Graph Nonnegative Matrix Factorization with Alternative Smoothed L_0 Regulizations

Keyi Chen[1], Hangjun Che[2*], Xinqi Li[3] and Man-Fai Leung[4]

[1]College of Electronic and Information Engineering, Southwest University, Chongqing, 400715, China.

[2*]Chongqing Key Laboratory of Nonlinear Circuits and Intelligent Information Processing, College of Electronic and Information Engineering, Southwest University, Chongqing, 400715, China.

[3]Department of Computer Science, University of Exeter, Exeter, United Kingdom.

[4]School of Computing and Information Science, Faculty of Science and Engineering, Anglia Ruskin University, Cambridge, United Kingdom.

*Corresponding author(s). E-mail(s): hjche123@swu.edu.cn;
Contributing authors: cky123@email.swu.edu.cn;
xinqi.li@my.cityu.edu.hk; man-fai.leung@aru.ac.uk;

Abstract

Graph nonnegative matrix factorization (GNMF) can discover the data's intrinsic low-dimensional structure embedded in the high dimensional space. So it has superior performance for data representation and clustering. Unfortunately it's sensitive to noise and outliers. In this paper, to improve the robustness of GNMF, $\boldsymbol{l_0}$ norm is introduced to enhance the sparsity of factorized matrices. As the discontinuity of $\boldsymbol{l_0}$ norm and minimizing it is a NP-hard problem, five functions approximating $\boldsymbol{l_0}$ norm are used to transform the problem of the sparse graph nonnegative matrix factorization (SGNMF) to a global optimization problem. Finally the multiplicative updating rules (MUR) is designed to solve the problem and the convergence of algorithm is proven. In the experiment, the accuracy and normalized mutual information of clustering results show the superior performance of SGNMF on five public datasets.

Keywords: Sparse graph nonnegative matrix, alternative approximation function, multiplicative updating rules

1 Introduction

Motivated by the evidence in Psychology [1] and Physiology [2], nonnegative matrix factorization(NMF) is firstly proposed in Nature [3] to learn the part of object. It can be expressed as follows:

$$\boldsymbol{X} \approx \boldsymbol{A}\boldsymbol{B}^T$$

$\boldsymbol{X} \in R_+^{m\times n}$ is the data matrix, $\boldsymbol{A} \in R_+^{m\times c}$ is the basis matrix, $\boldsymbol{B} \in R_+^{n\times c}$ is the coefficient matrix. NMF gets feature whose basis are the column vector of $\boldsymbol{A}$, so row vector of $\boldsymbol{B}$ can replace the original data and be seen as the extracted feature. c is the number of components defined according to demands. For clustering, it can be set to the number of clusters. For data reconstruction, the bigger c is, the better the data matrix $\boldsymbol{X}$ is reconstructed. It's one of the most popular algorithm for data processing, which is widely used in hyperspectral unmixing [4], text mining[5], medical data [6], gene selection and tumor classification [7]. Many algorithms are proposed based on NMF, such as sparse NMF [8], spectral-spatial joint sparse NMF [9] and NMFAN [10]. Besides, as a non-convex optimization problem, a variety of methods are proposed to find the global optimal solution [11–15].

Graph nonnegative matrix factorization(GNMF) [16] is one of variants. Based on the local invariance in the manifold learning, it discovers deeper structure of the data and extracts more representative features. The superior performance of GNMF makes it very popular on hyperspectral unmixing [17], gene clustering [18], computer vision[19] and disease detection [20].

However, there can be some noise existing in data or introduced by mapping data from the high-dimensional space to the low-dimensional space, and GNMF is sensitive to these noise and outliers [21]. To avoid these problems, sparsity of the factorized matrices is required to enhance the robustness of the performance of GNMF. In [21, 22], l_2 and l_1 norm are introduced to alleviates the impact of noise and outliers. In [18], $l_{2,1}$ norm is used to measure the error of matrix factorization to enhance robustness.

Sparsity is defined by l_0 norm which aims to compute the number of nonzero elements in a vector. But l_0 norm is discontinuous and minimizing it is a NP-hard problem. l_1 norm is the convex relaxation of l_0 norm and some smooth functions are proposed to approximate it to make the problem enable to solve. In [23], the inverted Gaussian function is proposed to approximate l_0 norm which is used in [11] to get a sparser solution. In [24], a neurodynamic approach is proposed to optimize l_0 norm constrained problem where the inverted Gaussian function is used as an approximation of l_0 norm. In [25], a Hyperbolic Tangent functions which is closer to l_0 norm is proposed to get a sparse solution of NMF.

Motivated by the above work and idea, in this paper, sparsity-constrained graph nonnegative matrix factorization (SGNMF) is proposed to enhance the robustness and eliminate noise. A fraction with absolute value of variable is

proposed to approximate the l_0 norm. Additionally, other four previously proposed functions are introduced to compare the performance. Besides, a general algorithm is proposed for GNMF with approximate function-based regulization and the convergence of the algorithm is proven.
The structure of rest paper is as follows: in the second section, preliminaries are provided including NMF, GNMF. In the third section, some functions used to approximate l_0 norm and the algorithm are given. Optimization algorithm and the proof of convergence are given in the fourth section. Then the experiment results are shown in the fifth section and conclusions are in the last.

2 Preliminaries

In this section, basic formulations and definitions of NMF and GNMF are given.

2.1 Nonnegative Matrix Factorization

Denote $\boldsymbol{X} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}] \in R_+^{m \times n}$ as the data matrix, $\boldsymbol{x_j} \in R_+^m$ is the j-th sample, it's the j-th column vector of $\boldsymbol{X}$. NMF can be expressed as follows:

$$\begin{aligned} &\min \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{B^T}\|_F \\ &\text{s.t. } \boldsymbol{A} \geqslant 0, \boldsymbol{B} \geqslant 0. \end{aligned} \tag{1}$$

The Frobenius norm is used to measure the loss of two matrices in (1). $\boldsymbol{A} \in R_+^{m \times c}$ is the basis matrix, $\boldsymbol{B} \in R_+^{n \times c}$ is the coefficient matrix, c is the number of basis. Denote $\boldsymbol{b}_j \in R_+^{1 \times c}$ as the i-th row of $\boldsymbol{B}$.
The updating rules are given in (2) to solve the problem, the a_{ik} is number in the i-th row and k-th column of the $\boldsymbol{A}$, b_{jk} has similar definition for $\boldsymbol{B}$.

$$\begin{aligned} a_{ik} &= a_{ik} \cdot \frac{(\boldsymbol{X}\boldsymbol{B})_{ik}}{(\boldsymbol{A}\boldsymbol{B}^T B)_{ik}} \\ b_{jk} &= b_{jk} \cdot \frac{(\boldsymbol{X}^T\boldsymbol{A})_{jk}}{(\boldsymbol{B}\boldsymbol{A}^T\boldsymbol{A})_{jk}} \end{aligned} \tag{2}$$

In [26], it has proven that the objective function value is non-increasing under the updating rules (2). Besides, in [27], an optimization method based on discrete-time projection neural network is proposed to find a solution closer to global optimal solution.

2.2 Graph Regularized Nonnegative Matrix Factorization

Based on the local invariance in the manifold learning, Cai et al propose the GNMF [16]. Compare to NMF, GNMF exploits the intrinsic geometry of the data distribution and learns a more sparse matrix.
To observe the local structure of the original data, a k-nearest graph or a full

connected graph $\boldsymbol{W}$ is required, which is called adjacency matrix. The weight between two vertexes can be defined in the following ways:

- **0-1 weight:** if two vertexes $\boldsymbol{x}_l$ and $\boldsymbol{x}_j$ are connected, the weight w_{jl} between them is 1, otherwise it's 0. However, it can't be used in the full connected graph.
$$w_{jl} = \begin{cases} 1 \ (\boldsymbol{x}_l \text{ and } \boldsymbol{x}_j \text{ are connected}) \\ 0 \ (\text{others}) \end{cases} \tag{3}$$
- **Gaussian kernel weight:** if $\boldsymbol{x}_l$ and $\boldsymbol{x}_j$ are connected, the w_{jl} is defined as follows:
$$w_{jl} = \begin{cases} e^{-\frac{\|\boldsymbol{x}_l - \boldsymbol{x}_j\|^2}{2\sigma^2}} \ (\boldsymbol{x}_i \text{ and } \boldsymbol{x}_l \text{ are connected}) \\ 0 \ (\text{others}) \end{cases}. \tag{4}$$
It reflects the nonlinear local structure, but there is a parameter σ.

