Engineering Applications of Artificial Intelligence 69 (2018) 24–35

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

# Dual-graph regularized non-negative matrix factorization with sparse and orthogonal constraints

Yang Meng [a], Ronghua Shang [a,*], Licheng Jiao [a], Wenya Zhang [b], Shuyuan Yang [a]

[a] Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an, Shaanxi Province 710071, China
[b] School of Computer Sciences, Xidian University, Xi'an, Shaanxi Province 710071, China

ARTICLE INFO

*Keywords:*
Semi-supervised non-negative matrix factorization
Dual-graph model
Orthogonal constraint
Bi-orthogonal constraints
Cluster

ABSTRACT

Semi-supervised Non-negative Matrix Factorization (NMF) can not only utilize a fraction of label information, but also effectively learn local information of the objectives, such as documents and faces. Semi-supervised NMF is an efficient technique for dimensionality reduction of high dimensional data. In this paper, we propose a novel semi-supervised NMF, called Dual-graph regularized Non-negative Matrix Factorization with Sparse and Orthogonal constraints (SODNMF). Dual-graph model is added into semi-supervised NMF, and the manifold structures of the data space and the feature space are taken into account simultaneously. In addition, the sparse constraint is used in SODNMF, which can simplify the calculation and accelerate the processing speed. The most important is that SODNMF makes use of bi-orthogonal constraints, which can avoid the non-correspondence between images and basic vectors. Therefore, it can effectively enhance the discrimination and the exclusivity of clustering, and improve the clustering performance. We give the objective function, the iterative updating rules and the convergence proof. Empirical experiments demonstrate encouraging results of our novel algorithm in comparison to four algorithms within some state-of-the-art algorithms through a set of evaluations based on three real datasets.



## 1. Introduction

With the advent of big data era, the amount of data increases more and more, and the dimensionality of data becomes larger and larger (Shang et al., 2016a; Lee and Seung, 1999). How to deal with massive high dimensional data and find an appropriate low dimensional representation of data are important issues in data mining and analysis (Shang et al., 2017, 2016b). Such as the applications of data mining and analysis in commodity recommendation and security monitoring (Tian and Chen, 2017). An effective low dimensional representation of data can not only mine the latent structural information of data (Ma et al., 2016), but also remove the redundant features in the original data and rapidly deal with massive high dimensional data (Gu et al., 2017).

Matrix factorization is an efficient dimensionality reduction technique, which can reduce the dimensionality of high dimensional data. There are many classical matrix factorization techniques, such as Singular Value Decomposition (SVD) (Duda et al., 2012), Principal Component Analysis (PCA) (Jolliffe, 1986), Linear Discriminant Analysis (LDA) (Pang et al., 2014), Non-negative Matrix Factorization (NMF) (Zheng et al., 2007; Paatero and Tapper, 1994) and so on. However, compared with other matrix factorization technique, NMF can effectively learn local information of the objectives, such as documents and faces. Therefore, NMF can make better performance on document clustering (Lee and Seung, 1999; Paatero and Tapper, 1994; Xu et al., 2003; Shahnaz et al., 2006), face recognition (Lee and Seung, 1999; Li et al., 2001) and other practical applications.

NMF aims to decompose the original high dimensional data matrix into two low dimensional data matrices, and the product of the two low dimensional data matrices approximates the original high dimensional data matrix as far as possible. In this way, we can reduce the dimensionality of the original high dimensional data. Classical NMF is an unsupervised learning algorithm, which has been widely used in data clustering. However, there is often a fraction of label information in the original data in the real world, and the classical NMF algorithms cannot make full use of the label information in the original data. Many machine learning researchers have found that the semi-supervised algorithm using a fraction of label information can improve the accuracy of learning (Belkin et al., 2006; Feng et al., 2016; Yang et al., 2014), so the accuracy of unsupervised NMF learning is inferior to many

* Corresponding author.
*E-mail address:* rhshang@mail.xidian.edu.cn (R. Shang).

https://doi.org/10.1016/j.engappai.2017.11.008
Received 9 May 2017; Received in revised form 4 September 2017; Accepted 24 November 2017
Availableonline 20 December 2017
0952-1976/

semi-supervised algorithms. Recently, in order to make full use of the label information, many scholars have improved the unsupervised NMF and proposed many semi-supervised NMF algorithms (Liu et al., 2012; Babaee et al., 2016), which can not only take advantages of the local information of the object in NMF, but also effectively utilize a fraction of label information to improve the accuracy of NMF learning. In Liu et al. (2012), Liu et al. proposed a semi-supervised NMF called Constrained Non-negative Matrix Factorization (CNMF) which embeds the label information as hard constraints into the objective function of NMF. In the new low dimensional representation space, the points with the same label have the same coordinates. In Discriminative Nonnegative Matrix Factorization (DNMF) (Babaee et al., 2016), a fraction of label information is introduced by coupling discriminative regularizer to the objective function of the semi-supervised NMF. However, many algorithms like CNMF and DNMF cannot exploit the local geometric information of data and make full use of the potential structural information. In addition, the previous semi-supervised NMF algorithms do not take advantages of the sparsity of matrices, which results in the complex calculation and the long optimization time. The non-correspondence between images and basic vectors also makes the previous semi-supervised NMF algorithms lack of discrimination.

In order to solve the above problems, we propose a novel semi-supervised non-negative matrix factorization algorithm, called Dual-graph regularized Non-negative Matrix Factorization with Sparse and Orthogonal constraints (SODNMF). Motivated by recent progress in dual regularization (Sindhwani et al., 2009; Gu and Zhou, 2009) and structural information (Ma et al., 2016; Gu et al., 2017), we combine dual-graph model with semi-supervised NMF to make full use of the potential structural information, so the manifold structures of the data space and the feature space are taken into account simultaneously. In addition, inspired by the recent development of sparse constraint (Luo and Zhang, 2014) and orthogonal constraint (Ding et al., 2006), we introduce sparse constraint and bi-orthogonal constraints into semi-supervised NMF, which can not only overcome the disadvantage of the slow optimization and the complex calculation in many existing semi-supervised NMF algorithms, but also avoid the non-correspondence between images and basic vectors to effectively enhance the discrimination and the exclusivity of clustering. In order to prove the efficiency and effectiveness of our algorithm, we give the convergence proof of the objective function and the experimental results of SODNMF and other related algorithms on three real datasets ORL, PIE and TDT2.

Our main contributions are the following four aspects:

1. Use of the iterative updating rules derived from ordinary non-negative matrix factorization incorporating all the constraints may not induce a reliable solution due to scaling issue in several low dimensional matrices. To avoid this problem, we introduce a diagonal scaling matrix into semi-supervised NMF.
2. Dual-graph model is added into semi-supervised NMF, which constructs two neighbor graphs of the data space and the feature space respectively. In this way, we can make full use of the potential structural information on account of preserving the manifold structures of the data space and the feature space simultaneously.
3. Sparse constraint with $L_{2,1/2}$-norm on the coefficient matrix in the feature space is incorporated as the additional condition, which can not only make the coefficient matrix with a good sparsity and simplify the calculation, but also enhance the local learning ability and robustness of the algorithm.
4. Bi-orthogonal constraints are adopted in SODNMF. Each image can correspond to the unique basic vector with the orthogonal constraint on the coefficient matrix in the feature space, which can effectively enhance the discrimination of clustering. The orthogonal constraint on the basic matrix in the data space can enhance the exclusivity across the classes and improve the clustering performance.

The rest of the paper is organized as follows: In Section 2, we give an introduction of the classical NMF and some related NMF algorithms. In Section 3, we introduce the mathematical model and the solution procedure of our algorithm, and then prove the convergence of the optimization scheme. In Section 4, we provide a large number of experiments to demonstrate the efficiency and effectiveness of our algorithm. Finally, we draw a conclusion and provide suggestions for future work in Section 5.

## 2. Related work

### 2.1. NMF

NMF can obtain two low dimensional data matrices by decomposing the original high dimensional data matrix, and the product of the two low dimensional data matrices approximates the original data matrix as far as possible to find an appropriate low dimensional representation of the original data matrix. We have an original data matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \Re^{m\times n}$, where $m$ is the number of the feature dimensions, $n$ is the number of the samples. $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{im}]^T \in \Re^m$ is the $i$th vector. In NMF, the original high dimensional data matrix $\boldsymbol{X}$ should be decomposed into two low dimensional data matrices $\boldsymbol{U} = [u_{ij}] \in \Re^{m\times k}$ and $\boldsymbol{V} = [v_{ij}]^T \in \Re^{n\times k}$, where $k$ is the clustering number, $\boldsymbol{U}$ is the coefficient matrix in the feature space and $\boldsymbol{V}$ is the basic matrix in the data space. The purpose of NMF is to let the product of the coefficient matrix $\boldsymbol{U}$ and the basic matrix $\boldsymbol{V}$ approximate the original data matrix $\boldsymbol{X}$ as far as possible:

$$\boldsymbol{X} \approx \boldsymbol{U}\boldsymbol{V}^T. \tag{1}$$

In other words, we should minimize the following residual error matrix:

$$O = \|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^T\|_F^2 \quad s.t.\ u_{ij} \geq 0, v_{ij} \geq 0 \tag{2}$$

where $\|\cdot\|_F$ is Frobenius norm (F-norm) of the matrix. We can get the Euclidean distance of two matrices by calculating the square of the F-norm. In Zheng et al. (2007), Lee et al. provided the iterative updating rules to solve such a minimization problem, and prove the convergence. The iterative updating rules of the coefficient matrix $\boldsymbol{U}$ and the basic matrix $\boldsymbol{V}$ of NMF are as follows:

$$u_{ij} = u_{ij}\frac{(\boldsymbol{X}\boldsymbol{V})_{ij}}{(\boldsymbol{U}\boldsymbol{V}^T\boldsymbol{V})_{ij}} \tag{3}$$

$$v_{ij} = v_{ij}\frac{(\boldsymbol{X}^T\boldsymbol{U})_{ij}}{(\boldsymbol{V}\boldsymbol{U}^T\boldsymbol{U})_{ij}}. \tag{4}$$

First of the iterative process, we initialize the coefficient matrix $\boldsymbol{U}$ and the basic matrix $\boldsymbol{V}$ randomly, and then update them according to the iterative updating rules in formula (3) and (4) until the final condition is reached.

### 2.2. GNMF

In Cai et al. (2011), Cai et al. proposed Graph Regularized Nonnegative Matrix Factorization (GNMF) which adds manifold learning into the classical NMF (Belkin et al., 2006; Cai et al., 2009a, b). GNMF constructs a neighbor graph to simulate the local geometric structure of data. For $n$ samples, a neighbor graph with $n$ vertices is constructed, and each vertex corresponds to a sample. For the vertex $x_i$, we aim to find its $k$-nearest neighbors and establish the edges and weights with $x_i$ which represent the similarities between them. Therefore, the weight matrix $\boldsymbol{W}$ is also called the similarity matrix. There are many methods to construct the weight matrix in the neighbor graph, the common methods are (Cai et al., 2011): 0–1 weighting, heat kernel weighting and dot-product weighting. Then, $Tr(\boldsymbol{V}^T\boldsymbol{L}\boldsymbol{V})$ can be used to measure the smoothness of the low dimensional representation, so the objective function of GNMF is measured as follows:

$$O = \|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^T\|_F^2 + Tr(\boldsymbol{V}^T\boldsymbol{L}\boldsymbol{V}) \quad s.t.\ u_{ij} \geq 0, v_{ij} \geq 0. \tag{5}$$

where the Laplacian matrix is $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, $\boldsymbol{D}$ is a diagonal matrix and $[\boldsymbol{D}]_{ii} = \sum_j [\boldsymbol{W}]_{ij}$, $Tr(\cdot)$ is the trace of the matrix. The iterative updating rules of the coefficient matrix $\boldsymbol{U}$ and the basic matrix $\boldsymbol{V}$ of GNMF are as follows:

$$u_{ij} \leftarrow u_{ij} \frac{(\boldsymbol{XV})_{ij}}{(\boldsymbol{UV}^T\boldsymbol{V})_{ij}}, \tag{6}$$

$$v_{ij} \leftarrow v_{ij} \frac{(\boldsymbol{X}^T\boldsymbol{U} + \lambda \boldsymbol{WV})_{ij}}{(\boldsymbol{VU}^T\boldsymbol{U} + \lambda \boldsymbol{DV})_{ij}}. \tag{7}$$

We can see from the objective function of GNMF, the second item is the manifold regularization that contains the local geometric information, and the iterative updating rule of the basic matrix $\boldsymbol{V}$ in formula (7) also uses the local geometric information.

