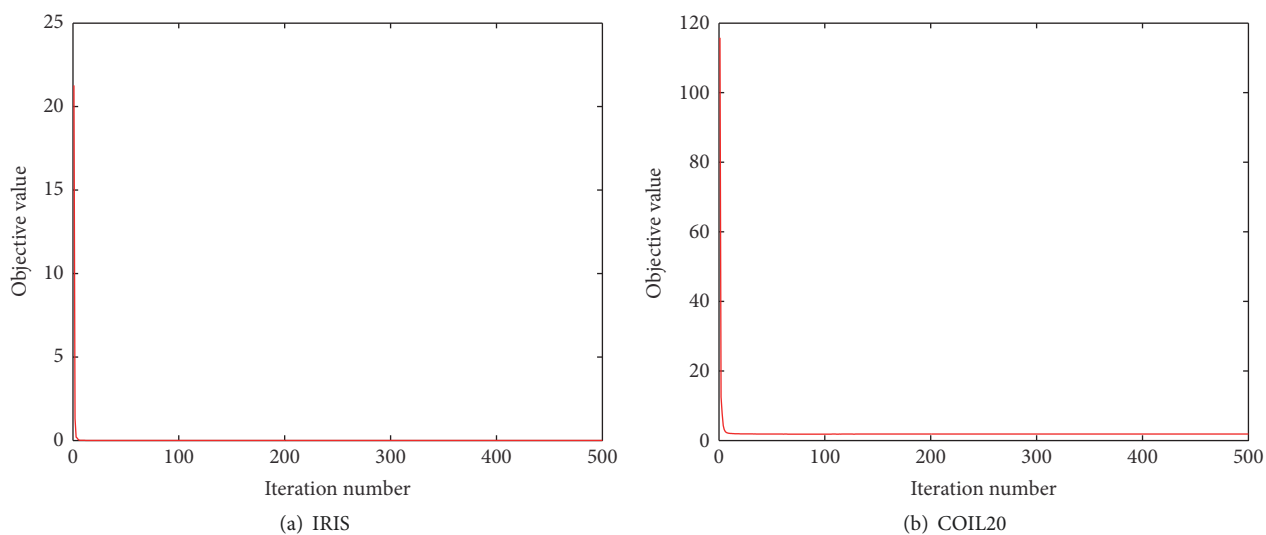


FIGURE 4: Iteration number versus objective value (100 : 1 scale) for GSNMF when  $\lambda = 5$  on IRIS and COIL20 data sets.

inertial terms  $\lambda$ . Secondly, the clustering performance cannot be affected by the negative initial values. However, GNMF and NMF have poor performance in clustering with negative initial values. Thirdly, if the initial values are sparse, our proposed method performs better than GNMF and NMF in the clustering.

Several topics remain to be discussed in our future work:

- (i) There is a parameter  $\lambda$  which searches the global optimal solution of GSNMF. Thus, a suitable value of  $\lambda$  is critical to our algorithm. It remains unclear how to select  $\lambda$  theoretically.
- (ii)  $h$  is a step length to decide the convergence rate in Algorithm 2. If it is assigned a small value, slow convergence makes a bad clustering performance. Thus, an adaptive step length should be considered.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References