The objective function of the local invariance is described as follows:

$$\sum_{j,l=1}^{n} \|\boldsymbol{b}_j - \boldsymbol{b}_l\|^2 w_{jl}. \tag{5}$$

$\boldsymbol{D}$ is the degree matrix, it can be computed as follows:

$$d_{jl} = \begin{cases} \sum_{l=0}^{n} w_{jl} \ (l = j) \\ 0 \ (\text{others}) \end{cases}.$$

Besides Laplacian matrix $\boldsymbol{L}$ is defined as follows:

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}. \tag{6}$$

So the (5) is simplified as follows:

$$\begin{aligned} &\sum_{j,l=1}^{n} \|\boldsymbol{b}_j - \boldsymbol{b}_l\|^2 \boldsymbol{W}_{jl} \\ =&\sum_{j=1}^{n} \boldsymbol{b}_j^T \boldsymbol{b}_j \boldsymbol{D}_{jj} - \sum_{j,l=1}^{n} \boldsymbol{b}_j^T \boldsymbol{b}_l \boldsymbol{W}_{jl} \\ =&\operatorname{Tr}(\boldsymbol{B}^T \boldsymbol{D} \boldsymbol{B}) - \operatorname{Tr}(\boldsymbol{B}^T \boldsymbol{W} \boldsymbol{B}) = \operatorname{Tr}(\boldsymbol{B}^T \boldsymbol{L} \boldsymbol{B}). \end{aligned} \tag{7}$$

Add (7) as a regularizer and get the objective function of GNMF:

$$\|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{B}^{\boldsymbol{T}}\| + \operatorname{Tr}(\boldsymbol{B}^T \boldsymbol{L} \boldsymbol{B}). \tag{8}$$

The problem is formulated as follows:

$$\begin{aligned} &\min \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{B}^{\boldsymbol{T}}\|_F + \lambda \operatorname{Tr}(\boldsymbol{B}^T \boldsymbol{L} \boldsymbol{B}) \\ &\text{s.t. } \boldsymbol{A} \geqslant 0, \boldsymbol{B} \geqslant 0. \end{aligned} \tag{9}$$

(9) is convex for $\boldsymbol{A}$ or $\boldsymbol{B}$. But it's not convex for both of them. Like NMF, the updating rules are given in (10) .

$$\begin{aligned} a_{ik} &\leftarrow a_{ik}\frac{(\boldsymbol{XB})_{ik}}{(\boldsymbol{AB}^T\boldsymbol{B})_{ik}} \\ b_{jk} &\leftarrow b_{jk}\frac{\left(\boldsymbol{X}^T\boldsymbol{A}+\lambda\boldsymbol{WB}\right)_{jk}}{\left(\boldsymbol{BA}^T\boldsymbol{A}+\lambda\boldsymbol{DB}\right)_{jk}} \end{aligned} \tag{10}$$

In [16], it's proven that the objective function (8) is non-increasing under the updating rules.

3 Sparsity-constrained Graph Nonnegative Matrix Factorization

3.1 Alternative smoothed l_0 approximate functions

l_0 norm computes the number of nonzero elements of a vector which is used to enforce the required sparsity. Sparsity is always required to eliminate useless information and enhance interpretability.
l_0 norm is described as follows:

$$\|(\boldsymbol{x})\|_0 = \sum_{i=1}^{n} 1 - \sigma(x_{ij}). \tag{11}$$

$\sigma(x)$ is the unit impulse function, it's discontinuous and minimizing it is a NP-hard problem. To solve it, each situation needs to be enumerated.
However, there are several functions used to approximate l_0 norm, they all have following form:

$$\lim_{\sigma\to\infty} f(x,\sigma) = \begin{cases} 1 & (x=0) \\ 0 & (\text{others}) \end{cases}. \tag{12}$$

σ is used to controls the degree of approaching l_0 norm. In this paper, four functions previously proposed are used to approximate l_0 norm to measure the sparsity of the matrices factorized by GNMF:

- Inv. Gaussian [23]: $f_1(x) = 1 - e^{-\frac{x^2}{\sigma^2}}$.
- Inv. Laplacian [28]: $f_2(x) = 1 - e^{-\frac{|x|}{|\sigma|}}$.
- Comp. inv. func [29]: $f_3(x) = \frac{x^2}{x^2+\sigma^2}$.
- Symmetric. CT [30]: $f_4(x) = \sin\left(\arctan\left(\frac{x^2}{\sigma^2}\right)\right)$.

For functions f_1, f_3, f_4, they have following properties:

$$\lim_{x\to 0} \bigtriangledown f(x,\sigma) = 0. \tag{13}$$

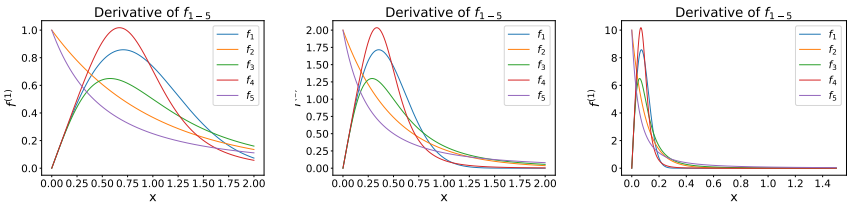
As the update rules of NMF and GNMF are equivalent to the gradient descent method [16], they may just lead values to be very small but not zero, so the sparsity is not gotten.
Before giving our algorithm, we give a function f_5, it's easy to calculate and concave for R_+:

$$f_5(x) = \frac{\mid x \mid}{\mid x \mid + \sigma}. \tag{14}$$

The second derivative of f_5 is less than 0:

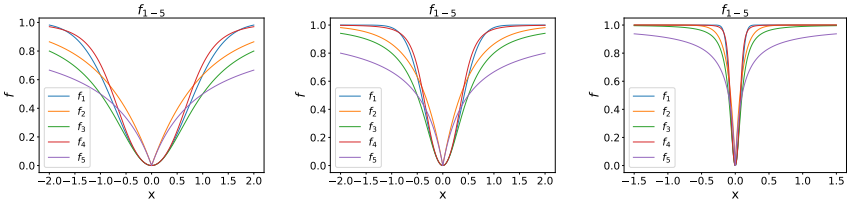
$$f_5^{(2)} = -\frac{2\sigma}{(\sigma + x)^3} < 0. \tag{15}$$

It's obvious that f_5 satisfies (12). In fig: 1, it's shown that the derivation



(a) $\sigma = 1$ (b) $\sigma = 0.5$ (c) $\sigma = 0.1$

Fig. 1 The derivative of five alternative approximation functions.



(a) $\sigma = 1$ (b) $\sigma = 0.5$ (c) $\sigma = 0.1$

Fig. 2 The comparisons of five function with different σ.

becomes much larger when x approaches zero. This makes f_5 more efficient to lead sparsity. In fig: 2, it's shown that the smaller σ is, f_{1-5} is closer to l_0 norm.

3.2 Sparsity-constrained GNMF

The sparsity of the features extracted by GNMF can be expressed by l_0 norm as following equation:

$$\sum_{j=0}^{n} \|(\boldsymbol{b}_j)\|_0. \tag{16}$$

And the optimization problem is described as follows:

$$\min_{\boldsymbol{B}\geqslant 0} \sum_{j=0}^{n} \|(\boldsymbol{b}_j)\|_0. \tag{17}$$

As l_0 norm is discontinuous and the problem is NP-hard, so we use $f(x, \sigma)$ to replace the l_0 norm to make the problem enable to solve and obtain the followings problem:

$$\min_{\boldsymbol{B}\geqslant 0} \sum_{j=0}^{n} \sum_{k=0}^{c} f(b_{jk}, \sigma). \tag{18}$$

Add it as a regulization, the sparsity-constrained GNMF (SGNMF) is

$$\min_{\boldsymbol{A},\boldsymbol{B}\geqslant 0} \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{B}^T\|_F + \lambda \operatorname{Tr}\left(\boldsymbol{B}^T\boldsymbol{L}\boldsymbol{B}\right) + \beta \sum_{j=0}^{n} \sum_{k=0}^{c} f(b_{jk}, \sigma). \tag{19}$$

By solving problem (19), the extracted features can get deeper structure of the original data by the regularizer of GNMF, some noise can be eliminated and it's guaranteed that NMF learns part of the object. Besides interpretability of the data led by sparsity is obtained.

4 Optimization Algorithm

4.1 Updating Rules

The problem of NMF and GNMF is non-convex, the optimal solution is not guaranteed. So a solution can find local optima of SGNMF.