### 2.3. CNMF

NMF can find the local linear representation of the original data and reduce the dimensionality of the original data. However, NMF is an unsupervised algorithm which cannot make full use of the label information in the original data. In order to solve this problem, Liu et al. proposed a semi-supervised non-negative matrix factorization algorithm called Constrained Nonnegative Matrix Factorization (CNMF) (Liu et al., 2012) which takes the label information as additional hard constraints into NMF. The core of CNMF is to construct a label constraint matrix $\boldsymbol{A}$ with label information, and then use the product of the label constraint matrix $\boldsymbol{A}$ and the label auxiliary matrix $\boldsymbol{Z}$ instead of the basic matrix $\boldsymbol{V}$, so the objective function of CNMF is measured as follows:

$$O = \|\boldsymbol{X} - \boldsymbol{UZ}^T\boldsymbol{A}^T\|_F^2 \quad s.t.\ u_{ij} \geq 0, z_{ij} \geq 0. \tag{8}$$

The iterative updating rules of the coefficient matrix $\boldsymbol{U}$ and the basic matrix $\boldsymbol{V}$ of GNMF are as follows:

$$u_{ij} \leftarrow u_{ij} \frac{(\boldsymbol{XAZ})_{ij}}{(\boldsymbol{UZ}^T\boldsymbol{A}^T\boldsymbol{AZ})_{ij}} \tag{9}$$

$$z_{ij} \leftarrow z_{ij} \frac{(\boldsymbol{A}^T\boldsymbol{X}^T\boldsymbol{U})_{ij}}{(\boldsymbol{A}^T\boldsymbol{AZU}^T\boldsymbol{U})_{ij}}. \tag{10}$$

We can see the objective function of CNMF and the iterative updating rules of the coefficient matrix $\boldsymbol{U}$ and the label auxiliary matrix $\boldsymbol{Z}$ all contain the label constraint matrix $\boldsymbol{A}$, so CNMF is a semi-supervised algorithm which is superior to unsupervised NMF and GNMF in the clustering performance.

### 2.4. DNMF

In Babaee et al. (2016), M. Babaee et al. proposed a new semi-supervised NMF called Discriminative Non-negative Matrix Factorization (DNMF) which utilizes the label information of a fraction of data as a discriminative constraint. Unlike CNMF, DNMF merges the samples with the same label into a same axis in the new representation instead of a single data point, the objective function of DNMF is measured as follows:

$$O = \|\boldsymbol{X} - \boldsymbol{UV}^T\|^2 + \alpha\|\boldsymbol{Q} - \boldsymbol{AV}_l^T\|^2 \quad s.t.\ u_{ij} \geq 0, z_{ij} \geq 0, \tag{11}$$

where $\boldsymbol{Q}_{ij}$ is 1 if sample $j$ is labeled and belongs to the $i$th class and 0 otherwise.

The iterative updating rules of the matrices $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{A}$ of DNMF are as follows:

$$u_{ij} \leftarrow u_{ij} \frac{(\boldsymbol{XV})_{ij}}{(\boldsymbol{UV}^T\boldsymbol{V})_{ij}} \tag{12}$$

$$v_{ij} \leftarrow v_{ij} \frac{(\boldsymbol{X}^T\boldsymbol{U} + \alpha(\boldsymbol{V}_l\boldsymbol{A}^T\boldsymbol{A})^- + \alpha(\boldsymbol{Q}^T\boldsymbol{A})^+)_{ij}}{(\boldsymbol{U}^T\boldsymbol{UV} + \alpha(\boldsymbol{V}_l\boldsymbol{A}^T\boldsymbol{A})^+ + \alpha(\boldsymbol{Q}^T\boldsymbol{A})^-)_{ij}} \tag{13}$$

$$\boldsymbol{A} \leftarrow \boldsymbol{QV}_l(\boldsymbol{V}_l^T\boldsymbol{V}_l)^{-1}. \tag{14}$$

Similarly, the objective function of DNMF and the iterative updating rules of the matrices $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{A}$ all contain the label constraint matrix $\boldsymbol{V}_l$, so DNMF is a semi-supervised algorithm.

### 2.5. GSNMFC

Based on CNMF, GSNMFC added the graph regular term and used F-norm as the sparse constraint (Sun et al., 2016). The mathematical model of GSNMFC is as follows:

$$O = \|\boldsymbol{X} - \boldsymbol{UZ}^T\boldsymbol{A}^T\|_F^2 + \lambda Tr(\boldsymbol{Z}^T\boldsymbol{A}^T\boldsymbol{LAZ}) + \beta\|\boldsymbol{U}\|_F^2 \quad s.t.\ u_{ij} \geq 0, z_{ij} \geq 0 \tag{15}$$

where the second term is a graph regular term with the label constraint matrix $\boldsymbol{A}$, and the third item is sparse constraint term.

## 3. Dual-graph regularized non-negative matrix factorization with sparse and orthogonal constraints

In recent years, researchers have found that semi-supervised algorithms utilizing the label information of a fraction of data can improve the accuracy of learning and have a better clustering performance in comparison with the unsupervised algorithms (Belkin et al., 2006; Feng et al., 2016; Yang et al., 2014). However, the existing semi-supervised NMF algorithm has many limitations. For example, there is no effective use of the local geometric information of the data space and the feature space. On the other hand, the data matrices with dimensionality reduction are lack of sparsity and discrimination, which result in the complex calculation and the poor clustering performance. Therefore, we propose a novel semi-supervised non-negative NMF called Dual-graph regularized Non-negative Matrix Factorization with Sparse and Orthogonal constraints (SODNMF) which combines dual-graph model with semi-supervised NMF and incorporates sparse and orthogonal constraints simultaneously.

### 3.1. Objective function

We have a dataset $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \Re^{m\times n}$, where $m$ is the number of the feature dimensions, $n$ is the number of the samples. In $\boldsymbol{X}$, the previous $l$ data points $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_l$ contain the label information, and the rest of the data points do not. We define a label indicator matrix $\boldsymbol{D} = [d_{ij}] \in \Re^{l\times c}$, and $d_{ij}$ is 1 if $\boldsymbol{x}_i$ belongs to the $j$th class and 0 otherwise, where $c$ is the clustering number.

The original high dimensional data matrix is decomposed into two low dimensional data matrices, and the product of the two low dimensional data matrices approximates the original data matrix. The two low dimensional data matrices are the coefficient matrix in the feature space $\boldsymbol{P} = [\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_m]^T \in \Re^{m\times c}$ and the basic matrix in the data space $\boldsymbol{S} = [\boldsymbol{S}_1, \boldsymbol{S}_2, \ldots, \boldsymbol{S}_n]^T \in \Re^{n\times c}$. $\boldsymbol{P}_i = [p_{i1}, p_{i2}, \ldots, p_{ic}] \in \Re^c$ is the $i$th coefficient vector, $\boldsymbol{S}_i = [s_{i1}, s_{i2}, \ldots, s_{ic}] \in \Re^c$ is the $i$th basic vector. The purpose of NMF is to let the product of the coefficient matrix $\boldsymbol{P}$ and the basic matrix $\boldsymbol{S}$ approximate the original data matrix $\boldsymbol{X}$ as far as possible. According to label indicator matrix $\boldsymbol{D}$, we construct the label constraint matrix $\boldsymbol{C}$ as follows:

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{D}_{l\times c} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I}_{n-l} \end{bmatrix}. \tag{16}$$

NMF maps the original data matrix $\boldsymbol{X}$ in an $m$-dimensional space to a basic matrix $\boldsymbol{S}$ in a $c$-dimensional space. However, the label constraint matrix $\boldsymbol{C} = [c_{ij}] \in \Re^{n\times(c+n-l)}$ is in a $(c + n - l)$-dimensional space, so we need to introduce a label auxiliary matrix $\boldsymbol{A} = [a_{ij}] \in \Re^{(c+n-l)\times k}$ to establish the correspondence between the label constraint matrix $\boldsymbol{C}$ and the basic matrix $\boldsymbol{S}$ as follows:

$$\boldsymbol{S} = \boldsymbol{CA}. \tag{17}$$

#### 3.1.1. Scaling adjustment

In Hong et al. (2016), Hong et al. proposed that use of the iterative updating rules derived from ordinary matrix factorization incorporating all constraints may not induce a reliable solution due to scaling issue in several low dimensional matrices (Hong et al., 2016). In this paper, we involve both sparse constraint and bi-orthogonal constraints together, so there are certain risks to induce an unreliable solution. To avoid this problem, we introduce a diagonal scaling matrix $\boldsymbol{R}=[r_{ij}] \in \Re^{c\times c}$, and then the semi-supervised NMF is transformed into the objective function as follows:

$$O = \|\boldsymbol{X} - \boldsymbol{P}\boldsymbol{R}\boldsymbol{S}^T\|_F^2 = \|\boldsymbol{X} - \boldsymbol{P}\boldsymbol{R}\boldsymbol{A}^T\boldsymbol{C}^T\|_F^2. \tag{18}$$

It can not only achieve a semi-supervised NMF which introduces the label information into the NMF, but also avoid obtaining an unreliable solution.

#### 3.1.2. Dual-graph model

Recent studies in manifold learning theory and spectral graph theory have proved that the local geometric structure can be effectively modeled through a nearest neighbor graph on a scatter of data points (Cai et al., 2011). For each data point, we find its neighbors and put edges between them. Specifically, if the two data points have the same label, a large weight can be set to the edge between them. If the two data points have different labels, the corresponding weight is assigned to be 0. There are many choices to define the similarity matrix on the graph. In this paper, the heat kernel weighting have been selected to construct the weight matrix in the neighbor graph, $\boldsymbol{W}_{ij} = e^{-\frac{\|x_i - x_j\|^2}{\sigma}}$ when the two data points $i$ and $j$ are connected.

In order to preserve the manifold structures of the data space and the feature space simultaneously and make full use of the potential structural information in the original data, we add dual-graph model into semi-supervised NMF. We construct two nearest neighbor graphs in the feature space and the data space, called feature graph and data graph.

We define a similarity matrix of the feature graph as $\boldsymbol{W}^P$. Based on the coefficient matrix $\boldsymbol{P}$, we can measure the smoothness of the low dimensional representation in the feature space as follows (Cai et al., 2011):

$$G^P = Tr(\boldsymbol{P}^T\boldsymbol{L}^P\boldsymbol{P}) \tag{19}$$

where the Laplacian matrix of the feature space is $\boldsymbol{L}^P = \boldsymbol{D}^P - \boldsymbol{W}^P$, $\boldsymbol{D}^P$ is a diagonal matrix and $[\boldsymbol{D}^P]_{ii} = \sum_j [\boldsymbol{W}^P]_{ij}$.

Similarly, we define a similarity matrix of the data graph as $\boldsymbol{W}^S$. Based on the basic matrix $\boldsymbol{S}$, we can measure the smoothness of the low dimensional representation in the data space as follows:

$$G^S = Tr(\boldsymbol{S}^T\boldsymbol{L}^S\boldsymbol{S}) = Tr(\boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{L}^S\boldsymbol{C}\boldsymbol{A}) \tag{20}$$

where the Laplacian matrix of the data space is $\boldsymbol{L}^S = \boldsymbol{D}^S - \boldsymbol{W}^S$, $\boldsymbol{D}^S$ is a diagonal matrix and $[\boldsymbol{D}^S]_{ii} = \sum_j [\boldsymbol{W}^S]_{ij}$.