- [1] S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1475–1490, 2004.
- [2] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the Association for Information Science and Technology*, vol. 41, no. 6, pp. 391–407, 1990.
- [3] 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.
- [4] D. Kalman, "A Singularly Valuable Decomposition: The SVD of a Matrix," *The College Mathematics Journal*, vol. 27, no. 1, p. 2, 1996.
- [5] I. T. Jolliffe, *Principal Component Analysis*, Springer Series in Statistics, Springer, New York, NY, USA, 2nd edition, 2002.
- [6] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Springer, Boston, MA, USA, 1992.
- [7] A. Cichocki, R. Zdunek, and S.-I. Amari, "New algorithms for non-negative matrix factorization in applications to blind source separation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06)*, vol. 5, pp. V621–V624, Toulouse, France, May 2006.
- [8] V. P. Pauca, F. Shahnaz, M. W. Berry, and R. J. Plemmons, "Text mining using non-negative matrix factorizations," in *Proceedings of the 4th SIAM International Conference on Data Mining*, pp. 452–456, 2004.
- [9] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval (SIGIR '03)*, pp. 267–273, Toronto, Canada, August 2003.
- [10] P. O. Hoyer, "Non-negative sparse coding," in *Proceedings of the 12th IEEE Workshop on Neural Networks for Signal Processing*, pp. 557–565, 2002.
- [11] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.
- [12] W.-S. Chen, Y. Zhao, B. Pan, and B. Chen, "Supervised kernel nonnegative matrix factorization for face recognition," *Neurocomputing*, vol. 205, pp. 165–181, 2016.
- [13] H. Liu, Z. Wu, X. Li, D. Cai, and T. S. Huang, "Constrained nonnegative matrix factorization for image representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1299–1311, 2012.
- [14] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biological Cybernetics*, vol. 52, no. 3, pp. 141–152, 1985.
- [15] Y. Xia, H. Leung, and J. Wang, "A projection neural network and its application to constrained optimization problems," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 4, pp. 447–458, 2002.
- [16] X. He, T. Huang, J. Yu, and C. Li, "An inertial projection neural network for solving variational inequalities," *IEEE Transactions on Cybernetics*, vol. 99, pp. 1–6, 2016.
- [17] H. Che, C. Li, X. He, and T. Huang, "An intelligent method of swarm neural networks for equalities-constrained nonconvex optimization," *Neurocomputing*, vol. 167, pp. 569–577, 2015.
- [18] J. Wang, "Analysis and design of a recurrent neural network for linear programming," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 40, no. 9, pp. 613–618, 1993.
- [19] X. Hu and B. Zhang, "An alternative recurrent neural network for solving variational inequalities and related optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 6, pp. 1640–1645, 2009.
- [20] X. Hu and J. Wang, "A recurrent neural network for solving a class of general variational inequalities," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 3, pp. 528–539, 2007.
- [21] X. Hu and J. Wang, "Design of general projection neural networks for solving monotone linear variational inequalities and linear and quadratic optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 5, pp. 1414–1421, 2007.
- [22] Z. Yan, J. Wang, and G. Li, "A collective neurodynamic optimization approach to bound-constrained nonconvex optimization," *Neural Networks*, vol. 55, pp. 20–29, 2014.
- [23] Q. Liu and J. Wang, "A one-layer recurrent neural network with a discontinuous hard-limiting activation function for quadratic programming," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 19, no. 4, pp. 558–570, 2008.
- [24] J. Fan and J. Wang, "A collective neurodynamic optimization approach to nonnegative matrix factorization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2344–2356, 2017.
- [25] Z. Yan and J. Wang, "Nonlinear model predictive control based on collective neurodynamic optimization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 4, pp. 840–850, 2015.
- [26] J. Shi, X. Ren, G. Dai, J. Wang, and Z. Zhang, "A non-convex relaxation approach to sparse dictionary learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '11)*, pp. 1809–1816, June 2011.
- [27] G. Li, Z. Yan, and J. Wang, "A one-layer recurrent neural network for constrained nonconvex optimization," *Neural Networks*, vol. 61, pp. 10–21, 2015.

- [28] H. Lee, A. Battle, R. Raina, and A. Ng, "Efficient sparse coding algorithms," in *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 801–808, 2006.
- [29] D. Cai, H. Bao, and X. He, "Sparse concept coding for visual analysis," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011*, pp. 2905–2910, USA, June 2011.
- [30] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.
- [31] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of Optimization Theory and Applications*, vol. 109, no. 3, pp. 475–494, 2001.
- [32] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 1, no. 4, pp. 586–597, 2007.
- [33] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 12, pp. 1624–1637, 2005.

