Information Sciences 585 (2022) 571–593

Contents lists available at ScienceDirect

Information Sciences

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

Dual semi-supervised convex nonnegative matrix factorization for data representation

Siyuan Peng [a], Zhijing Yang [a], Bingo Wing-Kuen Ling [a], Badong Chen [b], Zhiping Lin [c,*]

[a] School of Information Engineering, Guangdong University of Technology, Guangzhou 510006, China
[b] Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China
[c] School of Electrical and Electronic Engineering, Nanyang Technological University, 639798, Singapore

ARTICLE INFO

Article history:
Received 19 April 2021
Received in revised form 9 October 2021
Accepted 14 November 2021
Available online 26 November 2021

Keywords:
Semi-supervised learning
Convex nonnegative matrix factorization
data representation
clustering

ABSTRACT

Semi-supervised nonnegative matrix factorization (NMF) has received considerable attention in machine learning and data mining. A new semi-supervised NMF method, called dual semi-supervised convex nonnegative matrix factorization (DCNMF), is proposed in this paper for fully using the limited label information. Specifically, DCNMF simultaneously incorporates the pointwise and pairwise constraints of labeled samples as dual supervisory information into convex NMF, which results in a better low-dimensional data representation. Moreover, DCNMF imposes the nonnegative constraint only on the coefficient matrix but not on the base matrix. Consequently, DCNMF can process mixed-sign data, and hence enlarge the range of applications. We derive an efficient alternating iterative algorithm for DCNMF to solve the optimization, and analyze the proposed DCNMF method in terms of the convergence and computational complexity. We also discuss the relationships between DCNMF and several typical NMF based methods. Experimental results illustrate that DCNMF outperforms the related state-of-the-art NMF methods on nonnegative and mixed-sign datasets for clustering applications.

© 2021 Elsevier Inc. All rights reserved.

1. Introduction

Due to the curse of dimensionality, data representation plays a significant role in the fields of machine learning and data mining, which aims to discover the latent low-dimensional information for representing high-dimensional data [1]. A good data representation can obviously enhance the performance of learning algorithms. In recent decades, matrix factorization techniques, such as singular value decomposition [2], principal component analysis (PCA) [3,4], independent component analysis [5], nonnegative matrix factorization (NMF) [6,7], have been widely used to high-dimensional datasets for obtaining efficient data representations. Since NMF has clear physical meaning and strong theoretical interpretations, it has been successfully adopted in numerous applications, e.g., image processing [8,9] and document clustering [10,11]. As one of the most commonly used matrix factorization techniques, NMF is also considered as an important unsupervised learning technique for dimensional reduction and data representation. Furthermore, NMF has close relationships with spectral clustering and the k-means algorithm [12]. Therefore the overwhelming interest of most NMF based methods mainly focus on the real-world clustering applications, particularly for image and text datasets [13].

* Corresponding author.
E-mail address: EZPLin@ntu.edu.sg (Z. Lin).

https://doi.org/10.1016/j.ins.2021.11.045
0020-0255/© 2021 Elsevier Inc. All rights reserved.

NMF aims to find two low-dimensional nonnegative matrices such that the product can provide an excellent approximation to the data matrix. Since the learned matrices are enforced to be nonnegative that only allow the additive operations, without any subtractive combinations, NMF usually results in the parts-based representation [13]. Although the original NMF has good properties, it still suffers from some limitations: 1) it ignores the geometric structure of data; 2) it is only used for nonnegative data which limits its applicable range; 3) it fails to utilize any supervised information. To overcome those shortcomings, many improved variants of NMF have been proposed in the recent decade. For instance, to discover the geometrical information of data, some NMF with graph regularization methods have been developed by incorporating the graph embedding framework into NMF [14,15]. In order to enlarge the applicable range of the NMF, convex NMF and semi-NMF methods have also been proposed for processing the mixed-sign data [16–18].

The aforementioned NMF based methods are the unsupervised learning algorithms, without considering any supervisory information. However, in practical tasks, usually a small amount of supervisory information can be available from domain experts. Especially, using limited supervised information can improve the learning performance of algorithms. Therefore, in the past decade, some semi-supervised NMF algorithms have been proposed [19–24], and the most representative methods are the constrained NMF (CNMF) [19] and the semi-supervised NMF via constraint propagation (CPSNMF) [20]. CNMF directly incorporates the pointwise constraints of labeled data points into NMF as the hard supervision, which guarantees the data points with same label will have the same coordinate in the low-dimensional representation. The main limitations of CNMF are that it does not make full use of the unlabel data and ignores the intrinsic geometrical information of data. CPSNMF can overcome the limitations of CNMF. By using the constraint propagation algorithm (CPA), CPSNMF first propagates the pairwise constraints to the unlabeled data samples, and then incorporates the total pairwise constraints into p-nearest neighbour to reconstruct the weight matrix of data graph. However, CPSNMF adopts the pairwise constraints to guide the construction of the data weight matrix, which is difficult to ensure the learned lower data representations with same (or different) labels to be close (or far) enough. Furthermore, when a small amount of supervised information is obtained, the improvement will be very limited for the semi-supervised NMF methods. Recently, the robust correntropy based semi-supervised NMF (CSNMF) has been developed in [23], which utilizes two type of semi-supervised information simultaneously. However, CSNMF requires a lot of computing time in practical tasks, and cannot deal with mixed-sign data. In this paper, a novel semi-supervised NMF method, namely dual semi-supervised convex nonnegative matrix factorization (DCNMF), is proposed to solve the aforementioned issues by inheriting the advantages of those previous semi-supervised NMF methods. Specifically, DCNMF simultaneously uses two types of supervisory information in the forms of the pointwise and pairwise constraints of labeled data points into the objective function for learning the more efficient and discriminative data representation. On the one hand, DCNMF explicitly utilizes the exact pointwise constraints into the convex NMF, that guarantees the learned low-dimensional data representations with same (or different) labels to be close (or far) enough. On the other hand, DCNMF propagates the pairwise constraints obtained from the labeled samples to the unlabeled samples by using CPA and reconstructs a new data weight matrix that exploits the intrinsic geometrical information of data. Since the pairwise constraints are directly derived from the labeled samples, only a small amount of label information is required for the proposed method. That means DCNMF can make full use of the limited supervised information to improve the performance. The framework of the proposed DCNMF is roughly shown in Fig. 1. Moreover, DCNMF can process the mixed-sign data, because it imposes the nonnegative constraint only on the coefficient matrix but not on the base matrix. By optimizing the objective function of DCNMF, an efficient alternating iterative algorithm is derived for the proposed method with convergence guaranteed. The computational complexity of DCNMF is analyzed by comparing with several related methods, and the relationships between DCNMF and some typical NMF based methods are also discussed. Experimental results confirm