#### 3.1.3. Sparse constraint

Recently, sparse constraint has attracted a lot of interest because it can select the discriminative sparse features to improve the efficiency and effectiveness of the algorithm. Sparse constraint aims to use an appropriate sparse model to achieve the sparse data representation. Although NMF can reduce the dimensionality of the original data, the operation of high dimensional data is still complex. Therefore, sparse constraint with $L_{2,1/2}$-norm on the coefficient matrix $\boldsymbol{P}$ is incorporated as the additional condition. In spite of the computational convenience, many researchers select $L_{2,1}$-norm as a sparse model (Li et al., 2012; Ma et al., 2012; Nie et al., 2010). In recent years, Xu et al. have concluded when $p$ is 1/2, the $L_p$-norm, *i.e.* $L_{1/2}$-norm has the best sparsity (Xu et al., 2012, 2010). Therefore, sparse constraint with $L_{2,1/2}$-norm on the coefficient matrix $\boldsymbol{P}$ is incorporated as follows (Xu et al., 2012, 2010):

$$\|\boldsymbol{P}\|_{2,1/2} = \left(\sum_{i=1}^{m} \|\boldsymbol{P}_i\|_2^{1/2}\right)^2. \tag{21}$$

Sparse constraint with $L_{2,1/2}$-norm can not only make the coefficient matrix $\boldsymbol{P}$ sparser and simplify the calculation, but also enhance the local learning ability and robustness of the algorithm and improve the clustering performance.

#### 3.1.4. Bi-orthogonal constraints

For given solution $(\boldsymbol{U}, \boldsymbol{V})$ of NMF: $\boldsymbol{X} \approx \boldsymbol{U}\boldsymbol{V}^T$, there are also some solution $(\boldsymbol{U}\boldsymbol{A}, \boldsymbol{V}\boldsymbol{B})$ when $\boldsymbol{A}\boldsymbol{B}^T = \boldsymbol{I}, \boldsymbol{U}\boldsymbol{A} \geq 0, \boldsymbol{V}\boldsymbol{B} \geq 0$ (Ding et al., 2006). In order to avoid this problem, the orthogonal constraint $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$ should be add in NMF. In addition, there is no good correspondence between images and the basic matrix in many existing NMF algorithms. Therefore, in order to let each image correspond to the unique basic vector in the basic matrix and effectively enhance the discriminant of clustering, we add the following orthogonal constraint on the coefficient matrix $\boldsymbol{P}$ in the feature space (Ding et al., 2006):

$$\boldsymbol{P}^T\boldsymbol{P} = \boldsymbol{I}. \tag{22}$$

Similarly, in order to enhance the exclusivity across the classes and improve the clustering performance, we add the following orthogonal constraint on the basic matrix $\boldsymbol{S}$ in the data space:

$$\boldsymbol{S}^T\boldsymbol{S} = \boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A} = \boldsymbol{I}. \tag{23}$$

The orthogonal constraints on the coefficient matrix $\boldsymbol{P}$ and the basic matrix $\boldsymbol{S}$ provide a strong capability of simultaneously clustering rows and columns, which we call bi-orthogonal constraints.

#### 3.1.5. Iterative updating rules

Taking the above factors into account, we define the objective function of Dual-graph regularized Non-negative Matrix Factorization with Sparse and Orthogonal constraints (SODNMF) as follows:

$$\begin{aligned} O_{SODNMF} &= \|\boldsymbol{X} - \boldsymbol{P}\boldsymbol{R}\boldsymbol{A}^T\boldsymbol{C}^T\|_F^2 + \alpha[Tr(\boldsymbol{P}^T\boldsymbol{L}^P\boldsymbol{P}) \\ &\quad + Tr(\boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{L}^S\boldsymbol{C}\boldsymbol{A})] + \theta\|\boldsymbol{P}\|_{2,1/2}^{1/2} \\ &= Tr(\boldsymbol{X}\boldsymbol{X}^T) - 2Tr(\boldsymbol{P}\boldsymbol{R}\boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{X}^T) \\ &\quad + Tr(\boldsymbol{P}\boldsymbol{R}\boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{A}\boldsymbol{R}^T\boldsymbol{P}^T) \\ &\quad + \alpha[Tr(\boldsymbol{P}^T\boldsymbol{L}^P\boldsymbol{P}) + Tr(\boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{L}^S\boldsymbol{C}\boldsymbol{A})] + \theta\|\boldsymbol{P}\|_{2,1/2}^{1/2} \\ &\quad \text{s.t. } \boldsymbol{P}^T\boldsymbol{P} = \boldsymbol{I}, \boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A} = \boldsymbol{I} \end{aligned} \tag{24}$$

where $\alpha$, $\theta$ are non-negative, which can balance the weight of the first reconstruction error term and the next several terms. $\alpha$ is the dual-graph parameter and $\theta$ is the sparse parameter.

In order to optimize this objective function, we define the bi-orthogonal parameter $\beta$ as the Lagrange multiplier to restrict the bi-orthogonal term, so we can translate the Lagrange function of the objective function in the formula (24) as follows:

$$\begin{aligned} L &= Tr(\boldsymbol{X}\boldsymbol{X}^T) - 2Tr(\boldsymbol{P}\boldsymbol{R}\boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{X}^T) + Tr(\boldsymbol{P}\boldsymbol{R}\boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A}\boldsymbol{R}^T\boldsymbol{P}^T) \\ &\quad + \alpha[Tr(\boldsymbol{P}^T\boldsymbol{L}^P\boldsymbol{P}) + Tr(\boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{L}^S\boldsymbol{C}\boldsymbol{A})] + \beta[Tr(\boldsymbol{P}^T\boldsymbol{P} - \boldsymbol{I}) \\ &\quad + Tr(\boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A} - \boldsymbol{I})] + 4\theta Tr(\boldsymbol{P}^T\boldsymbol{Q}\boldsymbol{P}) \end{aligned} \tag{25}$$

where $\boldsymbol{Q} = [q_{ij}] \in \Re^{m\times m}$ is a diagonal matrix. We can calculate the $i$th diagonal element $q_{ii}$ of the diagonal matrix $\boldsymbol{Q}$ as follows:

$$q_{ii} = \frac{1}{4\|\boldsymbol{P}_i\|_2^{3/2}}. \tag{26}$$

In order to avoid overflow, we add a small enough constant $\varepsilon$ into the definition of the matrix $\boldsymbol{Q}$, so the formula (26) can be rewritten as follows:

$$q_{ii} = \frac{1}{4\max(\|\boldsymbol{P}_i\|_2^{3/2}, \varepsilon)}. \tag{27}$$

**Table 1**
Procedure of SODNMF.

| | |
|---|---|
| Input: the dataset $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \Re^{m\times n}$, the clustering number $c$, the ratio of training samples *per*, the dual-graph parameter $\alpha$, the sparse parameter $\theta$, the bi-orthogonal parameter $\beta$, the neighbor number $k$, the bandwidth of heat kernel weighting $\sigma$ and the maximum iteration number *Niter*. Output: the coefficient matrix $\boldsymbol{P}$, the label auxiliary matrix $\boldsymbol{A}$, the label constraint matrix $\boldsymbol{C}$, the diagonal scaling matrix $\boldsymbol{R}$ and the clustering *label*. | |
| 1. | Normalize the data matrix $\boldsymbol{X}$. |
| 2. | Extract $c$ classes from the original data as the experimental data. |
| 3. | Pick up *per* percent from the experimental data as the available label information to construct the label constraint matrix $\boldsymbol{C}$. |
| 4. | Construct two nearest neighbor graphs in the feature space and the data space called feature graph and data graph, and calculate the similarity matrices of the feature graph and the data graph respectively. |
| 5. | Initialize the coefficient matrix $\boldsymbol{P}$, the label auxiliary matrix $\boldsymbol{A}$, the label constraint matrix $\boldsymbol{C}$, the diagonal scaling matrix $\boldsymbol{R}$ and the diagonal matrix $\boldsymbol{Q}$. |
| 6. | Update the coefficient matrix $\boldsymbol{P}$, the label auxiliary matrix $\boldsymbol{A}$, the label constraint matrix $\boldsymbol{C}$ and the diagonal scaling matrix $\boldsymbol{R}$ in the iterative updating rules (32)–(35). Update the diagonal matrix $\boldsymbol{Q}$ according to the coefficient matrix $\boldsymbol{P}$ of the moment by the formula (27). |
| 7. | Calculate the Lagrange function of the objective function by the formula (25). |
| 8. | Check the convergence of the iteration, if it is convergent, output the coefficient matrix $\boldsymbol{P}$, the label auxiliary matrix $\boldsymbol{A}$, the label constraint matrix $\boldsymbol{C}$ and the diagonal scaling matrix $\boldsymbol{R}$, otherwise, jump to step 6 and continue to update the iteration. |
| 9. | Use $k$-means to cluster the new representation and obtain the clustering *label*. |

In order to obtain the iterative updating rules of the coefficient matrix $\boldsymbol{P}$, the label auxiliary matrix $\boldsymbol{A}$, the label constraint matrix $\boldsymbol{C}$ and the diagonal scaling matrix $\boldsymbol{R}$, we should take the partial derivatives of $L$:

$$\frac{\partial L}{\partial \boldsymbol{P}} = -2\boldsymbol{X}\boldsymbol{C}\boldsymbol{A}\boldsymbol{R}^T + 2\boldsymbol{P}\boldsymbol{R}\boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A}\boldsymbol{R}^T + 2\alpha\boldsymbol{L}^P\boldsymbol{P} + 2\beta\boldsymbol{P} + 8\theta\boldsymbol{Q}\boldsymbol{P} \quad (28)$$

$$\frac{\partial L}{\partial \boldsymbol{A}} = -2\boldsymbol{C}^T\boldsymbol{X}^T\boldsymbol{P}\boldsymbol{R} + 2\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A}\boldsymbol{R}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{R} + 2\alpha\boldsymbol{C}^T\boldsymbol{L}^S\boldsymbol{C}\boldsymbol{A} + 2\beta\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A} \quad (29)$$

$$\frac{\partial L}{\partial \boldsymbol{C}} = -2\boldsymbol{X}^T\boldsymbol{P}\boldsymbol{R}\boldsymbol{A}^T + 2\boldsymbol{C}\boldsymbol{A}\boldsymbol{R}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{R}\boldsymbol{A}^T + 2\alpha\boldsymbol{L}^S\boldsymbol{C}\boldsymbol{A}\boldsymbol{A}^T + 2\beta\boldsymbol{C}\boldsymbol{A}\boldsymbol{A}^T \quad (30)$$

$$\frac{\partial L}{\partial \boldsymbol{R}} = -2\boldsymbol{P}^T\boldsymbol{X}\boldsymbol{C}\boldsymbol{A} + 2\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{R}\boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A}. \quad (31)$$

Considering the Karush–Kuhn–Tucker (KKT) conditions (Shang et al., 2016c), we can obtain the iterative updating rules of the coefficient matrix $\boldsymbol{P}$, the label auxiliary matrix $\boldsymbol{A}$, the label constraint matrix $\boldsymbol{C}$ and the diagonal scaling matrix $\boldsymbol{R}$ as follows:

$$p_{ij} \leftarrow p_{ij}\frac{(\boldsymbol{X}\boldsymbol{C}\boldsymbol{A}\boldsymbol{R}^T + \alpha\boldsymbol{W}^P\boldsymbol{P})_{ij}}{(\boldsymbol{P}\boldsymbol{R}\boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A}\boldsymbol{R}^T + \alpha\boldsymbol{D}^P\boldsymbol{P} + \beta\boldsymbol{P} + 4\theta\boldsymbol{Q}\boldsymbol{P})_{ij}} \quad (32)$$

$$a_{ij} \leftarrow a_{ij}\frac{(\boldsymbol{C}^T\boldsymbol{X}^T\boldsymbol{P}\boldsymbol{R} + \alpha\boldsymbol{C}^T\boldsymbol{W}^S\boldsymbol{C}\boldsymbol{A})_{ij}}{(\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A}\boldsymbol{R}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{R} + \alpha\boldsymbol{C}^T\boldsymbol{D}^S\boldsymbol{C}\boldsymbol{A} + \beta\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A})_{ij}} \quad (33)$$

$$c_{ij} \leftarrow c_{ij}\frac{(\boldsymbol{X}^T\boldsymbol{P}\boldsymbol{R}\boldsymbol{A}^T + \alpha\boldsymbol{W}^S\boldsymbol{C}\boldsymbol{A}\boldsymbol{A}^T)_{ij}}{(\boldsymbol{C}\boldsymbol{A}\boldsymbol{R}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{R}\boldsymbol{A}^T + \alpha\boldsymbol{D}^S\boldsymbol{C}\boldsymbol{A}\boldsymbol{A}^T + \beta\boldsymbol{C}\boldsymbol{A}\boldsymbol{A}^T)_{ij}} \quad (34)$$

$$r_{ij} \leftarrow r_{ij}\frac{(\boldsymbol{P}^T\boldsymbol{X}\boldsymbol{C}\boldsymbol{A})_{ij}}{(\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{R}\boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A})_{ij}}. \quad (35)$$

### 3.2. Procedure of SODNMF

Based on the above analysis, we can get the procedure of SODNMF as shown in Table 1.