Denote $\boldsymbol{\Omega} \in R_+^{m\times c}, \boldsymbol{\Theta} \in R_+^{n\times c}$ respectively as the Lagrangian multipliers for the constraint $a_{ik} \geqslant 0, b_{jk} \geqslant 0$, ω_{ik} is the element in the i-th row and k-th column of $\boldsymbol{\Omega}$. θ_{jk} is the element in j-th row and k-th column of $\boldsymbol{\Theta}$. The objective function can be transformed as follows:

$$\begin{aligned} &\operatorname{Tr}\left(\left(\boldsymbol{X} - \boldsymbol{A}\boldsymbol{B}^T\right)\left(\boldsymbol{X} - \boldsymbol{A}\boldsymbol{B}^T\right)^T\right) + \lambda \operatorname{Tr}\left(\boldsymbol{B}^T\boldsymbol{L}\boldsymbol{B}\right) \\ &+ \beta \sum_{j=0}^{n} \sum_{k=0}^{c} f(b_{jk}, \sigma) \\ =& \operatorname{Tr}\left(\boldsymbol{X}\boldsymbol{X}^T\right) - 2\operatorname{Tr}\left(\boldsymbol{X}\boldsymbol{B}\boldsymbol{A}^T\right) + \operatorname{Tr}\left(\boldsymbol{A}\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{A}^T\right) \\ &\lambda \operatorname{Tr}\left(\boldsymbol{B}^T\boldsymbol{L}\boldsymbol{B}\right) + \beta \sum_{j=0}^{n} \sum_{k=0}^{c} f(b_{jk}, \sigma). \end{aligned} \tag{20}$$

The Lagrangian function

$$\begin{aligned}\mathcal{L} = &\operatorname{Tr}\left(\boldsymbol{X}\boldsymbol{X}^T\right) - 2\operatorname{Tr}\left(\boldsymbol{X}\boldsymbol{B}\boldsymbol{A}^T\right) + \operatorname{Tr}\left(\boldsymbol{A}\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{A}^T\right)\\ &\lambda\operatorname{Tr}\left(\boldsymbol{B}^T\boldsymbol{L}\boldsymbol{B}\right) + \beta\sum_{j=0}^{n}\sum_{k=0}^{c} f(b_{jk},\sigma)\\ &+ \operatorname{Tr}\left(\Omega\boldsymbol{A}^T\right) + \operatorname{Tr}\left(\Theta\boldsymbol{B}^T\right).\end{aligned} \tag{21}$$

Denote $f^{(d)}$ as d-th derivative. The partial derivatives of $\mathcal{L}$ with respect to $\boldsymbol{A}$ and $\boldsymbol{B}$ are

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{A}} = -2\boldsymbol{X}\boldsymbol{B} + 2\boldsymbol{A}\boldsymbol{B}^T\boldsymbol{B} + \Omega, \tag{22}$$

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\boldsymbol{B}} = &-2\boldsymbol{X}^T\boldsymbol{A} + 2\boldsymbol{B}\boldsymbol{A}^T\boldsymbol{A} + 2\lambda\boldsymbol{L}\boldsymbol{B}\\ &+ \beta f^{(1)}(b_{jk,\sigma}) + \Theta.\end{aligned} \tag{23}$$

Using the KKT conditions $a_{ik}\omega_{ik} = 0$ and $\theta_{jk}b_{jk} = 0$, the following equations for a_{ik} and b_{jk} are obtained:

$$-\left(\boldsymbol{X}\boldsymbol{B}\right)_{ik}a_{ik} + \left(\boldsymbol{A}\boldsymbol{B}^T\boldsymbol{B}\right)_{ik}a_{ik} = 0, \tag{24}$$

$$\begin{aligned}&-\left(\boldsymbol{X}^T\boldsymbol{A}\right)_{jk}b_{jk} + \left(\boldsymbol{B}\boldsymbol{A}^T\boldsymbol{A}\right)_{jk}b_{jk}\\ &+ \lambda(\boldsymbol{L}\boldsymbol{B})_{jk}b_{jk} + \beta f^{(1)}(b_{jk},\sigma)b_{jk} = 0.\end{aligned} \tag{25}$$

Based on (24) and (25), the following updating rules are proposed

$$a_{ik} \leftarrow a_{ik}\frac{(\boldsymbol{X}\boldsymbol{B})_{ik}}{(\boldsymbol{A}\boldsymbol{B}^T\boldsymbol{B})_{ik}}. \tag{26}$$

$$b_{jk} \leftarrow b_{jk}\frac{\left(\boldsymbol{X}^T\boldsymbol{A} + \lambda\boldsymbol{W}\boldsymbol{B}\right)_{jk}}{\left(\boldsymbol{B}\boldsymbol{A}^T\boldsymbol{A} + \lambda\boldsymbol{D}\boldsymbol{B}\right)_{jk} + 0.5\beta f^{(1)}(b_{jk},\sigma)}. \tag{27}$$

It's obvious that the updating rules equal to that of NMF if the $\lambda = 0$ and $\beta = 0$. They equal to that of GNMF if $\beta = 0$. The whole algorithm of SGNMF is described in Algorithm 1.

4.2 Convergence Study

The proof of convergence about updating rules (26) is the same with NMF. So the convergence is needed to prove under updating rule (27). To prove the convergence, following definition and lemmas are given.

Definition 1. $\varphi(b, b')$ is the auxiliary function for a function $F(b)$ if

$$\varphi(b, b') \geqslant F(b), \quad \varphi(b', b') = F(b').$$

The property of the auxiliary function is shown in the following lemma.

Algorithm 1 Algorithm of SGNMF

Require: $\boldsymbol{X}$: data matrix($\boldsymbol{X} \in \mathcal{R}_{+}^{m \times n}$, each column vector is a sample); r: max iteration; k: number of components; e: max error tolerance; λ, β, σ_1: parameters in SGNMF; σ_2: parameter in f_{1-5}; p: number of neighborhoods.

Ensure: $\boldsymbol{A}$: basis matrix; $\boldsymbol{B}$: coefficient matrix.

if X is from images **then**
  for element x in $\boldsymbol{X}$ **do**
    normalize x
  end for
end if
$counter = 0$
initialize the basis matrix $\boldsymbol{A}(\boldsymbol{A} \in R_{+}^{m \times c})$ and the coefficient matrix $\boldsymbol{B}(\boldsymbol{B} \in R_{+}^{n \times c})$ with NNDSVD[31].
$\boldsymbol{W} \leftarrow$ Adjacency matrix
$\boldsymbol{D} \leftarrow$ Degree matrix
Laplacian matrix: $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$
$error \leftarrow$ objective function value of SGNMF
while $error > e$ and $counter < r$ **do**
  for element a in $\boldsymbol{A}$ **do**
    $Den_1 \leftarrow$ denominator of updating rule(26) for a
    if $Den_1 \neq 0$ **then**
      $a \leftarrow$ the new a(computed by updating rule(26))
    end if
  end for
  for element b in $\boldsymbol{B}$ **do**
    $Den_2 \leftarrow$ denominator of updating rule(27) for b
    if $Den_2 \neq 0$ **then**
      $b \leftarrow$ the new b(computed by updating rule(27))
    end if
  end for
  $error \leftarrow$ value of (19)
  $counter = counter + 1$
end while
return $\boldsymbol{A}, \boldsymbol{B}$

Lemma 1. If φ is an auxiliary function of the F. Under the following updating rule, F is non-increasing:

$$b^{t+1} = \arg\min_{b} \quad \varphi\left(b, b^{t}\right). \tag{28}$$

Proof: $F\left(b^{t+1}\right) \leq \varphi\left(b^{t+1}, b^{t}\right) \leq \varphi\left(b^{t}, b^{t}\right) = F\left(b^{t}\right)$.

Theorem 1.

If $\alpha(x)$ is concave, for objective function

$$\mathcal{F} = \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{B}^T\|_F + \lambda \operatorname{Tr}(\boldsymbol{B}\boldsymbol{L}\boldsymbol{B}^T) + \sum_{k=1,j=1}^{c,n} \beta\alpha(b_{jk}) \tag{29}$$

will be non-increasing under the following updating rules:

$$b_{jk}^{t+1} = b_{jk}^t \frac{(\boldsymbol{X}^T\boldsymbol{A} + \lambda\boldsymbol{W}\boldsymbol{B})_{jk}}{(\boldsymbol{B}\boldsymbol{A}^T A + \lambda\boldsymbol{D}\boldsymbol{B})_{jk} + 0.5\beta\alpha^{(1)}(b_{jk}^t)}. \tag{30}$$

Proof: Denote $\mathcal{F}$ as the objective function of SGNMF with respect to $\boldsymbol{B}$,

$$\mathcal{F}_{jk}^{(1)} = \left(-2\boldsymbol{X}^T\boldsymbol{A} + 2\boldsymbol{B}\boldsymbol{A}^T\boldsymbol{A} + 2\lambda\boldsymbol{L}\boldsymbol{B}\right)_{jk} + \beta\alpha^{(1)}(b_{jk}), \tag{31}$$

$$\mathcal{F}_{jk}^{(2)} = 2\left(\boldsymbol{A}^T\boldsymbol{A}\right)_{kk} + 2\lambda\boldsymbol{L}_{jj} + \beta\alpha^{(2)}(b_{jk}). \tag{32}$$