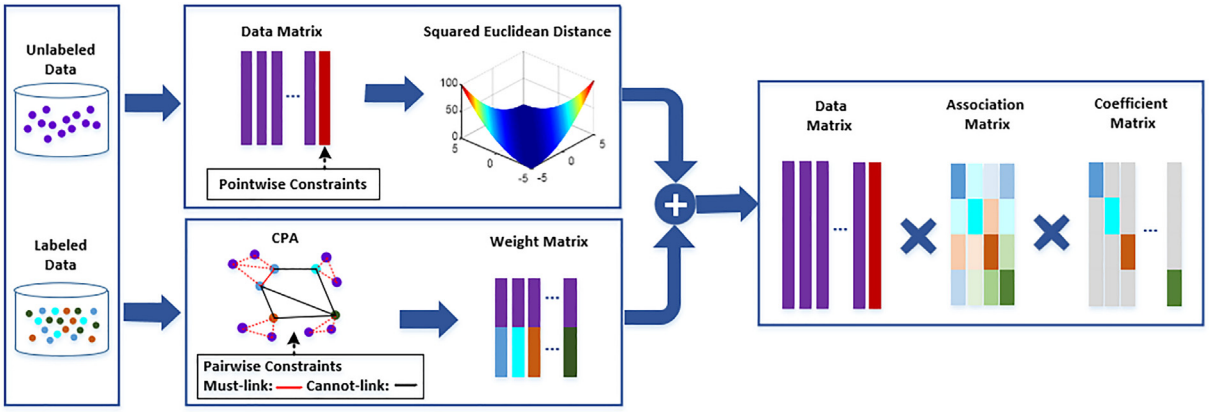

Fig. 1. The framework of the proposed DCNMF method. From the beginning, the data (including the labeled and unlabeled samples) are encoded as the data matrix with the pointwise constraints, and are utilized to reconstruct the new weight matrix by CPA with the pairwise constraints. After finite iterations, the proposed DCNMF method will learn a good coefficient matrix as the low-dimensional representation of the data matrix.

the effectiveness of DCNMF in clustering tasks on nonnegative and mixed-sign datasets by making a comparison with the related NMF methods.

In summary, the main contributions of this work are fourfold: 1) a novel semi-supervised NMF method namely DCNMF is proposed, which incorporates two different supervised information into convex NMF for discovering the good low-dimensional data representation, while only a small amount of label information is required; 2) DCNMF can handle mixed-sign data, which enlarges the range of applications; 3) the alternating iterative update rules of DCNMF are derived, and the convergence and computational complexity are analyzed; 4) the relationships between DCNMF and several typical NMF based methods are discussed.

The rest of this paper is organized as follows. In Section 2, the related works are reviewed. DCNMF is derived and analyzed in Section 3. Section 4 discusses the relationships between DCNMF and several previous NMF methods. Extensive experiments are conducted to validate the effectiveness of DCNMF in Section 5. Finally, Section 6 draws the conclusion and future works. For clarity, some important abbreviations that have been used in this paper are listed in Table 1.

2. Related works

Thanks to the good characteristics of NMF [6], NMF has a variety of applications in machine learning and data mining, and many advanced versions have been proposed in recent years for solving different issues [25–28]. Considering that the proposed method is a kind of semi-supervised method and can process the mixed-sign data, we mainly review the semi-supervised NMF methods, the semi-NMF and convex NMF methods.

2.1. Semi-supervised NMF methods

Many researchers have found that using limited supervised information in NMF can improve the performance of algorithms, and accordingly several semi-supervised NMF algorithms have been proposed. For instance, Liu et al. first proposed the CNMF method by enforcing the label information into the objective function of NMF [19]; Li et al. incorporated the local manifold information and the label information of data as regularization into NMF, and derived the graph based discriminative NMF (GDNMF) method [29]; Li et al. also further proposed a novel semi- supervised graph-based discriminative concept factorization (GDCF) method [30], where concept factorization (CF) can be regarded as an extension of NMF; Lu et al. utilized the pairwise constraints and local geometrical structure of data into CF, and derived the constrained neighborhood preserving concept factorization (CNPCF) method [31]; He et al. proposed the robust structured NMF (RSNMF) method for image representation by explicitly using the $l_{2,p}$-norm loss function and pursuing the block-diagonal structure [32]; Meng et al. combined the dual graph model and bi-orthogonal constraints into semi-supervised NMF, and presented the semi-supervised dual graph regularized NMF with sparse and orthogonal constraints (SODNMF) method [33]; Wang et al. proposed the CPSNMF method by implicitly utilizing the entire pairwise constraints of the data points [20]; Peng et al. proposed a robust CSNMF method by simultaneously using pointwise and pairwise constraints of the data points [23].

2.2. Semi-NMF and Convex NMF Methods

In order to enlarge the applicable range of NMF for processing mixed-sign data, semi-NMF and convex NMF methods have been proposed. For instance, Ding et al. proposed the original semi-NMF and convex NMF (Convex-NMF) methods by only enforcing the nonnegative constraint on the coefficient matrix [16]; Hu et al. further introduced the graph regularization into

Table 1
Abbreviations.

Abbreviations	Descriptions
NMF	nonnegative matrix factorization
PCA	principal component analysis
DCNMF	dual semi-supervised convex NMF
CPSNMF	semi-supervised NMF via constraint propagation
CPA	constraint propagation algorithm
CSNMF	correntropy based semi-supervised NMF
GDNMF	graph based discriminative NMF
GDCF	graph-based discriminative concept factorization
CNPCF	constrained neighborhood preserving concept factorization
RSNMF	robust structured NMF
SODNMF	dual graph regularized NMF with sparse and orthogonal constraints
GCNMF	graph regularized convex NMF
GGSemi-NMFD	group sparsity and graph regularized semi-NMF with discriminability
RFSNP	regularized fast semi-NMF-PCA
PGCNMF	pairwise constrained graph regularized convex NMF
NMI	normalized mutual information

the convex NMF, and derived the graph regularized convex NMF (GCNMF) method [18]; Luo et al. developed the group sparsity and graph regularized semi-NMF with discriminability (GGSemi-NMFD) method for obtaining good low-dimensional data representation [34]; Allab et al. proposed the graph Laplacian regularized fast semi-NMF-PCA (RFSNP) method [35]; Zhang et al. developed the pairwise constrained graph regularized convex NMF (PGCNMF) method [36], which is also a semi-supervised NMF method.