### 3.3. Convergence analysis

In this section, we analyze the convergence of our algorithm and prove the objective function in the formula (23) decreases monotonically in the iterative updating rules (32)–(35).

Firstly, we analyze the convergence of the iterative updating rule (33).

**Definition 1.** If the following conditions are satisfied (Shang et al., 2016c):

$$M(x,a) \geq N(x) \text{ and } M(x,x) = N(x) \quad (36)$$

$M(x,a)$ is an auxiliary function for $N(x)$.

Assuming that for the $(t+1)$th generation of the updating rule is as follows:

$$x^{t+1} = \underset{x}{\operatorname{argmin}} M\left(x, x^t\right). \quad (37)$$

Obviously, it can prove $N\left(x^{t+1}\right) \leq M\left(x^{t+1}, x^t\right) \leq M\left(x^t, x^t\right) = N\left(x^t\right)$ and $N(x)$ is convergent.

**Lemma 1.**

$$M\left(a_{ij}, a_{ij}^t\right) = N_{ij}\left(a_{ij}^t\right) + N_{ij}'\left(a_{ij}^t\right)(a_{ij} - a_{ij}^t) + \frac{(\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A}\boldsymbol{R}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{R} + \alpha\boldsymbol{C}^T\boldsymbol{D}^S\boldsymbol{C}\boldsymbol{A} + \beta\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A})_{ij}}{a_{ij}^t}(a_{ij} - a_{ij}^t)^2 \quad (38)$$

*is the auxiliary function for* $N_{ij}(a_{ij})$*, where* $N(\boldsymbol{A}) = L(\boldsymbol{A})$.

**Proof.** The first-order derivative and second-order derivative on $N(\boldsymbol{A})$ are: $N_{ij}'(\boldsymbol{A}) = (-2\boldsymbol{C}^T\boldsymbol{X}^T\boldsymbol{P}\boldsymbol{R} + 2\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A}\boldsymbol{R}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{R} + 2\alpha\boldsymbol{C}^T\boldsymbol{L}^S\boldsymbol{C}\boldsymbol{A} + 2\beta\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A})_{ij}$, $N_{ij}''(\boldsymbol{A}) = 2(\boldsymbol{C}^T\boldsymbol{C})_{ii}(\boldsymbol{R}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{R})_{jj} + 2\alpha(\boldsymbol{C}^T\boldsymbol{L}^S\boldsymbol{C})_{ii} + 2\beta(\boldsymbol{C}^T\boldsymbol{C})_{ii}$, so the Taylor expansion of $N_{ij}(a_{ij})$ can be measure as follows:

$$N_{ij}(a_{ij}) = N_{ij}\left(a_{ij}^t\right) + N_{ij}'\left(a_{ij}^t\right)(a_{ij} - a_{ij}^t) + (\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{R}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{R} + \alpha\boldsymbol{C}^T(\boldsymbol{L}^S)^T\boldsymbol{C} + \beta\boldsymbol{C}^T\boldsymbol{C})_{ij}(a_{ij} - a_{ij}^t)^2. \quad (39)$$

Since

$$\begin{cases} (\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A}\boldsymbol{R}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{R})_{ij} = \sum_h\sum_k(\boldsymbol{C}^T\boldsymbol{C})_{ik}a_{kh}^t(\boldsymbol{R}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{R})_{hj} \\ \quad \geq (\boldsymbol{C}^T\boldsymbol{C})_{ii}(\boldsymbol{R}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{R})_{jj}a_{ij}^t \\ \alpha(\boldsymbol{C}^T\boldsymbol{D}^S\boldsymbol{C}\boldsymbol{A})_{ij} = \alpha\sum_k(\boldsymbol{C}^T\boldsymbol{D}^S\boldsymbol{C})_{ik}a_{kj}^t \\ \quad \geq \alpha(\boldsymbol{C}^T\boldsymbol{D}^S\boldsymbol{C})_{ii}a_{ij}^t \geq \alpha(\boldsymbol{C}^T\boldsymbol{L}^S\boldsymbol{C})_{ii}a_{ij}^t \\ \beta(\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A})_{ij} = \beta\sum_k(\boldsymbol{C}^T\boldsymbol{C})_{ik}a_{kj}^t \geq \beta(\boldsymbol{C}^T\boldsymbol{C})_{ii}a_{ij}^t, \end{cases}$$

we have $\frac{(\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A}\boldsymbol{R}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{R} + \alpha\boldsymbol{C}^T\boldsymbol{D}^S\boldsymbol{C}\boldsymbol{A} + \beta\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A})_{ij}}{a_{ij}^t} \geq (\boldsymbol{C}^T\boldsymbol{C})_{ii}(\boldsymbol{R}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{R})_{jj} + \alpha(\boldsymbol{C}^T\boldsymbol{L}^S\boldsymbol{C})_{ii} + \beta(\boldsymbol{C}^T\boldsymbol{C})_{ii}$, so that $M\left(a_{ij}, a_{ij}^t\right) \geq N_{ij}\left(a_{ij}\right)$.

According to the simultaneous equations (37) and (38), we know $M\left(a_{ij}^{t+1}, a_{ij}^t\right)$ is the local minimum of (38) and $a_{ij}^{t+1}$ is the corresponding local minimum point:

$$\begin{aligned} a_{ij}^{t+1} &= a_{ij}^t - \frac{a_{ij}^t N_{ij}'\left(a_{ij}^t\right)}{2(\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A}\boldsymbol{R}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{R} + \alpha\boldsymbol{C}^T\boldsymbol{D}^S\boldsymbol{C}\boldsymbol{A} + \beta\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A})_{ij}} \\ &= a_{ij}^t\frac{(\boldsymbol{C}^T\boldsymbol{X}^T\boldsymbol{P}\boldsymbol{R} + \alpha\boldsymbol{C}^T\boldsymbol{W}^S\boldsymbol{C}\boldsymbol{A})_{ij}}{(\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A}\boldsymbol{R}^T\boldsymbol{P}^T\boldsymbol{P}\boldsymbol{R} + \alpha\boldsymbol{C}^T\boldsymbol{D}^S\boldsymbol{C}\boldsymbol{A} + \beta\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A})_{ij}} \end{aligned} \quad (40)$$

$M\left(a_{ij}, a_{ij}^t\right)$ is the auxiliary function for $N_{ij}(a_{ij})$, so $N_{ij}$decreases monotonically by the formula (33).

In the same method, we can prove $N(\boldsymbol{C}) = L(\boldsymbol{C})$ and $N(\boldsymbol{R}) = L(\boldsymbol{R})$ decreases monotonically in the iterative updating rules (34) and (35).

Nextly, we analyze the convergence of the iterative updating rule (32).

**Lemma 2** (*Shi et al., 2015*).

$$\sum_{i=1}^m\left(\left\|\boldsymbol{g}_i^{t+1}\right\|_2^{1/2} - \frac{\left\|\boldsymbol{g}_i^{t+1}\right\|_2^2}{4\left\|\boldsymbol{g}_i^t\right\|_2^{3/2}}\right) \leq \sum_{i=1}^m\left(\left\|\boldsymbol{g}_i^t\right\|_2^{1/2} - \frac{\left\|\boldsymbol{g}_i^t\right\|_2^2}{4\left\|\boldsymbol{g}_i^t\right\|_2^{3/2}}\right). \quad (41)$$

**Proof.** From Lemma 2, we have:

$$\sum_{i=1}^m\left(\left\|\boldsymbol{P}_i^{t+1}\right\|_2^{1/2} - \frac{\left\|\boldsymbol{P}_i^{t+1}\right\|_2^2}{4\left\|\boldsymbol{P}_i^t\right\|_2^{3/2}}\right) \leq \sum_{i=1}^m\left(\left\|\boldsymbol{P}_i^t\right\|_2^{1/2} - \frac{\left\|\boldsymbol{P}_i^t\right\|_2^2}{4\left\|\boldsymbol{P}_i^t\right\|_2^{3/2}}\right). \quad (42)$$

In the $i$th generation, we fix $\boldsymbol{Q}$ as $\boldsymbol{Q}^t$ to solve $\boldsymbol{P}^{t+1}$; $\boldsymbol{A}^{t+1}$; $\boldsymbol{C}^{t+1}$ and $\boldsymbol{R}^{t+1}$. We define a function as follows:

$$\begin{aligned} H(\boldsymbol{P},\boldsymbol{A},\boldsymbol{C},\boldsymbol{R}) &= Tr(\boldsymbol{X}\boldsymbol{X}^T) - 2Tr(\boldsymbol{P}\boldsymbol{R}\boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{X}^T) \\ &\quad + Tr(\boldsymbol{P}\boldsymbol{R}\boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A}\boldsymbol{R}^T\boldsymbol{P}^T) + \alpha[Tr(\boldsymbol{P}^T\boldsymbol{L}^P\boldsymbol{P}) \\ &\quad + Tr(\boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{L}^S\boldsymbol{C}\boldsymbol{A})] + \beta[Tr(\boldsymbol{P}^T\boldsymbol{P}-\boldsymbol{I}) \\ &\quad + Tr(\boldsymbol{A}^T\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{A}-\boldsymbol{I})]. \end{aligned} \tag{43}$$

Since $\|\boldsymbol{P}\|_{2,1/2}^{1/2} = \sum_{i=1}^{m}\|\boldsymbol{P}_i\|_2^{1/2}$, we can obtain the following inequation:

$$\begin{aligned} H^{t+1} + \theta\sum_{i=1}^{m}\frac{\left\|\boldsymbol{P}_i^{t+1}\right\|_2^2}{4\left\|\boldsymbol{P}_i^t\right\|_2^{3/2}} &= H^{t+1} + \theta\|\boldsymbol{P}^{t+1}\|_{2,1/2}^{1/2} \\ &\quad + \theta\sum_{i=1}^{m}\left(\frac{\left\|\boldsymbol{P}_i^{t+1}\right\|_2^2}{4\left\|\boldsymbol{P}_i^t\right\|_2^{3/2}} - \left\|\boldsymbol{P}_i^{t+1}\right\|_2^{1/2}\right) \\ \le H^t + \theta\sum_{i=1}^{m}\frac{\left\|\boldsymbol{P}_i^t\right\|_2^2}{4\left\|\boldsymbol{P}_i^t\right\|_2^{3/2}} &= H^t + \theta\|\boldsymbol{P}^t\|_{2,1/2}^{1/2} \\ &\quad + \theta\sum_{i=1}^{m}\left(\frac{\left\|\boldsymbol{P}_i^t\right\|_2^2}{4\left\|\boldsymbol{P}_i^t\right\|_2^{3/2}} - \left\|\boldsymbol{P}_i^t\right\|_2^{1/2}\right). \end{aligned} \tag{44}$$

Combining (41) with (44), we can get the following inequation:

$$H^{t+1} + \theta\|\boldsymbol{P}^{t+1}\|_{2,1/2}^{1/2} \le H^t + \theta\|\boldsymbol{P}^t\|_{2,1/2}^{1/2}. \tag{45}$$

Therefore, the objective function in the formula (24) decreases monotonically in the iterative updating rule (32). In summary, based on the above convergence analysis, we can know the objective function (24) decreases monotonically in the iterative updating rules (32)–(35).