Denote φ as the auxiliary function of the $\mathcal{F}_{jk}$ as follows:

$$\begin{aligned}\varphi(b, b^t) = {} & \mathcal{F}_{jk}(b^t) + \mathcal{F}_{jk}^{(1)}(b)(b - b^t) \\ & + \frac{\left(\mathbf{B}\mathbf{A}^T\mathbf{A}\right)_{jk} + \lambda(\mathbf{D}\mathbf{B})_{jk} + 0.5\beta\alpha^{(1)}(b_{jk}^t)}{b_{jk}^t}\left(b - b_{jk}^t\right)^2.\end{aligned} \tag{33}$$

By finding the extreme point of (33), it's easy to derive updating rules (27) and if the φ is the auxiliary function of $\mathcal{F}$, the convergence is proven with Lemma 1. As $\alpha(x)$ is concave, the first-order Taylor approximation of $\alpha(b)$ is bigger than or equals to $\alpha(b)$.

$$\alpha(b_{jk}^t) + \alpha^{(1)}(b_{jk}^t)(b - b_{jk}^t) \geqslant \alpha(b_{jk})$$

Then get the following:

$$\sum_{i=2}^{\infty} \frac{\alpha^{(i)}(b_{jk}^t)}{i!}(b - b_{jk}^t)^i \leqslant 0\ . \tag{34}$$

Add (29) to both sides of the (34). Get the following function φ_1 is bigger than $\mathcal{F}$:

$$\begin{aligned}\varphi_1(b, b_{jk}^t) = {} & \mathcal{F}_{jk}(b^t) + \mathcal{F}_{jk}^{(1)}(b_{jk}^t)(h - h_{jk}^t) \\ & + \left(\left(\mathbf{A}^T\mathbf{A}\right)_{kk} + \lambda\mathbf{L}_{jj}\right)(b - b_{jk}^t)^2.\end{aligned} \tag{35}$$

Then to prove that $\varphi_(b, b_{jk}^t)$ is bigger than $\varphi_1(b, b_{jk}^t)$, the following is given:

$$\left(\mathbf{B}\mathbf{A}^T\mathbf{A}\right)_{jk} = \sum_{l=1}^{c} h_{jl}^t \left(\mathbf{A}^T\mathbf{A}\right)_{lk} \geq b_{jk}^t \left(\mathbf{A}^T\mathbf{A}\right)_{kk}, \tag{36}$$

$$\begin{aligned}\lambda(\mathbf{DB})_{jk} &= \lambda\sum_{l=1}^{m}\mathbf{D}_{jl}b_{lk}^{t} \geq \lambda\mathbf{D}_{jj}b_{jk}^{t} \\ &\geq \lambda(\mathbf{D}-\mathbf{W})_{jj}b_{jk}^{t} = \lambda\mathbf{L}_{jj}b_{jk}^{t}.\end{aligned} \tag{37}$$

With the inequality above, it's easy to check that the third term of $\varphi_2(h, h_{ab}^{(t)})$ is bigger than the third term of $\varphi_1(h, h_{ab}^{(t)})$, so $\varphi(h, h_{ab}^{(t)}) \geqslant \varphi_1(h, h_{ab}^{(t)})$ holds. Besides,

$$\varphi(h_{ab}^{t}, h_{ab}^{t}) = F_{ab}(h_{ab}^{t}).$$

So $\varphi_(h, h_{ab}^{t})$ is the auxiliary function of $\mathcal{F}$, the proof completes. The proof mainly follows the idea in [16].

Lemma 2.: the update rules (27) is equivalent to gradient descent method.

Proof: : the gradient descent method has following format:

$$b_{jk} \leftarrow b_{jk} - \eta_{jk}\frac{\partial\mathcal{F}}{\partial b_{jk}}. \tag{38}$$

η is the learning rate. In Deng Cai's[16] paper, there is a technique to set η, Follow the idea, η can be set to

$$\frac{b_{jk}}{2(\boldsymbol{BG}^{T}\boldsymbol{G}+\lambda\boldsymbol{DB})_{jk}+\beta\alpha^{(1)}(b_{jk})}.$$

Then get the following:

$$\begin{aligned}&b_{jk} - \eta_{jk}\frac{\partial\mathcal{F}}{\partial b_{jk}} = \\ &b_{jk} - \frac{b_{jk}}{(2\boldsymbol{BA}^{T}\boldsymbol{A}+2\lambda\boldsymbol{DB})_{jk}} + \beta\alpha^{(1)}(b_{jk})\frac{\partial\mathcal{F}}{\partial b_{jk}} \\ =&b_{jk} - \frac{b_{jk}}{2\left(\boldsymbol{BA}^{T}\boldsymbol{A}+\lambda\boldsymbol{DB}\right)_{jk}+\beta\alpha^{(1)}(b_{jk})} \\ &\left(\left(-2\boldsymbol{X}^{T}\boldsymbol{A}+2\boldsymbol{BA}^{T}\boldsymbol{A}+2\lambda\boldsymbol{LB}\right)_{jk}+\beta\alpha^{(1)}(b_{jk})\right) \\ =&b_{jk}\frac{\left(\boldsymbol{X}^{T}\boldsymbol{A}+\lambda\boldsymbol{WB}\right)_{jk}}{\left(\boldsymbol{BA}^{T}\boldsymbol{A}+\lambda\boldsymbol{DB}\right)_{jk}+0.5\beta\alpha^{(1)}(b_{jk})}.\end{aligned} \tag{39}$$

It's obvious that (39) is the same with (27). The proof completes.

Lemma 2 indicates that the objective function is updated along with negative gradient direction in the feasible region.

5 Experimental Results

In this section, SGNMF is used to cluster on five public dataset. The clustering results, the convergence performance and the effect of the parameters are discussed.

5.1 Description of Dataset

The information of the datasets are listed as follows and the summary of the datasets is in TABLE 1:

Table 1 Dataset description.

Dataset	Feature Number	Sample Number	Clustering Number
YALE	77760	165	15
USPS	256	400	10
UMIST	2576	564	20
LIBRAS	90	360	15
JAFFE	65536	213	10

- **YALE** [1]: It contains 165 grayscale image of 15 persons, each person has different facial expression or configuration.
- **USPS** [2]: A dataset consists of 9298 images which are 16*16 grayscale pixels. In our experiment 400 images are used to cluster to show the performance of algorithm.
- **UMIST**: [3] It consists of 564 images of 20 persons, each person is in shown in a range of poses from profile to frontal views. Each image is resized in 46*56 pixels.
- **LIBRAS**: It's a dataset from UCI dataset [32], it consists of 15 classes of 24 instances, each class references to a hand movement type in LIBRAS.
- **JAFFE**: [4] It consists of 213 images from 10 Japanese female expressers with different facial expressions which are all 256*256 pixels.

5.2 Compared Algorithms

To compare the performance of the proposed approach, Accuracy(ACC) and normalized mutual information(NMI) are used, the detailed information of the comparison algorithms is listed as follows:

- K-means: it's the most classic clustering algorithm. Through several iterations, it can cluster very quickly and efficiently.
- K-means++: It is an algorithm based on K-means. But it avoids the uncertain performance caused by random initialization of K-means.
- Spectral clustering(SP-clustering): It is a clustering method based on graph theory which transforms the partition of data into the segmentation of graph. In the experiment, K-means++ is used to get the label of each sample. Ncut is used to cut the graph. And K-neighborhoods is used to construct the graph. The weight on the edge is computed by Gaussian kernel function.

[1] http://vision.ucsd.edu/content/yale-face-database
[2] https://www.kaggle.com/bistaumanga/usps-dataset
[3] https://www.visioneng.org.uk/datasets/
[4] https://zenodo.org/record/3451524#.YZVNGsWHqUk

- NMF-based clustering: NMF is used to extract the features of each sample. Then, K-means++ is used to get the tag of sample. When extracting the feature, the number of columns of the basis matrix in NMF is set to the number of items, and the number of iterations of the algorithm is 200. SVD initialization and normalization of basis matrix is used to increase stability and speed up convergence.
- KKM(kernel K-means): the algorithm is almost the same with K-means, but it pays more attention on kernel space other than Euclidean space.
- RKKM(robust kernel K-means): by introducing sparsity induced norm, the effects of outliers which is sensitive for K-means can decrease and a more stable result is obtained.
- AASC(affinity aggregation for spectral clustering): an algorithm is proposed in [33], it extends spectral clustering to a setting with multiple affinities available.
- RMKKM(robust multi-kernel K-means): based on RKKM it introduces multiple kernel functions to explore a better Hilbert space which is proposed in [34].
- CFSFDP(clustering by fast search and find of density peaks): it's a clustering algorithm based on density of data samples, which is proposed in [35].

For KKM, RKKM, RMKKM and AASC, they can be executed by following codes [5]. The rest algorithms can be executed in our repository[6].