3. DCNMF algorithm

3.1. Notations

Throughout this paper, we utilize the uppercase boldface letters (e.g., $\mathbf{A}$), lowercase boldface letters (e.g., $\mathbf{a}$), and normal font letters (e.g., a) to stand for matrices, vectors, and scalars, respectively. In particular, $\mathbf{a}_i, \mathbf{a}_{i*}$ and a_{ij} (or $\mathbf{A}_{ij}$) respectively stand for the i-th column vector, the i-the row vector, and the (i,j)-th entry of matrix $\mathbf{A}$. $|\mathbf{A}|$ denotes that every entry of the matrix $\mathbf{A}$ obtains the absolute value. For clarity, some significant notations used in this work are listed in Table 2.

3.2. Optimization problem

As mentioned before, NMF aims to decompose the nonnegative data matrix into two low-dimensional nonnegative matrices (i.e., the base matrix and the coefficient matrix), such that the product of the obtained low-dimensional matrices approximates to the data matrix. In contrast to the original NMF, the convex NMF has no nonnegative constraint on the data matrix and the base matrix while only enforcing the nonnegative condition on the coefficient matrix. The base matrix is also restricted to be the convex combinations of the data samples. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{M\times N}$ denote the data matrix, in which N denotes the total number of sample and $\mathbf{x}_j$ stands for a sample vector with M elements. $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{x}_K] \in \mathbb{R}^{M\times K}$ and $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K] \in \mathbb{R}^{N\times K}_{\geqslant 0}$ respectively denote the base matrix and the coefficient matrix. Based on the definition of convex NMF, we have [16]:

$$\mathbf{X} \approx \mathbf{U}\mathbf{V}^T = \mathbf{X}\mathbf{W}\mathbf{V}^T \tag{1}$$

where $\mathbf{U} = \mathbf{X}\mathbf{W}$, and $\mathbf{W} = [w_{jk}] \in \mathbb{R}^{N\times K}_{\geqslant 0}$ is a nonnegative association matrix. Compared with NMF, the convex NMF has the following advantages: 1) the convex NMF can handle the mixed-sign data matrix; 2) the columns of $\mathbf{W}$ can better capture the cluster centroid for reasons of interpretability.

Previous studies have shown that the performance of algorithms can be improved by using limited supervised information. Assume that the first L data samples are labeled, and the rest of data samples are unlabeled. Each data sample has one class and the labeled samples have C classes. Similar to [19,37], we built a label constraint matrix $\mathbf{A} \in \mathbb{R}^{N\times(N+C-L)}_{\geqslant 0}$:

$$\mathbf{A} = \begin{bmatrix} \mathbf{C}_{L\times C} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{N-L} \end{bmatrix} \tag{2}$$

Table 2
Notations.

Notations	Descriptions
$\mathbf{X}$	data matrix
$\mathbf{U}$	basis matrix
$\mathbf{V}$	coefficient matrix
$\mathbf{W}$	association matrix
$\mathbf{A}$	label constraint matrix
$\mathbf{Z}$	auxiliary matrix
$\mathbf{S}$	weight matrix
$\tilde{\mathbf{S}}$	weight matrix after using CPA
$\tilde{\mathbf{D}}$	diagonal matrix whose entries are column sums of $\tilde{\mathbf{S}}$
$\tilde{\mathbf{L}}$	Laplacian matrix ($\tilde{\mathbf{L}} = \tilde{\mathbf{D}} - \tilde{\mathbf{S}}$)
α	balance parameter in CPA
β	graph regularization parameter
p	number of nearest neighbors on the graph
K	embedding dimension of NMF
$\|\cdot\|_F$	Frobenius norm
$[\cdot]^T$	transpose operator
$Tr(\cdot)$	trace of matrix
$E[\cdot]$	expectation operator
$\mathbb{R}^{M\times N}$	set of nonnegative real matrix of dimension M-by-N

where $\mathbf{C}_{L\times C}$ is an $L \times C$ indicator matrix with $c_{ij} = 1$ if $\mathbf{x}_i$ belongs to the jth class, otherwise $c_{ij} = 0$. When the matrix $\mathbf{A}$ is enforced in convex NMF by introducing an auxiliary matrix $\mathbf{Z} \in \mathbb{R}_{\geqslant 0}^{(N+C-L)\times K}$, we derive the constrained convex NMF, and rewrite (1) as follows:

$$\mathbf{X} \approx \mathbf{XWV}^T = \mathbf{XW}(\mathbf{AZ})^T \tag{3}$$

where $\mathbf{V} = \mathbf{AZ}$. Up to now, the squared Euclidean distance (SED) is still the most commonly used loss function in NMF to measure the similarity, due to its simplicity. When SED is adopted in (3), the objective function for the constrained convex NMF is derived:

$$\mathcal{O}_1 = \|\mathbf{X} - \mathbf{XW}(\mathbf{AZ})^T\|_F^2 = \sum_{i=1}^{M}\sum_{j=1}^{N}\left(\mathbf{X}_{ij} - \left(\mathbf{XWZ}^T\mathbf{A}^T\right)_{ij}\right)^2 \tag{4}$$

Algorithm 1: Constraint Propagation Algorithm (CPA)

Input: The data matrix $\mathbf{X}$, the initial pairwise constraints matrix $\mathbf{H}$, the parameters $\alpha \in (0, 1)$ and p.
Output: The new weight matrix $\tilde{\mathbf{S}}$.