## 4. Experiments and analysis

There are many classified and clustering methods such as $k$-means, KNN, FCM, and SVM where the $v$-support vector classification has the advantage to control the number of support vectors and margin errors (Gu and Sheng, 2017) and the structural information of data is important for designing classifiers in real-world problems (Gu et al., 2015). In this section, we choose $k$-means as the clustering method to show the clustering experiments and the comparison of the clustering performance in SODNMF and other 4 algorithms on 3 datasets. The related algorithms are NMF (Zheng et al., 2007; Paatero and Tapper, 1994), GNMF (Cai et al., 2011), CNMF (Liu et al., 2012), DNMF (Babaee et al., 2016) and GSNMFC (Sun et al., 2016). A large number of experiments demonstrate the efficiency and effectiveness of our algorithm in image clustering.

### 4.1. Evaluation metrics

In the experiment, we take the number of the classes in the dataset as the clustering number. In general, the clustering performance can be shown by comparing the clustering labels with the ground truth labels. In this paper, two evaluation metrics are used to measure the clustering performance of the above clustering algorithms, which are clustering accuracy (ACC) (Liu et al., 2012; Cai et al., 2011) and normalized mutual information (NMI) (Liu et al., 2012; Cai et al., 2011).

**Clustering accuracy (ACC):** $\boldsymbol{x}_i$ is the given data point, $c_i$ is the clustering label, $g_i$ is the ground truth label. Clustering accuracy (ACC) can be defined as follows (Liu et al., 2012; Cai et al., 2011):

$$\mathrm{ACC} = \frac{\sum_{i=1}^{n}\delta(g_i, map(c_i))}{n} \tag{46}$$

where $n$ is the total number of the data, $\delta(\cdot)$ is delta function, $\delta(g_i, map(c_i))$ is 1 if $g_i = map(c_i)$ and 0 otherwise. $map(c_i)$ is the optimal mapping function obtained by Hungarian (Papadimitriou and Steiglitz, 1982), which maps the clustering label $c_i$ to the equivalent label from the dataset. The higher the clustering accuracy is, the better the clustering performance is.

**Normalized mutual information (NMI):** $\boldsymbol{C}$ is the clustering label set, $\boldsymbol{G}$ is the given ground truth set. Normalized mutual information can be defined as follows (Liu et al., 2012; Cai et al., 2011):

$$\mathrm{MI}(\boldsymbol{C},\boldsymbol{G}) = \sum_{c_i\in C, g_i\in G} p(c_i,g_i)\cdot\log_2\frac{p(c_i,g_i)}{p(c_i).p(g_i)} \tag{47}$$

where $p(c_i)$ is the probability that a random sample belongs to class $c_i$, $p(c_i, g_i)$ is the joint probability that a random sample belongs to class $c_i$ and class $g_i$ simultaneously. Then mutual information (MI) is normalized to normalized mutual information (NMI) as follows (Liu et al., 2012; Cai et al., 2011):

$$\mathrm{NMI}(\boldsymbol{C},\boldsymbol{G}) = \frac{\mathrm{MI}(\boldsymbol{C},\boldsymbol{G})}{\max(\mathrm{H}(\boldsymbol{C}),\mathrm{H}(\boldsymbol{G}))} \tag{48}$$

where $\mathrm{H}(\cdot)$ is the entropy function. NMI is 1 if two classes are exactly the same and 0 otherwise, so it is obvious that $\mathrm{NMI}\in[0,1]$. The bigger the NMI is, the higher the similarity of two classes is and the better the clustering performance is.

**Table 2**
Benchmark datasets.

| Dataset | Dimensionality | Size | Class |
|---|---|---|---|
| ORL | 1,024 | 400 | 40 |
| PIE | 1,024 | 2586 | 68 |
| TDT2 | 36,771 | 9394 | 30 |

### 4.2. Datasets

We do cluster experiments using the 5 algorithms (NMF, GNMF, CNMF, DNMF and GSNMFC) in Section 2 as well as our SODNMF on the 3 datasets in Table 2.

(1) ORL dataset (Liu et al., 2012):

The first dataset is AT&T ORL dataset which consists of 10 different images for each with 40 distinct subjects, thus 400 images in total. Each image contains 32 × 32 pixels with 256 gray levels per pixel. For some subjects, the images were taken at different lighting, times, facial expressions (smiling/not smiling, open/closed eyes), and facial details (glasses/no glasses). All the images were taken in a dark homogeneous background with the subjects in a frontal and upright position.

(2) PIE dataset (Babaee et al., 2016; Cai et al., 2011):

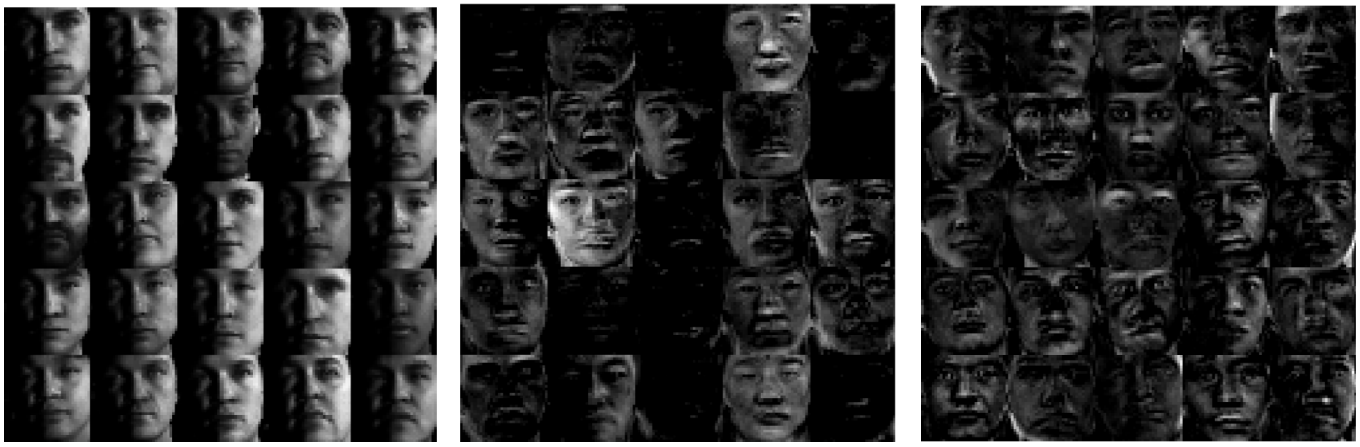
The second dataset is PIE_pose27 dataset which consists of 2586 images with 68 people. Each person has 42 facial images with 4 different expressions in different light and illumination conditions. Each image contains 32 × 32 pixels with 256 gray levels per pixel.

We pick up 42 different images for each with 25 distinct subjects, thus 1050 images in total. Fig. 1(a) shows the original 25 different images for each subject. Fig. 1(b) and (c) show the basic matrix learned by CNMF and SODNMF, brighter pixels indicate higher, where the clustering number $c$ is 25, the ratio of training samples $per$% is 10%.

Clustering accuracy (ACC) in CNMF is 52.57% and normalized mutual information (NMI) is 63.17%. Clustering accuracy (ACC) in SODNMF is 65.90% and normalized mutual information (NMI) is 78.34%.

(3) TDT2 dataset (Cai et al., 2011):

The third dataset is NIST Topic Detection and Tracking (TDT2) corpus. The data in TDT2 corpus was collected during the first half of 1998 and extracted from six sources including two television programs (ABC, CNN), two newswires (APW, NYT), and two radio programs (PRI, VOA). There are 11,201 on-topic documents classified into 96 categories in the TDT2 corpus. In this paper, the documents existing in two or more categories were deleted and only the largest 30 categories were reserved. Therefore, this TDT2 dataset consists of 9394 documents with 36,671 dimensions in total.



(a) original images. (b) basic matrix learned by CNMF. (c) basic matrix learned by SODNMF.

**Fig. 1.** Toy example on the PIE dataset.

### 4.3. Parameter analysis

We do clustering experiments measured by ACC and NMI on the three datasets in Table 2 in two unsupervised algorithms NMF, GNMF, three semi-supervised algorithms CNMF, DNMF, GSNMFC and our proposed SODNMF. The parameters for the 5 compared algorithms in Section 2 are stated in detail in Zheng et al. (2007), Liu et al. (2012), Babaee et al. (2016), Cai et al. (2011) and Sun et al. (2016). Here we mainly introduce the parameters for SODNMF.

From the Lagrange function of the objective function in the formula (23), we can see the main parameters for SODNMF are the dual-graph parameter $\alpha$, the bi-orthogonal parameter $\beta$ and the sparse parameter $\theta$. In addition, there are some common parameters, such as the clustering number $c$, the ratio of training samples *per*, the neighbor number $k$, the bandwidth of heat kernel weighting $\sigma$ and the maximum iteration number *Niter*. In the parameter sensitivity experiments, we fix the clustering number $c = 2$, the ratio of training samples $per\% = 10\%$ and select different $k$ and $\sigma$ to fix in different datasets, and then analyze the main parameters $\alpha$, $\beta$ and $\theta$ for SODNMF. We set $\alpha$ selected from {1, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000}, $\beta$ selected from $\{10^{-8}, 10^{-6}, 10^{-4}, 10^{-2}, 10^{0}, 10^{2}, 10^{4}, 10^{6}, 10^{8}\}$ and $\theta$ selected from {0.01, 0.05, 0.10, 0.50, 1.00, 5.00, 10.00}. Finally, for the sake of fairness, $k$-means is select to cluster the new representation in all the algorithms. The clustering results with the three main parameters $\alpha$, $\beta$ and $\theta$ for SODNMF are shown in Fig. 2.

We can see the clustering results of SODNMF from Fig. 2(a) and (b), the influence of the dual-graph parameter $\alpha$ on clustering accuracy is smaller than on normalized mutual information. From Fig. 2(c) and (d), we can see that SODNMF is very sensitive to the bi-orthogonal parameter $\beta$. For example, ACC and NMI with $\beta = 1000$ are significantly lower than other situations. Fig. 2(e) and (f) show that SODNMF is not sensitive to the sparse parameter $\theta$, ACC and NMI are stable with different values of $\theta$.

### 4.4. Convergence study

We obtain the local optimal values in the iterative updating rules. In Section 3.3, we have proved the objective function in the formula (23) decreases monotonically in the iterative updating rules. Here we show the convergence curve of unsupervised NMF, semi-supervised CNMF as well as our SODNMF on the three datasets ORL, PIE and TDT2 in Fig. 3. For each figure, the number of iterations is used as $X$-axis and the objective function value is used as $Y$-axis.

It can be seen from all the convergence curves, our algorithm converges faster than other algorithms. Other algorithms cannot converge when the number of iterations is less than 10, but our algorithm SODNMF can do it. On the ORL dataset, the number of iterations of NMF and CNMF are about 90 and 80 respectively, and the objective function tends to be stable. However, SODNMF can converge at 5 iterations. On the PIE data, the number of iterations of NMF and CNMF are about 50 and 40 respectively, and the objective function tends to be stable. However, SODNMF can converge at 8 iterations. On the massive high dimensional TDT2 dataset, NMF cannot converge within the first 40 iterations, and the downtrend of the objective function is still obvious. CNMF oscillates frequently within the first 20 iterations, and converges at about 40 iterations. On the contrary, SODNMF still shows the excellent convergence in TDT2 and converges at 3 iterations.

### 4.5. Clustering results and analysis

In order to make a comparison with different algorithms on the clustering performance, we extract the different $c$ classes from each dataset for clustering experiments. In dimensionality reduction algorithms, we usually set the new dimension equal to the clustering number $c$ and used $k$-means to cluster the new representation. Then we evaluated the clustering results of all algorithms by the evaluation metrics in Section 4.1. We randomly picked up 10% images from each category in the original data and use their category number as the available label information. For other parameters, cross-validation was performed on all algorithms and the parameter with the best results was selected for each dataset. In order to obtain convincing results, we carried out each experiment 10 times and calculated the mean as a result for each $c$. We randomly extracted $c$ classes from the dataset in every experiment.