5.3 Basis normalize and NNDSVD initialization

The basis of data matrix $\boldsymbol{X}$' space are basic unit vector group. SGNMF learns a space whose basis are the column vector of the matrix $\boldsymbol{A}$. To remain length of the basis the same, we can achieve it by follows:

$$a_{ik} = \frac{a_{ik}}{\sqrt{\sum_{i=0}^{m} a_{ik}^2}} \tag{40}$$

$$b_{jk} = b_{jk} \times \sqrt{\sum_{i=0}^{m} a_{ik}^2} \tag{41}$$

Besides, NNDSVD [31] is used to accelerate convergence and enhance stability in the experiment.

5.4 Clustering Results

In the comparison experiments, the component number is set to the number of clusters for each dataset. Each algorithm clusters on a dataset for 20 times, the mean and standard error of the Accuracy and NMI are given. For image datasets, the data is normalized before clustering. The clustering results are

[5] https://github.com/csliangdu/RMKKM
[6] https://github.com/chen12304/SGNMF

recorded in Tables 2, 3 where top three NMI and Accuracy's means are bold. The results are summarized as follows:

- SGNMF has a better performance than others. For the proposed algorithms with five smooth functions, at least one are ranked in the top three each time. For ACC, SGNMF gets top three all on JAFFE and UMIST, especially on JAFFE where has a 7%-8% higher ACC than others. For NMI, SGNMF gets top three all on UMIST and LIBRAS, especially on UMIST where almost are 15%-17% higher than others.
- SGNMF has a more stable result. these five functions used in SGNMF have at least three which have a standard error less than 1, for comparison algorithms, the standard errors are almost higher than 1.
- From fig. 3, the objective function values decrease with the increase of the iterations.

5.5 Sparsity Discussion

Sparsity can eliminate noise, increase the interpretability of data and ensure NMF to learn part of object. Sparsity performance is discussed as follows.
To observe the performance on sparsity, we plot the $\boldsymbol{B}$ as a grayscale image factorized by NMF, GNMF and SGNMF. To better show the sparsity, 20×20 grayscale patches of original pictures are shown in figs 4-8. The original pictures can be found through the link of footnote[7]. Furthermore, the sparsity of the matrices which are plotted is given in Tables 4. Sparsity is calculated by followings:

$$Sparsity(\boldsymbol{X}) = \frac{SF(\boldsymbol{X})}{mn} \tag{42}$$

$SF(\boldsymbol{X})$ is the sparseness factor of $\boldsymbol{X}$ which is used in [36]. It's described as follows:

$$SF(\boldsymbol{X}) = \sum_{i=1}^{m}\sum_{j=1}^{n} m(x_{ij}),$$
$$m(x) = \begin{cases} 1 & (x < \tau) \\ 0 & (x \geqslant \tau) \end{cases}.$$

τ is the threshold value to decide whether an element can be regard as zero.

For $\boldsymbol{B}$ in the figs: 4-8, both β and λ are set to 5. σ is set to 0.001. In the fig: 4, it's obvious that the $\boldsymbol{B}$ of SGNMF with f_1, f_2, f_3, f_4, f_5 are smoother than NMF and GNMF, in fig: 5, only SGNMF with f_1 learns a smoother result ,but for SGNMF with f_5, a much sparser $\boldsymbol{B}$ is gotten on USPS. In the fig: 6 SGNMF with the proposed f_5 also learns a much sparser matrix, but for f_1, f_2, f_3, f_4, they learn a smoother matrix. In the fig: 7 only SGNMF with f_1 learns a smoother matrix $\boldsymbol{B}$, others shown in fig: 7(d)-7(g) learn sparser $\boldsymbol{B}$. It's concluded that SGNMF with f_{1-4} may lead smoother results other than sparser results. But if using our proposed functions f_5 as a regularizer, the

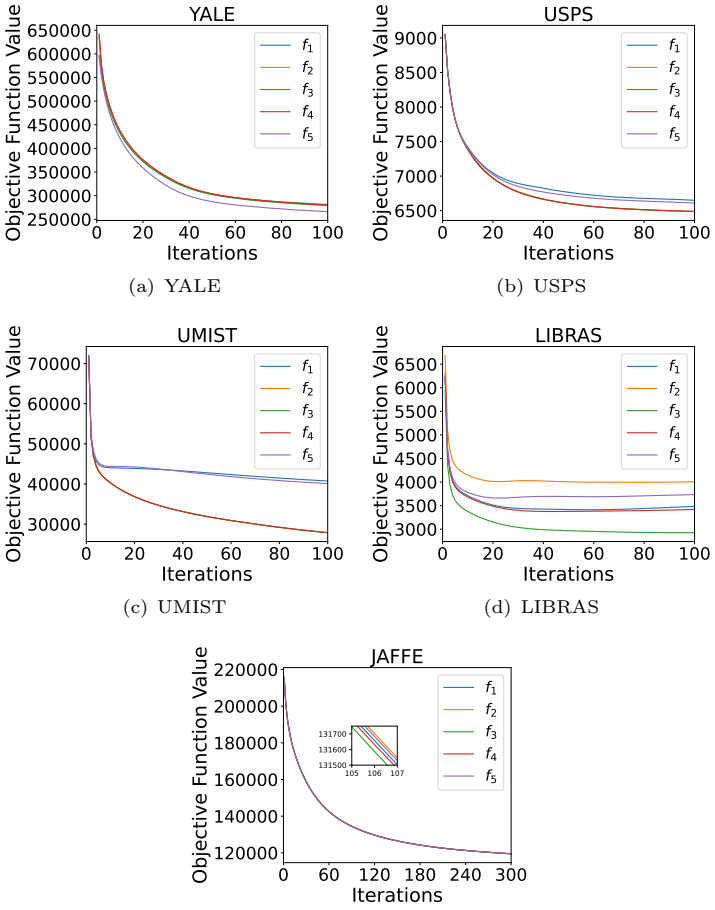
[7] https://github.com/chen12304/SGNMF/tree/main/pic_spar

Table 2 Accuracy of the algorithms' results.

Dataset		YALE (%)	USPS(%)	UMIST(%)	LIBRAS(%)	JAFFE(%)
K-means		58.24 ± 3.11	64.24 ± 1.37	42.28 ± 2.04	44.69 ± 2.26	82.16 ± 6.18
K-means++		60.82 ± 2.47	$\mathbf{65.05 \pm 2.12}$	42.29 ± 1.66	45.42 ± 2.27	88.57 ± 4.15
SP-cluster		$\mathbf{66.55 \pm 1.22}$	59.2 ± 0.56	59.2 ± 2.32	50.38 ± 1.23	79.84 ± 7.1
NMF		59.09 ± 1.85	57.25 ± 1.58	58.09 ± 1.53	46.73 ± 3.36	85.8 ± 5.303
KKM		41 ± 2.71	45.12 ± 5.49	43.82 ± 5.34	47.55 ± 6.83	62.54 ± 7.25
RKKM		41.06 ± 2.7	45.3 ± 5.47	43.84 ± 5.36	47.85 ± 6.87	62.77 ± 7.52
AASC		40.64 ± 2.63	52.54 ± 2.27	47.15 ± 2.72	47.34 ± 1.1	30.35 ± 1.05
RMKKM		52.18 ± 3.92	63.88 ± 7.41	57.45 ± 6.46	$\mathbf{62.85 \pm 8.15}$	87.07 ± 5.69
CFSFDP		**64.85**	58.75	51.13	46.94	84.04
SGNMF	f_1	62.3 ± 0.62	$\mathbf{65.18 \pm 2.7}$	61.53 ± 2.96	51.46 ± 1.3	$\mathbf{92.32 \pm 0.27}$
	f_2	62.21 ± 1.45	$\mathbf{65.6 \pm 1.84}$	61.69 ± 1.23	$\mathbf{52.43 \pm 0.56}$	92.3 ± 0.29
	f_3	62 ± 0.47	61.5 ± 0.35	$\mathbf{62.3 \pm 0.48}$	51.32 ± 0.61	$\mathbf{92.44 \pm 0.53}$
	f_4	62.27 ± 0.79	64.16 ± 2.98	$\mathbf{62.2 \pm 0.72}$	51.08 ± 0.62	92.02 ± 1.05
	f_5	$\mathbf{62.45 \pm 1.11}$	64.53 ± 1.11	$\mathbf{62.23 \pm 0.7}$	$\mathbf{52.47 \pm 0.41}$	$\mathbf{92.42 \pm 0.22}$

Table 3 NMI of algorithms' results.