1: Calculate the weight matrix $\mathbf{S}$ of data graph, and $\mathbf{S} = \left(\mathbf{S} + \mathbf{S}^T\right)/2$;
2: Calculate the graph Laplacian matrix $\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{SD}^{-1/2}$, where $\mathbf{D}$ is a diagonal matrix with $\mathbf{D}_{ii} = \sum_{j=1}^{N}\mathbf{S}_{ij}$;
3: **repeat**
4: $\quad\mathbf{F}_v(t+1) = \alpha\mathbf{LF}_v(t) + (1-\alpha)\mathbf{H}$;
5: **until** Convergence
6: **repeat**
7: $\quad\mathbf{F}_h(t+1) = \alpha\mathbf{F}_h(t)\mathbf{L} + (1-\alpha)\mathbf{F}_v^*$, where $\mathbf{F}_v^*$ denotes the limit of $\mathbf{F}_v$;
8: **until** Convergence
9: $\mathbf{F} = \mathbf{F}_h^*$, where $\mathbf{F}_h^*$ denotes the limit of $\mathbf{F}_h$;
10: Construct the new weight matrix $\tilde{\mathbf{S}}$ by

$$\tilde{s}_{ij} = \begin{cases} 1 - (1 - f_{ij})(1 - s_{ij}), & f_{ij} \geqslant 0 \\ (1 + f_{ij})s_{ij}, & f_{ij} < 0 \end{cases}$$

11: Return $\tilde{\mathbf{S}}$.

In order to further use the limited label information, the pairwise constraints of the labeled samples are obtained. Different from the weight matrix in traditional p-nearest neighbour graph regularized NMF methods, a new weight matrix $\tilde{\mathbf{S}}$ in DCNMF is constructed by using CPA that propagates the obtained pairwise constraints to the entire data [20,38]. All pairwise constraints information is reflected in the learned weight matrix. The complete CPA is illustrated in Algorithm 1, where α is the balance parameter, $\mathbf{H} = [h_{ij}] \in \mathbb{R}^{N\times N}$ is the initial pairwise constraints matrix with

$$h_{ij} = \begin{cases} +1, & (\mathbf{x}_i, \mathbf{x}_j) \in ML \\ -1, & (\mathbf{x}_i, \mathbf{x}_j) \in CL \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where $ML = \{(\mathbf{x}_i, \mathbf{x}_j)|l_i = l_j\}$ and $CL = \{(\mathbf{x}_i, \mathbf{x}_j)|l_i \neq l_j\}$ denote respectively the initial must-link constraints and cannot-link constraints, l_i stands for the label of sample $\mathbf{x}_i$, $\mathbf{F} = [f_{ij}] \in \mathbb{R}^{N\times N}$ denotes the propagated pairwise constraints matrix, and $\mathbf{F}_v$ (or $\mathbf{F}_h$) is the vertical (horizontal) propagation of $\mathbf{F}$. It is worth noting that $\tilde{\mathbf{S}}$ still has some nice properties such as nonnegativity and symmetry [39]. Under the clustering assumption, we have the smoothness of data samples:

$$\begin{aligned} \mathcal{O}_{data} &= \tfrac{1}{2}\sum_{i,j=1}^{N}\|\mathbf{v}_{i*}^T - \mathbf{v}_{j*}^T\|_2^2\widetilde{\mathbf{S}}_{ij} \\ &= Tr(\mathbf{V}^T\widetilde{\mathbf{D}}\mathbf{V}) - Tr(\mathbf{V}^T\widetilde{\mathbf{S}}\mathbf{V}) = Tr(\mathbf{V}^T\widetilde{\mathbf{L}}\mathbf{V}) = Tr(\mathbf{Z}^T\mathbf{A}^T\widetilde{\mathbf{L}}\mathbf{AZ}) \end{aligned} \tag{6}$$

where $\tilde{\mathbf{D}}$ is the diagonal matrix with $\tilde{\mathbf{D}}_{ii} = \sum_{j=1}^{N}\tilde{\mathbf{S}}_{ij}$, and $\tilde{\mathbf{L}} = \tilde{\mathbf{D}} - \tilde{\mathbf{S}}$ is the graph Laplacian matrix. Incorporating (6) into (4) as the graph regularization, the optimization problem of DCNMF is derived as follows:

$$\min_{\mathbf{W},\mathbf{Z}} \mathcal{O}_{DC} = \|\mathbf{X} - \mathbf{XWZ}^T\mathbf{A}^T\|_F^2 + \beta Tr(\mathbf{Z}^T\mathbf{A}^T\tilde{\mathbf{L}}\mathbf{AZ})$$

where $\mathcal{O}_{DC}$ is the objective function of DCNMF, and $\beta \geqslant 0$ denotes the parameter for the graph regularization.

3.3. Optimization algorithm

Although the objective function $\mathcal{O}_{DC}$ of DCNMF is convex with respect to the decomposed matrices $\mathbf{W}$ and $\mathbf{Z}$ separately, it is non-convex with respect to them together. In this case, an alternating iterative algorithm has been widely used to solve the optimization problem in (7), i.e., updating $\mathbf{W}$ with fixed $\mathbf{Z}$, and then updating $\mathbf{Z}$ with fixed $\mathbf{W}$ until convergence. First, we rewrite the objective function $\mathcal{O}_{DC}$ as:

$$\begin{aligned}\mathcal{O}_{DC} &= \|\mathbf{X}-\mathbf{X}\mathbf{W}\mathbf{Z}^T\mathbf{A}^T\|_F^2+\beta Tr(\mathbf{Z}^T\mathbf{A}^T\tilde{\mathbf{L}}\mathbf{A}\mathbf{Z})\\ &= Tr\left(\mathbf{X}\mathbf{X}^T-2\mathbf{X}\mathbf{W}\mathbf{Z}^T\mathbf{A}^T\mathbf{X}^T+\mathbf{X}\mathbf{W}\mathbf{Z}^T\mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{X}^T\right)+\beta Tr\left(\mathbf{Z}^T\mathbf{A}^T\tilde{\mathbf{L}}\mathbf{A}\mathbf{Z}\right)\end{aligned} \tag{8}$$

Since the matrices $\mathbf{W}$ and $\mathbf{Z}$ are nonnegative, the Lagrange function $\mathcal{L}_{DC}$ for DCNMF is given by defining $\Phi=[\Phi_{ik}]\in\mathbb{R}_{\geqslant 0}^{N\times K}$, and $\Psi=[\Psi_{jk}]\in\mathbb{R}_{\geqslant 0}^{(N+C-L)\times K}$ be the Lagrange multipliers for $\mathbf{W}$ and $\mathbf{Z}$ respectively:

$$\mathcal{L}_{DC} = Tr\left(\mathbf{X}\mathbf{X}^T-2\mathbf{X}\mathbf{W}\mathbf{Z}^T\mathbf{A}^T\mathbf{X}^T+\mathbf{X}\mathbf{W}\mathbf{Z}^T\mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{X}^T\right)+\beta Tr(\mathbf{Z}^T\mathbf{A}^T\tilde{\mathbf{L}}\mathbf{A}\mathbf{Z})+Tr(\Phi\mathbf{W}^T)+Tr(\Psi\mathbf{Z}^T) \tag{9}$$