Figs. 4–6 show the graph clustering results of the 6 algorithms (NMF, GNMF, CNMF, DNMF, GSNMFC, SODNMF) on the 3 datasets ORL, PIE and TDT2.

Fig. 4 shows clustering accuracy and normalized mutual information of 6 algorithms on the ORL dataset in vivid graphs. We can see that ACC and NMI curves of SODNMF are both above the curves of other algorithms in every clustering number, so SODNMF has the best clustering performance on the ORL dataset.

Similarly, Fig. 5 shows that ACC and NMI curves of SODNMF are also both above the curves of other algorithms in every clustering number, so SODNMF has the best clustering performance on the PIE dataset. In addition, it is obvious that clustering performances of semi-supervised CNMF, GSNMFC and SODNMF are superior to other algorithms especially NMF and GNMF.

As we can see from Fig. 6 on the TDT2 dataset, ACC and NMI curves of SODNMF are both above the curves of other algorithms in the most of clustering numbers especially NMI curves. From the clustering results,

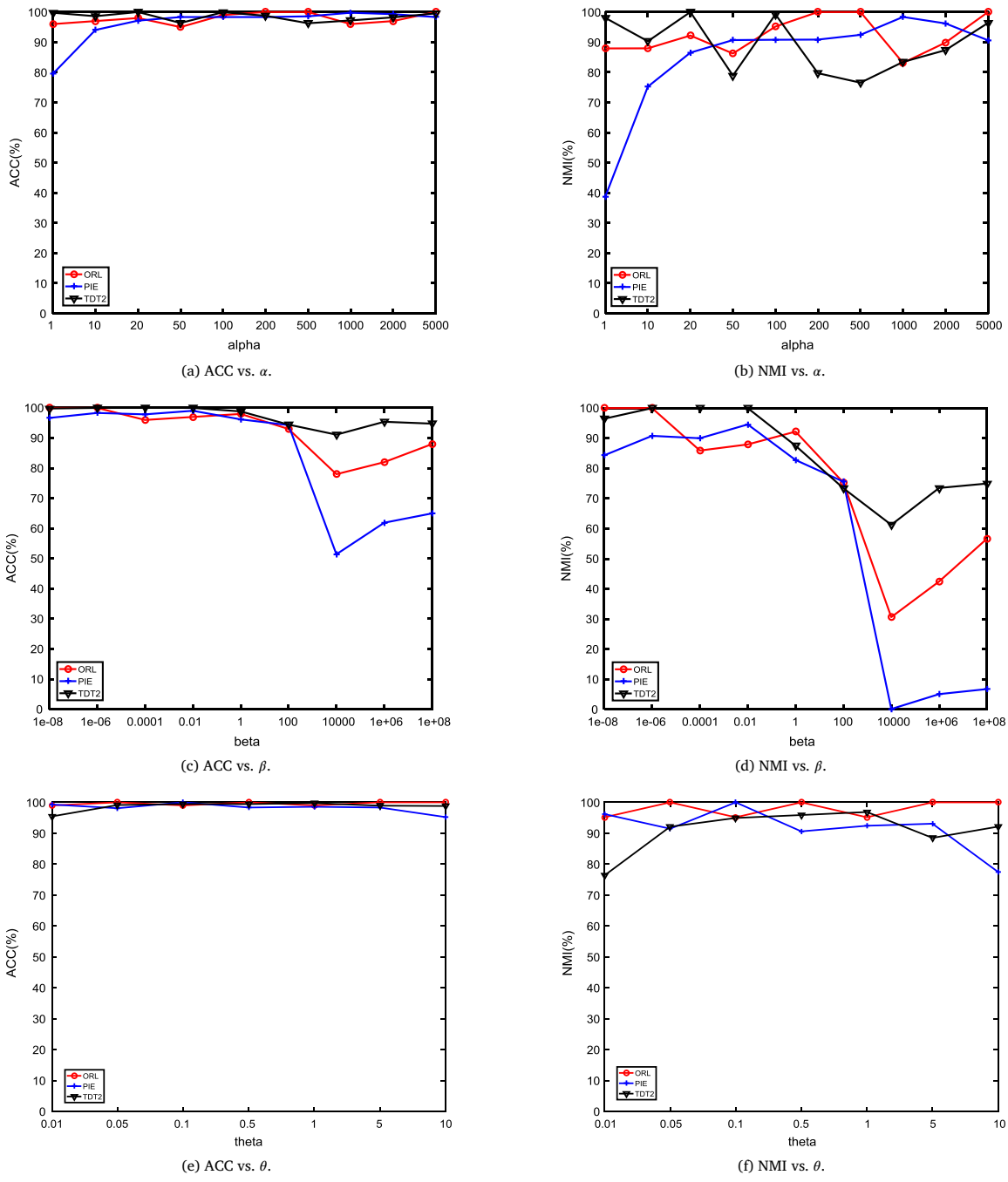


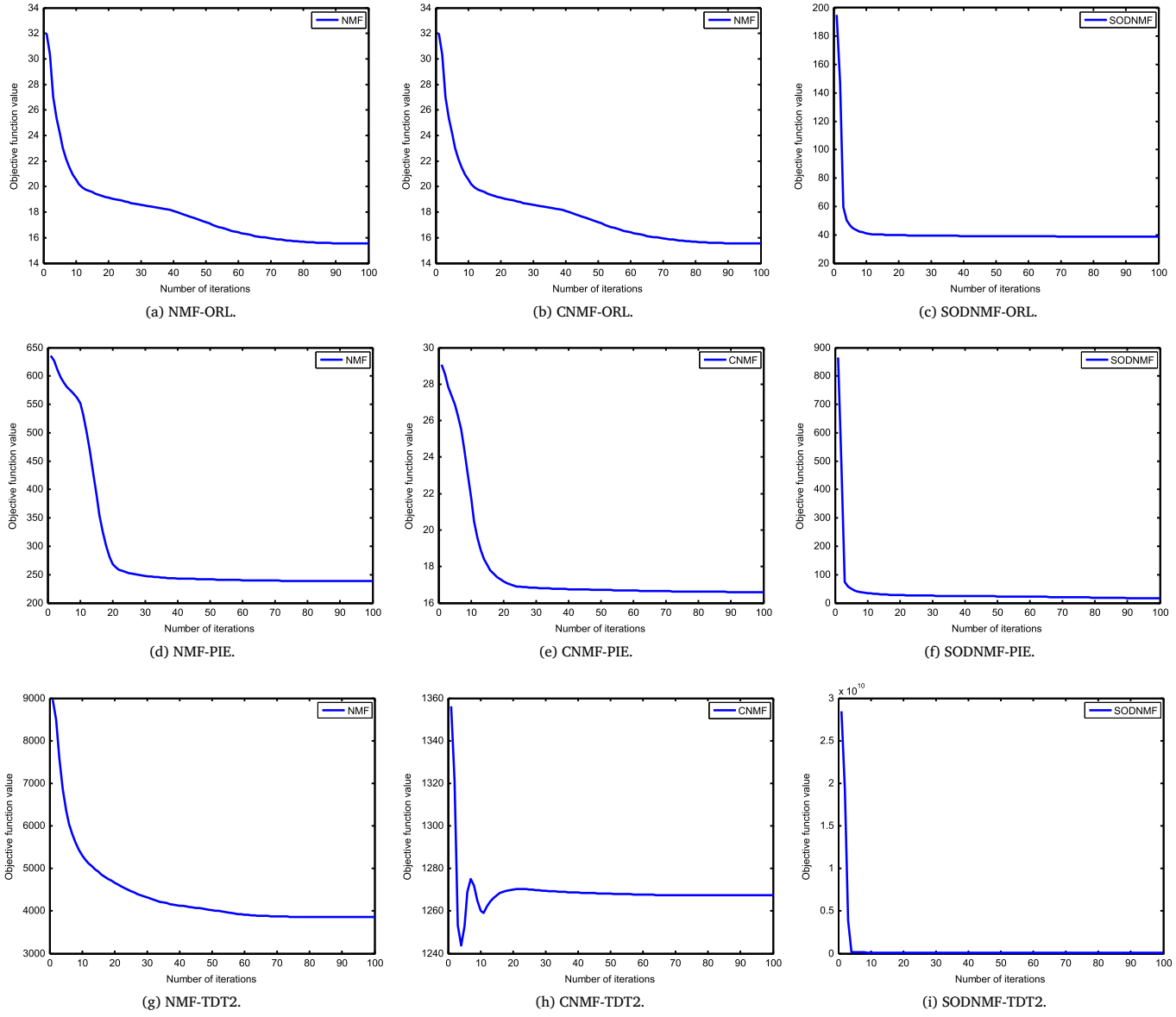

**Fig. 2.** Cluster performance of SODNMF vs. Parameters on three datasets.

we have reason to believe that the diagonal scaling matrix, dual-graph model and bi-orthogonal constraints used in SODNMF complement and cooperate with each other to enhance the learning ability and maintain the correlation between the data on the large scale datasets in different ways, which can improve the clustering performance and achieve good clustering results.

Tables 3–5 show the clustering results of the 6 algorithms (NMF, GNMF, CNMF, DNMF, GSNMFC and SODNMF) on 3 datasets ORL, PIE and TDT2 in detail. We bold mark the best result with each clustering number on each dataset.

Table 3 shows the clustering results of an experiment on the ORL dataset, where we can see SODNMF can achieve the best clustering performance with different clustering numbers. Compared with unsupervised algorithms, SODNMF is much better than NMF and GNMF. On clustering accuracy, SODNMF is 13.60% and 11.32% higher than unsupervised NMF and GNMF respectively on average. On normalized mutual information, SODNMF is 16.81% and 13.00% higher than unsupervised NMF and GNMF respectively on average. Compared with semi-supervised algorithms, SODNMF is also much better than CNMF, DNMF and GSNMFC. On clustering accuracy, SODNMF is 9.88%, 10.81% and 7.63% higher than semi-supervised CNMF, DNMF and GSNMFC respectively on average. On normalized mutual information, SODNMF is 11.92%, 10.58% and 8.15% higher than semi-supervised CNMF, DNMF and GSNMFC respectively on average.

Table 4 shows the clustering results of an experiment on the PIE dataset with a larger amount of data than the ORL dataset. We can see SODNMF can achieve the best clustering performance with different clustering numbers on the PIE dataset and have even greater advantages

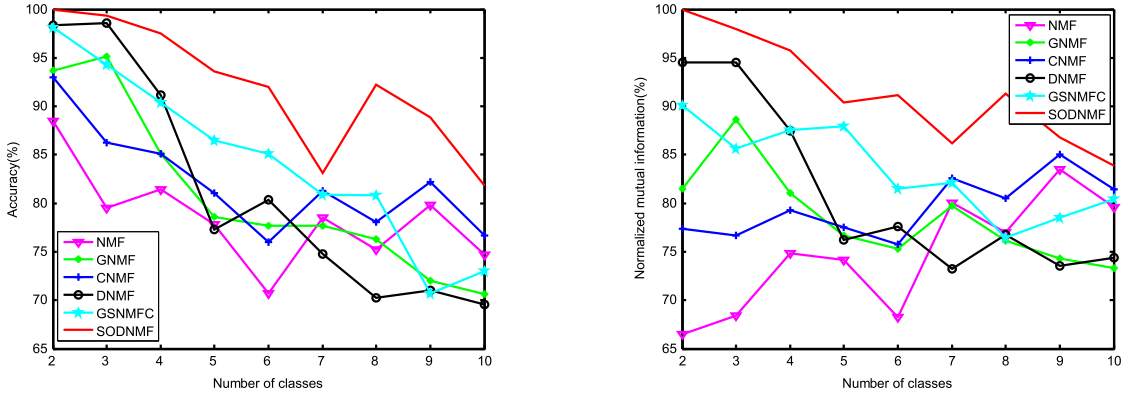



(a) NMF-ORL. (b) CNMF-ORL. (c) SODNMF-ORL.