Dataset		YALE (%)	USPS(%)	UMIST(%)	LIBRAS(%)	JAFFE (%)
K-means		65.65 ± 1.52	62.72 ± 1.69	64.93 ± 1.22	59.45 ± 1.3	86.75 ± 3.83
K-means++		66.54 ± 1.09	62.48 ± 1.25	65.32 ± 1.39	59.51 ± 1.18	89.86 ± 2.39
SP-cluster		$\mathbf{69.16 \pm 0.98}$	64.71 ± 0.32	81.71 ± 1.7	64.07 ± 0.27	87.82 ± 3.65
NMF		64.63 ± 0.8	57.22 ± 1.08	57.8 ± 1.14	59.55 ± 1.06	$\mathbf{90.19 \pm 0.79}$
KKM		45.71 ± 2.45	40.22 ± 5.6	35.04 ± 3.44	42.45 ± 6.05	69.62 ± 5.5
RKKM		46.01 ± 2.58	40.57 ± 5.43	35.06 ± 3.46	42.82 ± 5.94	70.17 ± 5.65
AASC		46.83 ± 2.68	41.94 ± 1.01	39.39 ± 1.61	43.97 ± 1.45	27.22 ± 0.77
RMKKM		55.58 ± 2.6	62.57 ± 5.64	56.33 ± 4.14	63.52 ± 5.91	89.37 ± 2.9
CFSFDP		**67.3**	58.7	74.9	64.19	**92.43**
SGNMF	f_1	66.76 ± 0.8	$\mathbf{65.58 \pm 0.55}$	81.88 ± 1.67	$\mathbf{68.06 \pm 0.59}$	90.04 ± 0.25
	f_2	$\mathbf{67.36 \pm 0.95}$	$\mathbf{65.43 \pm 0.55}$	82.85 ± 0.8	$\mathbf{68.04 \pm 0.35}$	90.08 ± 0.29
	f_3	66.94 ± 0.51	59.64 ± 0.26	$\mathbf{83.24 \pm 0.72}$	66.81 ± 0.43	89.97 ± 0.59
	f_4	67 ± 0.89	65.28 ± 0.97	$\mathbf{83.15 \pm 0.65}$	66.66 ± 0.55	89.81 ± 0.71
	f_5	67.11 ± 0.76	$\mathbf{65.35 \pm 1.12}$	$\mathbf{83.3 \pm 0.62}$	$\mathbf{68.08 \pm 0.19}$	$\mathbf{90.11 \pm 0.32}$



(a) YALE (b) USPS

(c) UMIST (d) LIBRAS

(e) JAFFE

Fig. 3 Convergence behaviors of the objective function value on five datasets.

smoother results can be avoided and sparser result can be gotten. Besides, it's shown in Table 4 that SGNMF with the proposed f_5 always gets the sparsest $\boldsymbol{B}$.

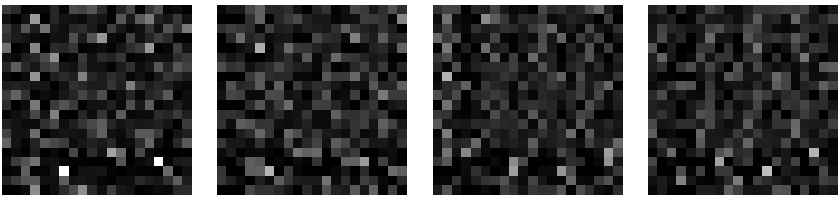
5.6 Parameter Setting

GNMF has two parameters(p and λ), p is used to construct the graph, λ is used to ensure local invariance. Two more parameters(β and σ) are introduced than GNMF. β is used to ensure the sparsity of the result, σ is used to ensure $f(x, \sigma)$ close to l_0 norm. The impact of β and σ to the performance on different datasets is shown in figs: 9 and 10. The ACC and NMI plotted in figure are the mean result of 20 times.

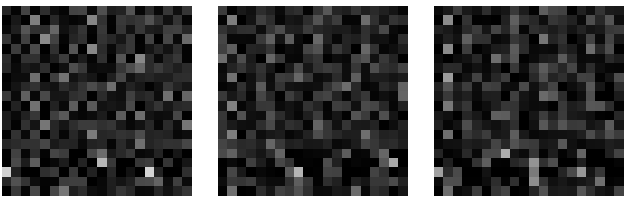
For β, it's shown in figs: 9 that performance decrease with a larger β on

Table 4 The Sparsity of the algorithms' results.

dataset		YALE(%)	USPS(%)	UMIST(%)	LIBRAS(%)	JAFFE(%)
NMF		6.91	42.88	9.70	7.59	2.95
GNMF		8.57	2.20	1.17	0.33	3.33
SGNMF	f1	7.72	1.40	1.71	0.46	2.86
	f2	8.40	33.5	11.73	37.89	3.28
	f3	9.25	24.45	12.17	27.04	3.66
	f4	9.05	38.03	16.10	55.04	4.69
	f5	**15.19**	**95.50**	**86.96**	**97.04**	**13.29**

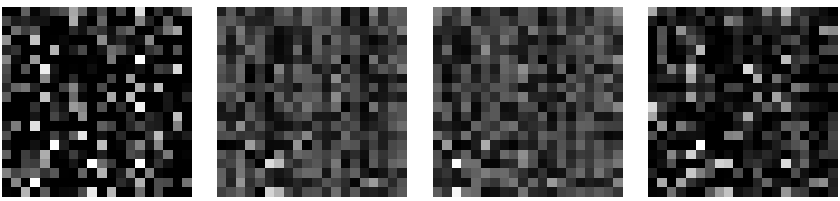


(a) NMF (b) GNMF (c) SGNMF with f_1 (d) SGNMF with f_2

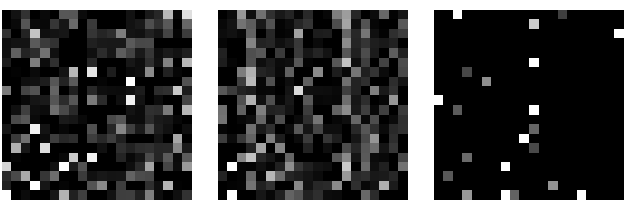


(e) SGNMF with f_3 (f) SGNMF with f_4 (g) SGNMF with f_5

Fig. 4 The comparisons of sparsity performance on YALE.

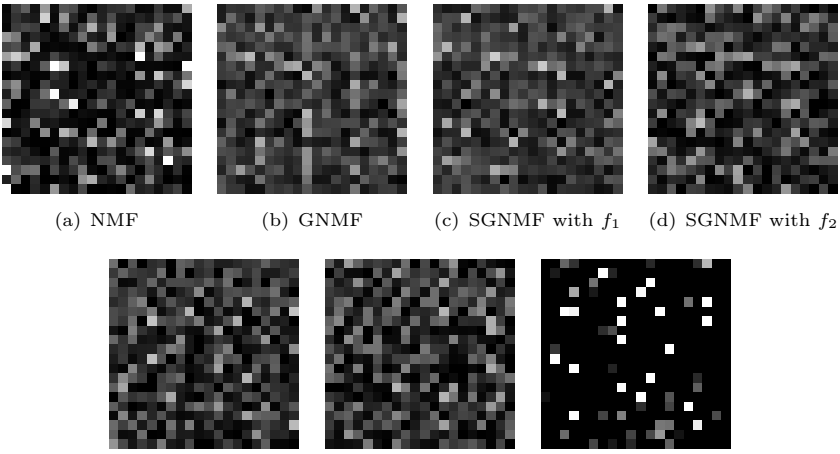


(a) NMF (b) GNMF (c) SGNMF with f_1 (d) SGNMF with f_2



(e) SGNMF with f_3 (f) SGNMF with f_4 (g) SGNMF with f_5

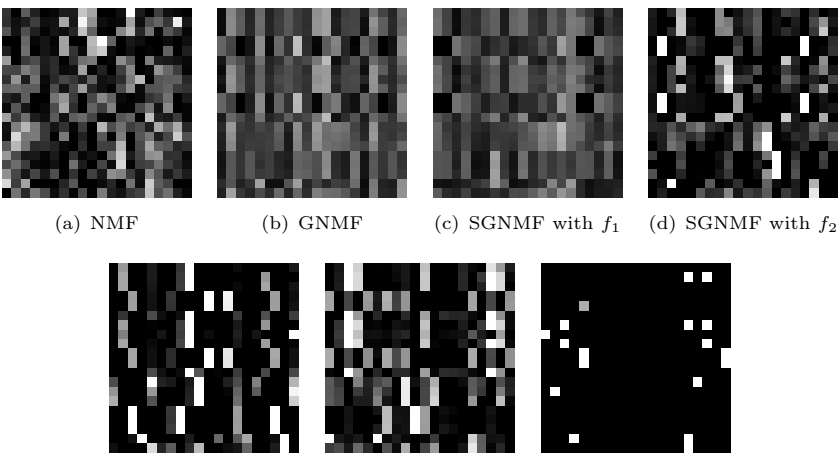
Fig. 5 The comparisons of sparsity performance on USPS.



(a) NMF (b) GNMF (c) SGNMF with f_1 (d) SGNMF with f_2

(e) SGNMF with f_3 (f) SGNMF with f_4 (g) SGNMF with f_5

Fig. 6 The comparisons of sparsity performance on UMIST.