Then the partial derivatives of $\mathcal{L}_{DC}$ for $\mathbf{W}$ and $\mathbf{Z}$ are derived as follows:

$$\frac{\partial\mathcal{L}_{DC}}{\partial\mathbf{W}}=-2\mathbf{X}^T\mathbf{X}\mathbf{A}\mathbf{Z}+2\mathbf{X}^T\mathbf{X}\mathbf{W}\mathbf{Z}^T\mathbf{A}^T\mathbf{A}\mathbf{Z}+\Phi \tag{10}$$

$$\frac{\partial\mathcal{L}_{DC}}{\partial\mathbf{Z}} = -2\mathbf{A}^T\mathbf{X}^T\mathbf{X}\mathbf{W}+2\mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W}+2\beta\mathbf{A}^T\tilde{\mathbf{L}}\mathbf{A}\mathbf{Z}+\Psi \tag{11}$$

Under the Karush–Kuhn–Tucker conditions [40], i.e., $\Phi_{ik}w_{ik}=0$ and $\Psi_{jk}z_{jk}=0$, we have:

$$-\left(\mathbf{X}^T\mathbf{X}\mathbf{A}\mathbf{Z}\right)_{ik}w_{ik}+\left(\mathbf{X}^T\mathbf{X}\mathbf{W}\mathbf{Z}^T\mathbf{A}^T\mathbf{A}\mathbf{Z}\right)_{ik}w_{ik}=0 \tag{12}$$

$$-\left(\mathbf{A}^T\mathbf{X}^T\mathbf{X}\mathbf{W}\right)_{jk}z_{jk}+\left(\mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W}\right)_{jk}z_{jk}+\beta\left(\mathbf{A}^T\tilde{\mathbf{L}}\mathbf{A}\mathbf{Z}\right)_{jk}z_{jk}=0 \tag{13}$$

Accordingly, the multiplicative update rules of DCNMF are easily derived:

$$\begin{aligned}w_{ik} &\leftarrow w_{ik}\sqrt{\frac{\left(\mathbf{M}^+\mathbf{A}\mathbf{Z}+\mathbf{M}^-\mathbf{W}\mathbf{Z}^T\mathbf{A}^T\mathbf{A}\mathbf{Z}\right)_{ik}}{\left(\mathbf{M}^-\mathbf{A}\mathbf{Z}+\mathbf{M}^+\mathbf{W}\mathbf{Z}^T\mathbf{A}^T\mathbf{A}\mathbf{Z}\right)_{ik}}}\\ z_{jk} &\leftarrow z_{jk}\sqrt{\frac{\left(\mathbf{A}^T\mathbf{M}^+\mathbf{W}+\mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{M}^-\mathbf{W}+\beta\mathbf{A}^T\tilde{\mathbf{S}}\mathbf{A}\mathbf{Z}\right)_{jk}}{\left(\mathbf{A}^T\mathbf{M}^-\mathbf{W}+\mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{M}^+\mathbf{W}+\beta\mathbf{A}^T\tilde{\mathbf{D}}\mathbf{A}\mathbf{Z})\right)_{jk}}}\end{aligned} \tag{15}$$

where $\mathbf{M}=\mathbf{X}^T\mathbf{X}, \mathbf{M}^+=\frac{|\mathbf{M}|+\mathbf{M}}{2}$ and $\mathbf{M}^-=\frac{|\mathbf{M}|-\mathbf{M}}{2}$ respectively denote the positive and negative parts of $\mathbf{M}$. The complete DCNMF algorithm is summarized in Algorithm 2.

Algorithm 2: DCNMF Algorithm

Input: Data matrix $\mathbf{X}$, label constraint matrix $\mathbf{A}$, and parameters α, p, β, and K.
Output: The data representation $\mathbf{V}$.
1: Initialize matrices $\mathbf{W}$ and $\mathbf{Z}$;
2: Construct the weight matrix $\tilde{\mathbf{S}}$ by using Algorithm 1;
3: **repeat**
4: Update $\mathbf{W}$ by using (14);
5: Update $\mathbf{Z}$ by using (15);
6: **until** Convergence
7: return $\mathbf{V}=\mathbf{A}\mathbf{Z}$.

Table 3
Computational operation counts of each iteration for DCNMF, PGCNMF, CPSNMF, CNPCF, GDCF, and GCNMF.

Methods	addition	multiplication	division	overall
DCNMF	$6KN^2+3NK^2+(N+C-L)K(3N+C-L+4)+N(p+7)K$	$6KN^2+3NK^2+(N+C-L)K(3N+C-L+1)+N(p+6)K$	$(2N+C-L)K$	$O(N^2K)$
PGCNMF [36]	$6KN^2+3NK^2+N(p+9)K$	$6KN^2+3NK^2+N(p+3)K$	$2NK$	$O(N^2K)$
CPSNMF [20]	$2MNK+2(M+N)K^2+N(p+3)K$	$2MNK+2(M+N)K^2+(N+M)K+N(p+1)K$	$(M+N)K$	$O(MNK)$
CNPCF [31]	$4KN^2+5NK^2+N(p+5)K$	$4KN^2+5NK^2+N(p+5)K$	$2NK$	$O(N^2K)$
GDCF [30]	$5KN^2+(6N+C)K^2+N(p+8+C)K$	$5KN^2+(6N+C)K^2+N(p+2+C)K$	$2NK$	$O(N^2K)$
GCNMF [18]	$4KN^2+4NK^2+N(p+3)K$	$4KN^2+4NK^2+N(p+3)K$	$2NK$	$O(N^2K)$

Since the update rules of DCNMF are obtained by using the majorization-minimization algorithm, the objective function $\hat{\mathcal{O}}_{DC}$ is monotonically nonincreasing.

Theorem 1. *1) Fixing* $\mathbf{Z}$*, the objective function* $\mathcal{O}_{DC}$ *decreases monotonically under the update rule (14); 2) Fixing* $\mathbf{W}$*, the objective function* $\mathcal{O}_{DC}$ *decreases monotonically under the update rule (15).*

The detailed proof can be seen in Appendix A.