(d) NMF-PIE. (e) CNMF-PIE. (f) SODNMF-PIE.

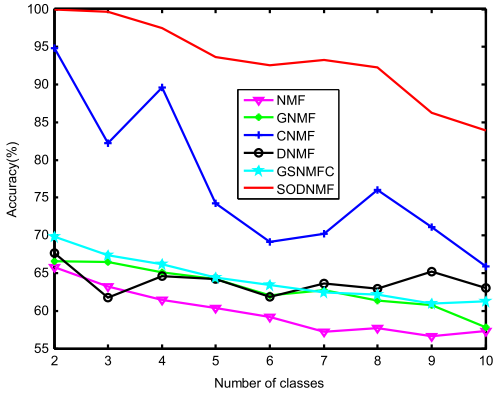
(g) NMF-TDT2. (h) CNMF-TDT2. (i) SODNMF-TDT2.

**Fig. 3.** Convergence curves of NMF, CNMF and SODNMF on three datasets.

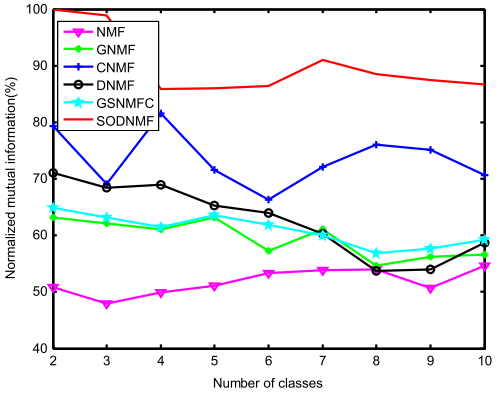



(a) Accuracy vs. number of classes. (b) Normalized mutual information vs. number of classes.

**Fig. 4.** Clustering performance on the ORL dataset.

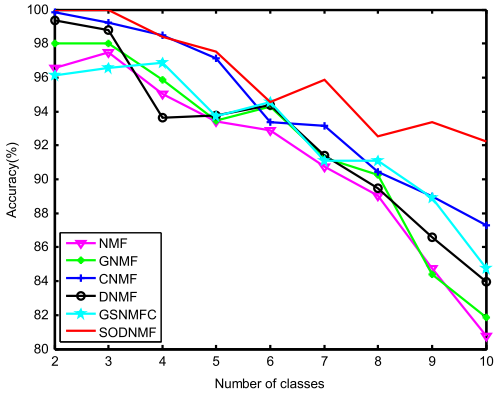



(a) Accuracy vs. number of classes.

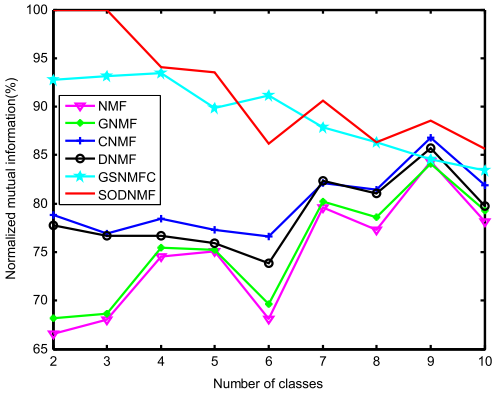



(b) Normalized mutual information vs. number of classes.

**Fig. 5.** Clustering performance on the PIE dataset.




(a) Accuracy vs. number of classes.




(b) Normalized mutual information vs. number of classes.

**Fig. 6.** Clustering performance on the TDT2 dataset.

**Table 3**
Clustering results of five algorithms on the ORL dataset.

| $c$ | Accuracy (%) | | | | | | Normalized mutual information (%) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NMF | GNMF | CNMF | DNMF | GSNMFC | SODNMF | NMF | GNMF | CNMF | DNMF | GSNMFC | SODNMF |
| 2 | 88.49 | 93.67 | 93.01 | 98.32 | 98.15 | **100.00** | 66.49 | 81.51 | 77.35 | 94.49 | 90.05 | **100.00** |
| 3 | 79.53 | 95.10 | 86.26 | 98.59 | 94.33 | **99.33** | 68.40 | 88.66 | 76.67 | 94.49 | 85.61 | **97.97** |
| 4 | 81.45 | 85.08 | 85.11 | 91.12 | 90.37 | **97.50** | 74.83 | 81.01 | 79.25 | 87.49 | 87.52 | **95.77** |
| 5 | 77.79 | 78.57 | 81.03 | 77.26 | 86.50 | **93.60** | 74.12 | 76.67 | 77.56 | 76.21 | 87.93 | **90.39** |
| 6 | 70.67 | 77.68 | 75.98 | 80.33 | 85.09 | **92.00** | 68.25 | 75.27 | 75.78 | 77.61 | 81.50 | **91.19** |
| 7 | 78.52 | 77.65 | 81.28 | 74.77 | 80.90 | **83.14** | 80.05 | 79.72 | 82.59 | 73.27 | 82.15 | **86.21** |
| 8 | 75.20 | 76.26 | 78.02 | 70.28 | 80.78 | **92.25** | 76.97 | 76.11 | 80.51 | 76.74 | 76.42 | **91.33** |
| 9 | 79.81 | 72.03 | 82.21 | 70.98 | 70.67 | **88.89** | 83.49 | 74.27 | 85.05 | 73.54 | 78.55 | **86.77** |
| 10 | 74.66 | 70.65 | 76.70 | 69.58 | 73.04 | **81.80** | 79.62 | 73.31 | 81.44 | 74.42 | 80.45 | **83.89** |
| Avg. | 78.46 | 80.74 | 82.18 | 81.25 | 84.43 | **92.06** | 74.69 | 78.50 | 79.58 | 80.92 | 83.35 | **91.50** |

**Table 4**
Clustering results of five algorithms on the PIEdataset.

| $c$ | Accuracy (%) | | | | | | Normalized mutual information (%) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NMF | GNMF | CNMF | DNMF | GSNMFC | SODNMF | NMF | GNMF | CNMF | DNMF | GSNMFC | SODNMF |
| 2 | 65.81 | 66.58 | 94.80 | 67.64 | 69.84 | **100.00** | 50.84 | 63.16 | 79.36 | 71.01 | 64.84 | **100.00** |
| 3 | 63.22 | 66.48 | 82.19 | 61.75 | 67.34 | **99.68** | 47.90 | 62.06 | 69.06 | 68.44 | 63.18 | **98.82** |
| 4 | 61.49 | 65.09 | 89.60 | 64.56 | 66.14 | **97.50** | 49.86 | 61.10 | 81.53 | 68.96 | 61.42 | **85.95** |
| 5 | 60.41 | 64.16 | 74.26 | 64.24 | 64.40 | **93.60** | 51.07 | 63.22 | 71.58 | 65.23 | 63.63 | **85.96** |
| 6 | 59.23 | 62.01 | 69.15 | 61.82 | 63.44 | **92.54** | 53.33 | 57.30 | 66.37 | 63.95 | 61.92 | **86.35** |
| 7 | 57.19 | 62.74 | 70.22 | 63.57 | 62.40 | **93.27** | 53.80 | 61.09 | 72.06 | 60.32 | 60.04 | **91.06** |
| 8 | 57.70 | 61.39 | 76.03 | 62.91 | 62.13 | **92.25** | 53.98 | 54.65 | 76.00 | 53.71 | 56.90 | **88.50** |
| 9 | 56.63 | 60.75 | 71.07 | 65.22 | 60.96 | **86.24** | 50.66 | 56.20 | 75.15 | 53.99 | 57.63 | **87.46** |
| 10 | 57.28 | 57.85 | 65.87 | 62.99 | 61.21 | **83.89** | 54.60 | 56.63 | 70.64 | 58.73 | 59.19 | **86.62** |
| Avg. | 59.88 | 63.01 | 77.02 | 63.86 | 64.21 | **93.22** | 51.78 | 59.49 | 73.53 | 62.70 | 60.97 | **90.08** |

**Table 5**
Clustering results of five algorithms on the TDT2 dataset.

| $c$ | Accuracy (%) | | | | | | Normalized MutualInformation (%) | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NMF | GNMF | CNMF | DNMF | GSNMFC | SODNMF | NMF | GNMF | CNMF | DNMF | GSNMFC | SODNMF |
| 2 | 96.55 | 98.02 | 99.84 | 99.37 | 96.12 | **100.00** | 66.59 | 68.16 | 78.83 | 77.74 | 92.80 | **100.00** |
| 3 | 97.50 | 97.99 | 99.23 | 98.78 | 96.55 | **100.00** | 68.03 | 68.62 | 76.90 | 76.66 | 93.17 | **100.00** |
| 4 | 95.03 | 95.88 | **98.48** | 93.64 | 96.87 | 98.40 | 74.51 | 75.46 | 78.45 | 76.65 | 93.46 | **94.07** |
| 5 | 93.42 | 93.46 | 97.14 | 93.76 | 93.71 | **97.51** | 75.10 | 75.24 | 77.31 | 75.95 | 89.89 | **93.51** |
| 6 | 92.89 | 94.27 | 93.39 | 94.39 | 94.53 | **94.54** | 68.12 | 69.60 | 76.59 | 73.82 | 91.13 | **86.19** |
| 7 | 90.75 | 91.28 | 93.15 | 91.41 | 91.09 | **95.88** | 79.56 | 80.19 | 82.15 | 82.31 | 87.83 | **90.62** |
| 8 | 89.02 | 90.27 | 90.43 | 89.47 | 91.10 | **92.55** | 77.27 | 78.62 | 81.45 | 81.02 | 86.31 | **86.30** |
| 9 | 84.77 | 84.41 | 88.99 | 86.58 | 88.89 | **93.38** | 84.35 | 84.09 | 86.78 | 85.70 | 84.57 | **88.53** |
| 10 | 80.78 | 81.88 | 87.30 | 83.96 | 84.74 | **92.23** | 78.12 | 79.32 | 81.87 | 79.72 | 83.44 | **85.62** |
| Avg. | 91.19 | 91.94 | 94.22 | 92.37 | 92.62 | **95.94** | 74.63 | 75.48 | 80.04 | 78.84 | 89.18 | **91.65** |

in comparison with ORL datasets with the smaller amount of data. Compared with unsupervised algorithms, SODNMF is much better than NMF and GNMF. On clustering accuracy, SODNMF is 33.34% and 30.21% higher than unsupervised NMF and GNMF respectively on average. On normalized mutual information, SODNMF is 38.30% and 30.59% higher than unsupervised NMF and GNMF respectively on average. Compared with semi-supervised algorithms, SODNMF is also much better than CNMF, DNMF and GSNMFC. On clustering accuracy, SODNMF is 16.20%, 29.36% and 29.01% higher than semi-supervised CNMF, DNMF and GSNMFC respectively on average. On normalized mutual information, SODNMF is 16.55%, 27.38% and 29.11% higher than semi-supervised CNMF, DNMF and GSNMFC respectively on average.

Table 5 shows the clustering results of an experiment on the TDT2 dataset with high dimensional and a larger amount of data, where we can see SODNMF can achieve the best clustering performance with most of the clustering numbers except that the clustering number is 2. Compared with unsupervised algorithms, SODNMF is much better than NMF and GNMF. On clustering accuracy, SODNMF is 4.75% and 4.00% higher than unsupervised NMF and GNMF respectively on average. On normalized mutual information, SODNMF is 17.02% and 16.17% higher than unsupervised NMF and GNMF respectively on average. Compared with semi-supervised algorithms, SODNMF is also much better than CNMF, DNMF and GSNMFC. On clustering accuracy, SODNMF is 1.72%, 3.57% and 3.32% higher than semi-supervised CNMF, DNMF and GSNMFC respectively on average. On normalized mutual information, SODNMF is 11.61%, 12.81% and 2.47% higher than semi-supervised CNMF, DNMF and GSNMFC respectively on average.

Therefore, it is obvious that SODNMF is superior to others algorithms on ACC and NMI, and have a good clustering performance. We have reason to believe that dual-graph model, bi-orthogonal constraints, diagonal scaling matrix and sparse constraint used in SODNMF can enhance the learning ability and improve the clustering performance in different ways.