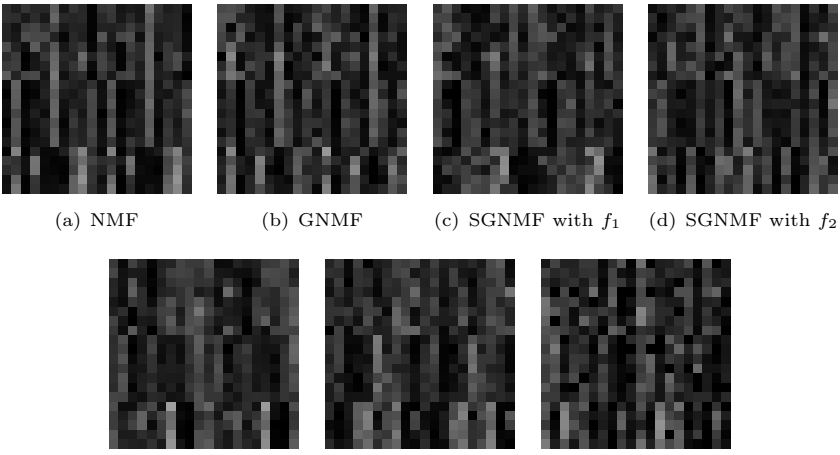
(a) NMF (b) GNMF (c) SGNMF with f_1 (d) SGNMF with f_2

(e) SGNMF with f_3 (f) SGNMF with f_4 (g) SGNMF with f_5

Fig. 7 The comparisons of sparsity performance on LIBRAS.

the whole. But in figs: 9(d) and 9(b), if $\beta \in (10^{-2}, 10^{-1})$, the performance increases with larger β and gets the best performance.

For σ, it's shown in figs: 10(e) and 10(f) that if $\sigma \in [10^{-4}, 10^{-3}]$ or $\sigma \in [1, 10]$, better performance can be gotten. If $\sigma \in [10^{-4}, 1]$, SGNMF with f_5 gets a much worse performance. Similarly, in figs: 10(a)-10(d), if $\sigma \in [10^{-4}, 10^{-1}]$, SGNMF with f_5 gets a much worse performance, but SGNMF with f_5 get the best performance when σ is set to an appropriate value in these figures. Such as $\sigma = 1$ in fig: 10(a), $\sigma = 10^{-2}$ in fig: 10(b), $\sigma = 4 \times 10^{-2}$ in both figs: 10(c) and 10(d). In summary, when using SGNMF with f_5, σ can be set in



(a) NMF (b) GNMF (c) SGNMF with f_1 (d) SGNMF with f_2

(e) SGNMF with f_3 (f) SGNMF with f_4 (g) SGNMF with f_5

Fig. 8 The comparisons of sparsity performance on JAFFE.

$(10^{-2}, 10^{-1})$ or larger than 1. For f_1, f_2, f_3 and f_4, σ is better to be set to larger than 1 or smaller than 10^{-2}.

6 Conclusions

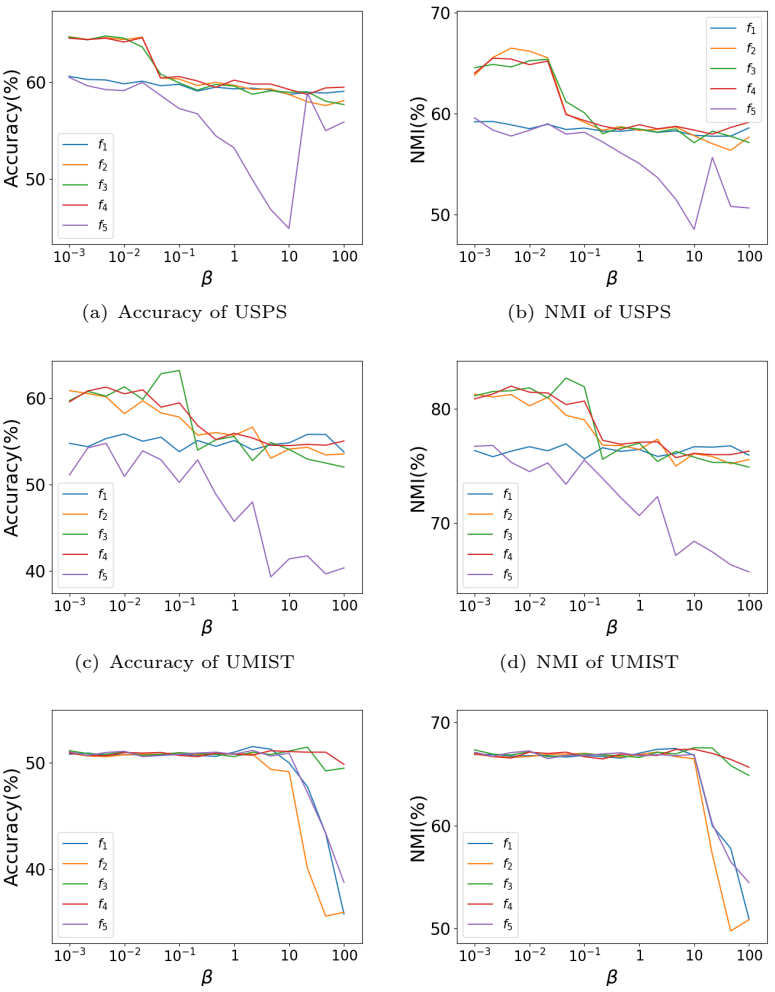
In this paper, sparse graph nonnegative matrix factorization is formulated as a global optimization problem by using the sum of the different smooth functions to approximate l_0 norm. A general algorithm with guaranteed convergence is designed. The clustering results on five public datasets show the proposed approach can enhance robustness of GNMF with high sparsity.

Acknowledgments. The work is supported by National Natural Science Foundation of China (Grant No. 62003281), in part by the Fundamental Research Funds for the Central Universities (Grant No.SWU020006) and in part by Natural Science Foundation of Chongqing, China (No.cstc2021jcyj-msxmX1169).

Declarations

Conflicts of interest/Competing interests. The authors declare that they have no competing interests.

Code availability. All data, algorithms, and code generated or used during the study appear in https://github.com/chen12304/SGNMF.



(a) Accuracy of USPS (b) NMI of USPS

(c) Accuracy of UMIST (d) NMI of UMIST

(e) Accuracy of LIBRAS (f) NMI of LIBRAS

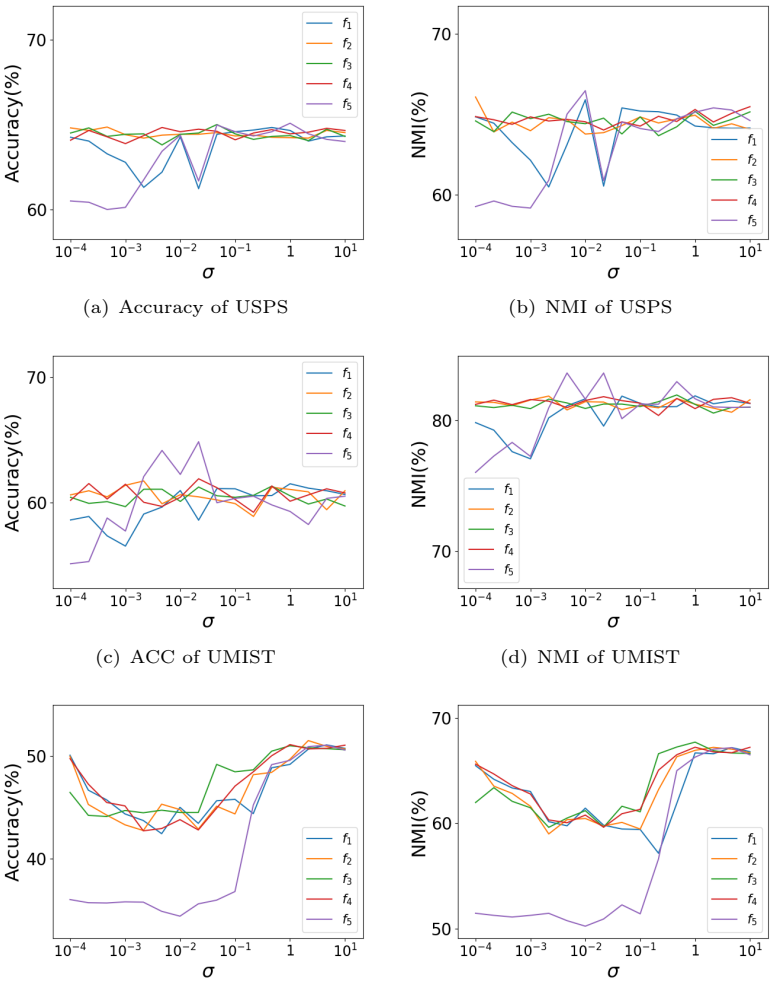
Fig. 9 β is set as $\{10^{-3}, 4\times10^{-3}, 7\times10^{-3}, 10^{-2}, \ldots, 100\}$ to show the effect on accuracy and NMI.