3.4. Computational complexity analysis

The computational complexity of the proposed DCNMF method is analyzed in this subsection by comparing with several related NMF methods such as PGCNMF, CPSNMF, CNPCF, GDCF, and GCNMF. Without loss of generality, the big O notation is used to represent the complexity of algorithms. The number of three arithmetic operations including addition, multiplication, and division, are counted for each updating step of these compared methods, and the results are shown in Table 3. From this table, one can see that the computational load of DCNMF is higher than PGCNMF, CNPCF, GDCF, and GCNMF, due to $K \ll \min(N,M)$. When $M \gg N$, the computational cost of DCNMF is lower than CPSNMF and vice versa. However, the overall costs of DCNMF, PGCNMF, CNPCF, GDCF, and GCNMF for updating rules are the same, i.e., $O(N^2K)$. It is worth noting that $O((M+p)N^2)$ is required to construct the p-nearest neighbor graph and propagate constraints. When T is the number of iteration, the overall cost of DCNMF is $O(TKN^2+(M+p)N^2)$.

4. Relationships with several previous NMF methods

The proposed DCNMF method has the close relationships with some previous NMF methods, and can be considered as the special cases of DCNMF from three aspects.

- Convex NMF methods: When all supervised information is dropped, DCNMF will lead to the GCNMF method [18]. Furthermore, if the graph regularized parameter β is set to 0, the proposed method will further reduce to the original convex NMF mehtod [16].
- Semi-supervised NMF methods: When the data is nonnegative, and $\mathbf{U} \leftarrow \mathbf{XW}$ is used, DCNMF is equivalent to the CPSNMF method [20] if only pairwise constraint supervised information is considered and the label supervised information is disregarded. Conversely, the proposed method will result in the CNMF method [19] if only the label supervised information is utilized and $\beta = 0$.
- Unsupervised NMF methods: When the data is nonnegative, and any supervised information is not considered, DCNMF will reduce to the locally consistent concept factorization (LCCF) method [41], and the original concept factorization method [42] if $\beta = 0$. Moreover, using $\mathbf{U} \leftarrow \mathbf{XW}$, DCNMF will further lead to the GNMF method [14] and the original NMF method [6].

Clearly, it can be observed that, under different scenarios, the proposed method includes several popular versions of NMF, such as convex NMF, semi-supervised NMF, and traditional unsupervised NMF. In this situation, the proposed CCNMF method can be regarded as a unified framework for the NMF family.

5. Experiments

5.1. Experimental settings

5.1.1. Datasets

Eighteen available benchmark datasets are adopted in the clustering experiments, and Table 4 summarizes the important statistics of these datasets. Particularly, the datasets can be classified into two categories: mixed-sign dataset and nonneg-

Table 4
Statistics of the datasets.

Datasets	Data Sign	Samples	Dimensions	Classes
Breast	±	699	10	2
Control	±	600	60	6
Isolet	±	7797	617	26
PalmData25	±	2000	256	100
USPS	±	9298	256	10
Waveform	±	2746	21	3
MSRA25	+	1799	256	12
CMU PIE	+	2856	1024	68
UMIST	+	575	1024	20
Mpeg7	+	6000	1400	70
COIL100	+	7200	1024	100
MnistData	+	6996	784	10
FBIS	+	2463	2000	17
k1a	+	2340	21839	20
la12	+	6279	31472	6
tr41	+	878	7454	10
tr45	+	690	8261	10
wap	+	1560	8460	20

ative sign dataset. The first category includes six datasets, which are Breast, Control, Isolet, PalmData25, USPS,[1] and Waveform. The second category contains twelve datasets, in which MSRA25,[2] CMU PIE[3] and UMIST[4] are the face image datasets, Mpeg7 and COIL100[5] are the object image datasets, MnistData is the handwritten image dataset, and FBIS,[6] k1a, la12, tr41, tr45 and wap are the document datasets. Since each document is represented by using the term-frequent vector in the document datasets, the term frequency inverse document frequency (TF-IDF) [43] weighting strategy is applied to pre-process the documents. For all datasets, the data are formatted as the $M \times N$ data matrices.

5.1.2. Evaluation metrics

In the clustering experiments, two popular evaluation metrics, i.e., the accuracy (ACC) and the normalized mutual information (NMI) [44,45], are employed to evaluate the clustering performance by comparing the real class labels provided by the datasets with the learned labels.

The accuracy measures the percentage of correct label, whose definition is shown as follows:

$$\mathrm{ACC} = \frac{\sum_{j=1}^{N} \delta(r_j, map(\bar{r}_j))}{N} \tag{16}$$

where r_j and $\bar{r}_j$ denote the real label and the learned cluster label respectively, $\delta(r_j, \bar{r}_j)$ is the delta function that equals 1 if $r_j = \bar{r}_j$ and equals 0 otherwise, and $map(\cdot)$ stands for the mapping function that relies on the Kuhn-Munkres algorithm [46].

The normalized mutual information is a measure to calculate the information shared between two statistical distributions, whose definition is given by:

$$\mathrm{NMI} = \frac{\mathrm{MI}(\mathbb{C}, \bar{\mathbb{C}})}{\max(H(\mathbb{C}), H(\bar{\mathbb{C}}))} \tag{17}$$

where $\mathbb{C}$ and $\bar{\mathbb{C}}$ are respectively the set of clusters from the ground truth and the clustering method, $H(\mathbb{C})$ and $H(\bar{\mathbb{C}})$ denote the entropies of $\mathbb{C}$ and $\bar{\mathbb{C}}$, and $\mathrm{MI}(\mathbb{C}, \bar{\mathbb{C}})$ stands for the mutual information (MI) of $\mathbb{C}$ and $\bar{\mathbb{C}}$, defined by

$$\mathrm{MI} = \sum_{c_i \in \mathbb{C}, \bar{c}_j \in \bar{\mathbb{C}}} p(c_i, \bar{c}_j) \log \frac{p(c_i, \bar{c}_j)}{p(c_i) p(\bar{c}_j)} \tag{18}$$

where $p(c_i)$ and $p(\bar{c}_j)$ are the probabilities that a data point arbitrarily selected from the dataset belongs to the clusters $\mathbb{C}$ and $\bar{\mathbb{C}}$ respectively, $p(c_i, \bar{c}_j)$ is the joint probabilities that this data point belongs to both of the clusters $\mathbb{C}$ and $\bar{\mathbb{C}}$.

The values of ACC and NMI are in the range of 0 to 1. And the larger the values of the used evaluation metrics are, the better the clustering performance.