### 4.6. Different variants of SODNMF

To explain the influence of dual-graph model, bi-orthogonal constraints, diagonal scaling matrix and sparse constraint used in SODNMF respectively, four different variants of SODNMF are tested on the overall performance. These variants of SODNMF are summarized as follows: "1" denotes the influence of dual-graph model, we set the dual-graph parameter $\alpha = 0$; "2" denotes the influence of bi-orthogonal constraints, we set the bi-orthogonal parameter $\beta = 0$; "3" denotes the influence of the scaling adjustment in the optimization process, which is the variant without the diagonal scaling matrix; "4" denotes the influence of sparse constraint, we set the sparse parameter $\theta = 0$.

In order to obtain convincing results, we carried out each experiment 10 times and calculated the mean as a result when the clustering number $c = 10$. Table 6 shows the clustering results of SODNMF and some variants on 3 datasets ORL, PIE and TDT2 in detail. We bold mark the best result on each dataset.

From Table 6, we can see that SODNMF with dual-graph model, bi-orthogonal constraints, diagonal scaling matrix and sparse constraint can achieve the best clustering results. This illustrates the importance of dual-graph model, bi-orthogonal constraints, diagonal scaling matrix and sparse constraint in SODNMF. Overall, the performance of "1" which is the variant without dual-graph model is the worst. Compared with this variant, SODNMF is 54.38% and 57.69% higher than this variant on clustering accuracy and normalized mutual information respectively on average. It means that the dual-graph model plays a very important role in SODNMF, which can make full use of the potential structural information on account of preserving the manifold structures of the data space and the feature space simultaneously.

When the bi-orthogonal parameter $\beta = 0$ in "2", this variant is SODNMF without the bi-orthogonal constraints. Compared with this variant, SODNMF is 4.29% and 2.01% higher than this variant on clustering accuracy and normalized mutual information respectively on average. It means that the bi-orthogonal constraints have also contributed to the cluster, which can avoid the non-correspondence between images and basic vectors so that it can improve the clustering performance.

"3" is the variant of SODNMF without the diagonal scaling matrix. Compared with this variant, SODNMF is 6.96% and 7.17% higher than this variant on clustering accuracy and normalized mutual information respectively on average. It means that the diagonal scaling matrix plays an important role in SODNMF, which can avoid inducing an unreliable solution due to scaling issue in several low dimensional matrices, so the clustering performance of SODNMF is better than "3".

When the bi-orthogonal parameter $\theta = 0$ in "4", this variant is SODNMF without the sparse constraint. Compared with this variant, SODNMF is 6.97% and 5.01% higher than this variant on clustering accuracy and normalized mutual information respectively on average. In addition, we compared the run time of SODNMF with "4" on ORL, PIE and TDT2, SODNMF was 2.39 s, 0.02 s and 5159.53 s faster than "4". Therefore, we can know that the sparse constraint in SODNMF can not only enhance the local learning ability and improve the clustering performance, but also simplify the calculation and accelerate the processing speed.

## 5. Conclusions

In this paper, we propose a novel semi-supervised non-negative NMF called Dual-graph regularized Non-negative Matrix Factorization with Sparse and Orthogonal constraints (SODNMF) which can not only utilize a fraction of label information, but also effectively learn local information of the objectives, such as documents and faces. Dual-graph model is added into SODNMF, which can make full use of the potential structural information on account of preserving the manifold structures of the data space and the feature space simultaneously. In addition, sparse constraint is incorporated as the additional condition in SODNMF, which can make the coefficient matrix with a good sparsity and simplify the calculation. Last but not least, bi-orthogonal constraints are adopted in SODNMF. Each image can correspond to the unique basic vector, which can effectively enhance the discrimination of clustering and the exclusivity across the classes, and improve the clustering performance. From all the experimental results above, we can make a conclusion that

**Table 6**
Clustering results of SODNMF and some variants.

| Dataset | Accuracy (%) | | | | | Normalized mutual information (%) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | "1" | "2" | "3" | "4" | SODNMF | "1" | "2" | "3" | "4" | SODNMF |
| ORL | 34.80 | 72.40 | 72.04 | 71.20 | **81.80** | 36.03 | 76.59 | 76.13 | 75.65 | **83.89** |
| PIE | 18.57 | 81.03 | 77.65 | 78.57 | **83.89** | 6.54 | 85.26 | 77.13 | 84.09 | **86.62** |
| TDT2 | 41.40 | 91.60 | 87.34 | 87.24 | **92.23** | 40.49 | 88.26 | 81.35 | 81.36 | **85.62** |
| Avg. | 31.59 | 81.68 | 79.01 | 79.00 | **85.97** | 27.69 | 83.37 | 78.21 | 80.37 | **85.38** |

our proposed algorithm SODNMF is with encouraging performance on both clustering accuracy and normalized mutual information. SODNMF can effectively reduce the dimensionality of the high dimensional data and deal with big data problems, but it should update 4 matrices in each iteration and the processing time is long. In order to solve this problem, we will reduce the number of new matrices in each iteration and further improve the clustering performance of SODNMF.

## Acknowledgments

This work was partially supported by the National Natural Science Foundation of China, under Grants 61371201, 61773304, 61771376 and 61772399, the National Basic Research Program (973 Program) of China under Grant 2013CB329402, the Project Supported by Natural Science Basic Research in Shaanxi Province of China (Program No. 2014JM2-1006).

## Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.engappai.2017.11.008.

## References

Babaee, M., Tsoukalas, S., Babaee, M., et al., 2016. Discriminative non-negative matrix factorization for dimensionality reduction. Neurocomputing 173, 212–223.
Belkin, M., Niyogi, P., Sindhwani, V., 2006. Manifold regularization: A geometric framework for learning from examples. J. Mach. Learn. Res. 7, 2399–2434.
Cai, D., He, X., Han, J., et al., 2011. Graph regularized non-negative matrix factorization for data representation. IEEE Trans. Pattern Anal. Mach. Intell. 33 (8), 1548–1560.
Cai, D., He, X., Wang, X., et al., 2009a. Locality preserving nonnegative matrix factorization. In: Proceedings of the International Joint Conference on Artificial Intelligence. pp. 1010–1015.
Cai, D., Wang, X., He, X., et al., 2009b. Probabilistic dyadic data analysis with local and global consistency. In: International Conference on Machine Learning. pp. 105–112.
Ding, C., Li, T., Peng, W., et al., 2006. Orthogonal nonnegative matrix tri-factorizations for clustering. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 126–135.
Duda, R., Hart, P., Stork, D., 2012. Pattern Classification. Wiley.
Feng, X., Jiao, Y., Lv, C., et al., 2016. Label consistent semi-supervised non-negative matrix factorization for maintenance activities identification. Eng. Appl. Artif. Intell. 52, 161–167.
Gu, B., Sheng, V.S., 2017. A robust regularization path algorithm for $\nu$-support vector classification. IEEE Trans. Neural Netw. Learn. Syst. 28 (5), 1241–1248.
Gu, B., Sheng, V.S., Tay, K.Y., et al., 2015. Incremental support vector learning for ordinal regression. IEEE Trans. Neural Netw. Learn. Syst. 26 (7), 1403–1416.
Gu, B., Sun, X., Sheng, V.S., 2017. Structural minimax probability machine. IEEE Trans. Neural Netw. Learn. Syst. 28 (7), 1646–1656.
Gu, Q., Zhou, J., 2009. Co-clustering on manifolds. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 359–368.
Hong, S., Choi, J., Feyereisl, J., et al., 2016. Joint image clustering and labeling by matrix factorization. IEEE Trans. Pattern Anal. Mach. Intell. 38 (7), 1411–1424.
Jolliffe, I., 1986. Principal Component Analysis, Vol. 87. Springer, Berlin, pp. 41–64.
Lee, D., Seung, H., 1999. Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791.
Li, S., Hou, X., Zhang, H., et al., 2001. Learning spatially localized, parts-based representation. Comput. Vis. Pattern Recognit. 207–212.
Li, Z., Yang, Y., Liu, J., et al., 2012. Unsupervised feature selection using nonnegative spectral analysis. In: National Conference on Artificial Intelligence. pp. 1026–1032.
Liu, H., Wu, Z., Li, X., et al., 2012. Constrained nonnegative matrix factorization for image representation. IEEE Trans. Pattern Anal. Mach. Intell. 34 (7), 1299–1311.
Luo, M., Zhang, K., 2014. A hybrid approach combining extreme learning machine and sparse representation for image classification. Eng. Appl. Artif. Intell. 27, 228–235.
Ma, Z., Nie, F., Yang, Y., et al., 2012. Discriminating joint feature analysis for multimedia data understanding. IEEE Trans. Multimedia 14 (6), 1662–1672.
Ma, T., Wang, Y., Tang, M., et al., 2016. LED: A fast overlapping communities detection algorithm based on structural clustering. Neurocomputing 207, 488–500.
Nie, F., Huang, H., Cai, X., et al., 2010. Efficient and robust feature selection via joint $L_{2,1}$-norms minimization. In: Advances in Neural Information Processing Systems. pp. 1813–1821.
Paatero, P., Tapper, U., 1994. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. Environmetrics 5 (2), 111–126.
Pang, Y., Wang, S., Yuan, Y., 2014. Learning regularized LDA by clustering. IEEE Trans. Neural Netw. Learn. Syst. 25 (12), 2191–2201.
Papadimitriou, C., Steiglitz, K., 1982. Combinatorial Optimization: Algorithms and Complexity. Prentice Hall.
Shahnaz, F., Berry, M., Pauca, V., et al., 2006. Document clustering using nonnegative matrix factorization. Inf. Process. Manage. 42, 373–386.
Shang, R., Wang, W., Stolkin, R., et al., 2016a. Subspace learning-based graph regularized feature selection. Knowl.-Based Syst. 112, 152–165.
Shang, R., Wang, W., Stolkin, R., et al., 2017. Non-negative spectral learning and sparse regression-based dual-graph regularized feature selection. IEEE Trans. Cybern. http://dx.doi.org/10.1109/TCYB.2017.2657007.
Shang, R., Zhang, Z., Jiao, L., et al., 2016b. Global discriminative-based nonnegative spectral clustering. Pattern Recognit. 55, 172–182.
Shang, R., Zhang, Z., Jiao, L., et al., 2016c. Self-representation based dual-graph regularized feature selection clustering. Neurocomputing 171, 1242–1253.
Shi, C., Ruan, Q., An, G., et al., 2015. Hessian semi-supervised sparse feature selection based on $L_{2,1/2}$-matrix norm.. IEEE Trans. Multimedia 17 (1), 16–28.
Sindhwani, V., Hu, J., Mojsilovic, A., 2009. Regularized co-clustering with dual supervision. Adv. Neural Inf. Process. Syst. 1505–1512.
Sun, F., Xu, M., Hu, X., et al., 2016. Graph regularized and sparse nonnegative matrix factorization with hard constraints for data representation. Neurocomputing 173, 233–244.
Tian, Q., Chen, S., 2017. Cross-heterogeneous-database age estimation through correlation representation learning. Neurocomputing 238, 286–295.
Xu, Z., Chang, X., Xu, F., et al., 2012. $L_{1/2}$ regularization: A thresholding representation theory and a fast solver. IEEE Trans. Neural Netw. Learn. Syst. 23 (7), 1013–1027.
Xu, W., Liu, X., Gong, Y., 2003. Document clustering based on non-negative matrix factorization. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 267–273.
Xu, Z., Zhang, H., Wang, Y., et al., 2010. $L_{1/2}$ regularization. Sci. China Inf. Sci. 53 (6), 1159–1169.
Yang, L., Wang, L., Gao, Y., et al., 2014. A convex relaxation framework for a class of semi-supervised learning methods and its application in pattern recognition. Eng. Appl. Artif. Intell. 35, 335–344.
Zheng, Z., Yang, J., Zhu, Y., 2007. Initialization enhancer for non-negative matrix factorization. Eng. Appl. Artif. Intell. 20 (1), 101–110.