References

[1] Palmer, S.E.: Hierarchical structure in perceptual representation. Cognitive Psychology **9**(4), 441–474 (1977)

[2] Wachsmuth, E., Oram, M., Perrett, D.: Recognition of objects and their component parts: responses of single units in the temporal cortex of the macaque. Cerebral Cortex **4**(5), 509–522 (1994)

[3] Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative



(a) Accuracy of USPS (b) NMI of USPS

(c) ACC of UMIST (d) NMI of UMIST

(e) Accuracy of LIBRAS (f) NMI of LIBRAS

Fig. 10 σ is set as $\{10^{-4}, 4\times10^{-4}, 7\times10^{-4}, 10^{-3}, \ldots, 10\}$ to show the effect on accuracy and NMI.

matrix factorization. Nature **401**(6755), 788–791 (1999)

[4] Lu, X., Dong, L., Yuan, Y.: Subspace clustering constrained sparse nmf for hyperspectral unmixing. IEEE Transactions on Geoscience and Remote Sensing **58**(5), 3007–3019 (2019)

[5] Hassani, A., Iranmanesh, A., Mansouri, N.: Text mining using nonnegative matrix factorization and latent semantic analysis. Neural Comput & Applic **33**(20), 13745–13766 (2021). https://doi.org/10.1007/s00521-021-06014-6. Accessed 2022-02-22

[6] Yu, N., Wu, M.-J., Liu, J.-X., Zheng, C.-H., Xu, Y.: Correntropy-based hypergraph regularized nmf for clustering and feature selection on multi-cancer integrated data. IEEE Transactions on Cybernetics **51**(8), 3952–3963 (2021). https://doi.org/10.1109/TCYB.2020.3000799

[7] Jiao, C.-N., Gao, Y.-L., Yu, N., Liu, J.-X., Qi, L.-Y.: Hyper-graph regularized constrained nmf for selecting differentially expressed genes and tumor classification. IEEE Journal of Biomedical and Health Informatics **24**(10), 3002–3011 (2020)

[8] Hoyer, P.: Nonnegative matrix factorization with sparseness constraints. Journal of Machine Learning Research - JMLR **5**, 1457–1469 (2004)

[9] Dong, L., Yuan, Y., Luxs, X.: Spectral–spatial joint sparse nmf for hyperspectral unmixing. IEEE Transactions on Geoscience and Remote Sensing **59**(3), 2391–2402 (2020)

[10] Huang, S., Xu, Z., Kang, Z., Ren, Y.: Regularized nonnegative matrix factorization with adaptive local structure learning. Neurocomputing **382**, 196–209 (2020)

[11] Che, H., Wang, J., Zhang, W.: A collaborative neurodynamic approach to sparse coding. In: International Symposium on Neural Networks, pp. 454–462 (2019). Springer

[12] Che, H., Wang, J.: A collaborative neurodynamic approach to symmetric nonnegative matrix factorization. In: International Conference on Neural Information Processing, pp. 453–462 (2018). Springer

[13] Che, H., Wang, J., Cichocki, A.: Bicriteria sparse nonnegative matrix factorization via two-timescale duplex neurodynamic optimization. IEEE Transactions on Neural Networks and Learning Systems, 1–11 (2021). https://doi.org/10.1109/TNNLS.2021.3125457

[14] Li, X., Wang, J., Kwong, S.: A discrete-time neurodynamic approach to sparsity-constrained nonnegative matrix factorization. Neural Computation **32**(8), 1531–1562 (2020)

[15] Dai, X., Li, C., He, X., Li, C.: Nonnegative matrix factorization algorithms based on the inertial projection neural network. Neural Comput & Applic **31**(8), 4215–4229 (2019). https://doi.org/10.1007/s00521-017-3337-5. Accessed 2022-02-22

[16] Cai, D., He, X., Han, J., Huang, T.S.: Graph regularized nonnegative matrix factorization for data representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **33**(8), 1548–1560 (2011). https://doi.org/10.1109/TPAMI.2010.231

[17] Rajabi, R., Ghassemian, H.: Hyperspectral data unmixing using gnmf method and sparseness constraint. In: 2013 IEEE International Geoscience and Remote Sensing Symposium-IGARSS, pp. 1450–1453 (2013). IEEE

[18] Zhu, R., Liu, J.-X., Zhang, Y.-K., Guo, Y.: A robust manifold graph regularized nonnegative matrix factorization algorithm for cancer gene clustering. Molecules **22**(12), 2131 (2017). https://doi.org/10.3390/molecules22122131

[19] Dai, X., Chen, G., Li, C.: A discriminant graph nonnegative matrix factorization approach to computer vision. Neural Comput & Applic **31**(11), 7879–7889 (2019). https://doi.org/10.1007/s00521-018-3608-9. Accessed 2022-02-22

[20] Mu, J., Dai, L., Liu, J.-X., Shang, J., Xu, F., Liu, X., Yuan, S.: Automatic detection for epileptic seizure using graph-regularized nonnegative matrix factorization and bayesian linear discriminate analysis. Biocybernetics and Biomedical Engineering **41**(4), 1258–1271 (2021)

[21] Huang, S., Wang, H., Li, T., Li, T., Xu, Z.: Robust graph regularized nonnegative matrix factorization for clustering. Data Mining and Knowledge Discovery **32**(2), 483–503 (2018)

[22] Wang, D., Liu, J.-X., Gao, Y.-L., Zheng, C.-H., Xu, Y.: Characteristic gene selection based on robust graph regularized non-negative matrix factorization. IEEE/ACM Transactions on Computational Biology and Bioinformatics **13**(6), 1059–1067 (2016)

[23] Mohimani, H., Babaie-Zadeh, M., Jutten, C.: A fast approach for overcomplete sparse decomposition based on smoothed l_0 norm. IEEE Transactions on Signal Processing **57**(1), 289–301 (2008)

[24] Wang, Y., Li, X., Wang, J.: A neurodynamic approach to l_0-constrained optimization. In: 2020 12th International Conference on Advanced Computational Intelligence (ICACI), pp. 44–50 (2020). IEEE

[25] Li, X., Wang, J., Kwong, S.: Sparse nonnegative matrix factorization based on a hyperbolic tangent approximation of l_0 norm and neurodynamic optimization. In: 2020 12th International Conference on Advanced Computational Intelligence (ICACI), pp. 542–549 (2020). IEEE

[26] Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Proceedings of the 13th International Conference on Neural Information Processing Systems. NIPS'00, pp. 535–541. MIT Press, Cambridge, MA, USA (2000)

[27] Che, H., Wang, J.: A nonnegative matrix factorization algorithm based on a discrete-time projection neural network. Neural Networks **103**, 63–71 (2018)

[28] Guo, Z., Wang, J.: A neurodynamic optimization approach to constrained sparsity maximization based on alternative objective functions. In: The 2010 International Joint Conference on Neural Networks (IJCNN), pp. 1–8 (2010). IEEE

[29] Wang, L., Ye, P., Xiang, J.: A modified algorithm based on smoothed l_0 norm in compressive sensing signal reconstruction. In: 2018 25th IEEE International Conference on Image Processing (ICIP), pp. 1812–1816 (2018). IEEE

[30] Xiang, J., Yue, H., Yin, X., Ruan, G.: A reweighted symmetric smoothed function approximating l_0-norm regularized sparse reconstruction method. Symmetry **10**(11), 583 (2018)

[31] Boutsidis, C., Gallopoulos, E.: Svd based initialization: A head start for nonnegative matrix factorization. Pattern Recognition **41**(4), 1350–1362 (2008). https://doi.org/10.1016/j.patcog.2007.09.010

[32] Dua, D., Graff, C.: UCI Machine Learning Repository (2017). http://archive.ics.uci.edu/ml

[33] Huang, H.-C., Chuang, Y.-Y., Chen, C.-S.: Affinity aggregation for spectral clustering. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 773–780 (2012). https://doi.org/10.1109/CVPR.2012.6247748

[34] Du, L., Zhou, P., Shi, L., Wang, H., Fan, M., Wang, W., Shen, Y.-D.: Robust multiple kernel k-means using $l_{2;\,1}$-norm. In: Proceedings of the 24th International Conference on Artificial Intelligence, pp. 3476–3482 (2015). IJCAI

[35] Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. Science **344**(6191), 1492–1496 (2014)

[36] Peharz, R., Pernkopf, F.: Sparse nonnegative matrix factorization with l_0-constraints. Neurocomputing **80**, 38–46 (2012). https://doi.org/10.1016/j.neucom.2011.09.024. Special Issue on Machine Learning for Signal Processing 2010