[1] USPS: U.S. Postal Service.
[2] MSRA: Microsoft Research Asia.
[3] CMU PIE: Carnegie Mellon University, Pose, Illumination, and Expression.
[4] UMIST: University of Manchester Institute of Science and Technology.
[5] COIL: Columbia Object Image Library.
[6] FBIS: Foreign Broadcast Information Service.

5.1.3. Compared methods

We compare DCNMF[7] with two categories of NMF based methods on the mixed-sign and nonnegative sign datasets respectively for clustering applications:

- The first category includes the Convex-NMF [16], GCNMF [18], GGSemiNMFD [34], RFSNP [35] and PGCNMF [36] methods, which are the semi-NMF (or convex NMF) methods used for the mixed-sign datasets.
- The second category includes the RSNMF [32], GDCF [30], SODNMF [33], CNPCF [31], PGCNMF and CPSNMF [20] methods, which are the semi-supervised methods utilized for the nonnegative sign datasets.

It should be remarked that in the first category, all compared methods are unsupervised expect for PGCNMF. However, in the second category, all compared methods are the semi-supervised NMF based methods. The PGCNMF method is adopted in both categories, because it is not only a convex NMF method but also a semi-supervised NMF method. Except for Convex-NMF and RSNMF, other methods are the graph regularized NMF methods. Without mentioning otherwise, we use the suggested parameters in each of the reference articles (see above) for those compared methods. The overall experimental process includes two stages separately for all methods. The first stage is to learn the low-dimensional data representation matrix, then the second stage is to apply the standard K-means algorithm to the learned representation matrix for obtaining the cluster label under the permutation mapping function.

5.1.4. Parameter settings

- The proposed method has four essential parameters, i.e., α, p, β, and K. In the experiments, the parameters α and p are set to 0.2 and 5 empirically. The value of β is selected in the grid $\{10, 50, 100, 500, 1000\}$, and the parameter K is set to N_c, where N_c denotes the class number of the dataset. The heat kernel weighting method is used for constructing the weight matrix with the kernel bandwidth being 1.
- 10% samples are randomly selected from each class as the available label information for all datasets. Note that additional experiments will be conducted and discussed later.
- We randomly initialize the matrices **W** and **Z**, and the values of the entries in **W** and **Z** are selected between 0 and 1.

5.2. Clustering results on mixed-sign datasets

Fig. 2 shows the clustering results of different methods on the mixed-sign datasets via categorical scatter plots. Without mentioning otherwise, the clustering experiments are conducted on 20 independent test runs with different initial values, and the average clustering performance is calculated as the final results. The maximal number of the iterations is set to 300, and the best result is highlighted. From this table,we have the following observations:

- the proposed DCNMF method has better clustering results than other compared methods on the average ACC and NMI for all cases, since DCNMF pays attention to the labeled data points and the unlabeled data points simultaneously. That indicates DCNMF is the most effective method for learning the latent low-dimensional data representation;
- the semi-supervised methods, i.e., PGCNMF and DCNMF, obviously outperform the unsupervised methods such as GCNMF, GGSemiNMFD and RFSNP. The main reason is illustrated in Fig. 3, where it can be seen that the weight matrices learned by PGCNMF and DCNMF are more informative than the unsupervised methods. That confirms the importance of semi-supervised information for enhancing the performance in clustering applications;
- For the compared methods, Convex-NMF achieves the worst clustering performance on all mixed-sign datasets. The main reason may be that, Convex-NMF fails to consider the intrinsic geometrical information of the data and the semi-supervised information, which always bring some improvement for clustering performance.

5.3. Clustering results on nonnegative datasets

Fig. 4 illustrates the clustering performance of DCNMF against the semi-supervised methods, i.e., RSNMF, GDCF, SODNMF, CNPCF, PGCNMF and CPSNMF, in terms of ACC and NMI on the nonnegative datasets. It can be observed from this table that in most situations, DCNMF is superior to other semi-supervised methods. For example, the clustering results of DCNMF can achieve great improvement in terms of ACC and NMI respectively on the MSRA25 and MnistData datasets, when compared with the second best results. On the CMU PIE and la12 datasets, the experimental results of the proposed method in NMI are still comparable to the best ones. The main reason is that, compared with these compared semi-supervised methods that only use the exact label information or the pairwise constraints of labels, DCNMF adopts two types supervised information simultaneously to obtain more discriminating data representation to enhance the performance in clustering applications. Moreover, the pairwise constraints based semi-supervised methods, i.e., CNPCF, PGCNMF, CPSNMF and DCNMF, have relatively better performance than RSNMF, GDCF and SODNMF, which directly use the exact label information. That demonstrates, when the number of the labeled samples is fixed, the pairwise constraints supervised information has larger contributions than the pointwise constraints supervised information in the semi-supervised methods. Last, DCNMF and

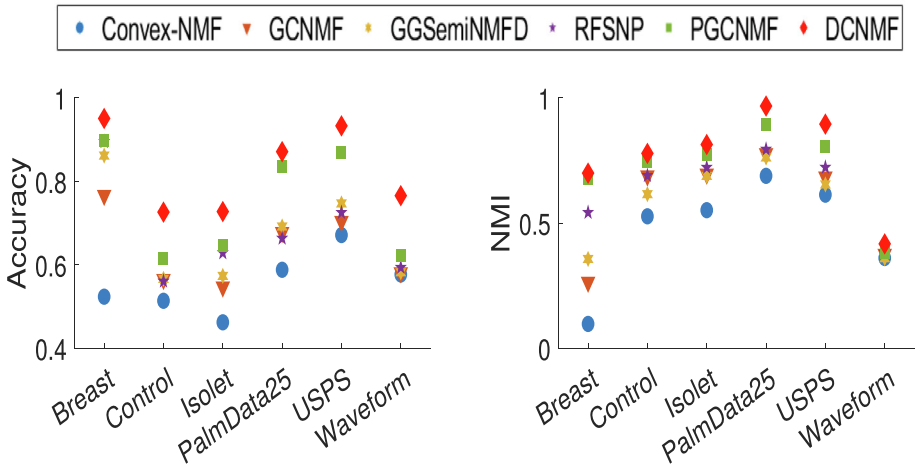

Fig. 2. Clustering results on the mixed-sign datasets.

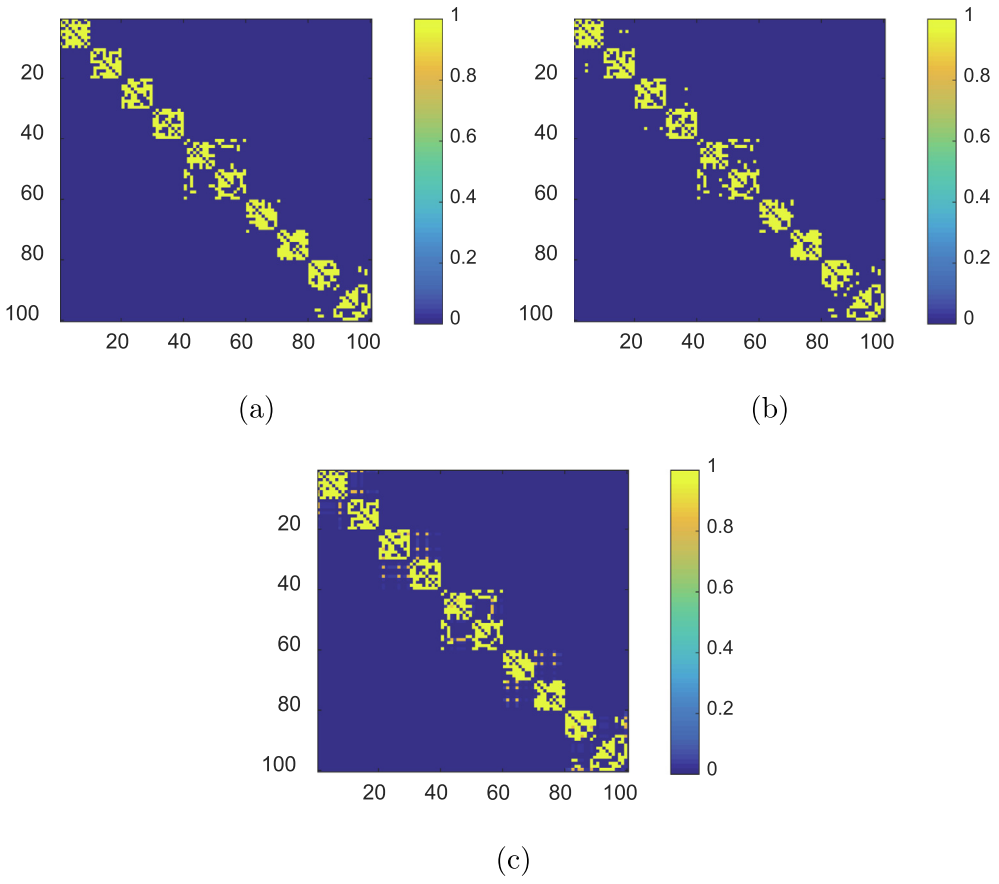

Fig. 3. Visual comparison of the weight matrices on PalmData25 dataset learned by (a) *p*-nearest neighbors (for GCNMF, GGSemiNMFD, and RFSNP); (b) *p*-nearest neighbors with pairwise constraints (for PGCNMF); (c) *p*-nearest neighbors with CPA (for DCNMF).

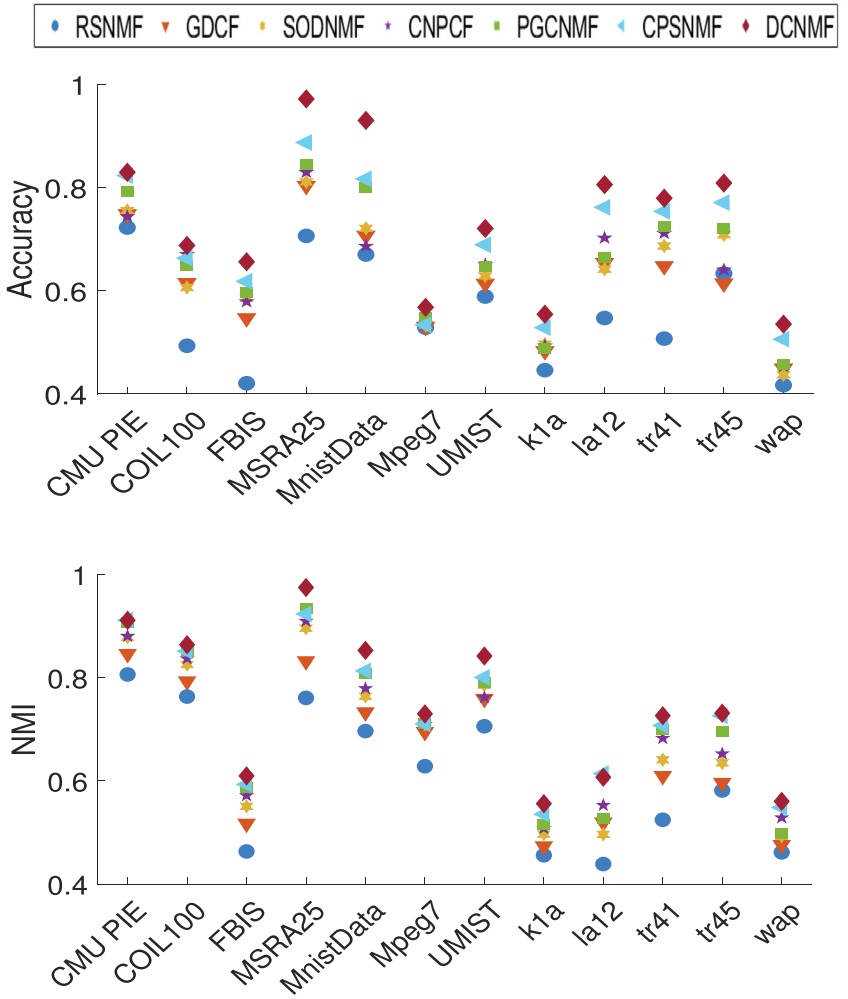

Fig. 4. Clustering results on the nonnegative datasets.

CPSNMF usually outperform other compared methods, because they can make full use of the limited supervised information to learn the more informative weight matrices as shown in Fig. 5.

5.4. Clustering results on real-world dataset

In this subsection, we apply the proposed DCNMF method to a real-world functional magnetic resonance imaging (fMRI) dataset with natively sparse label information. The mixed-sign fMRI dataset is measured from human brain activity with different orientations in Brain Imaging Center, Institute of Biophysics, Chinese Academy of Sciences. Fig. 6 shows an example of brain activity map for the real-world fMRI dataset. Specifically, the fMRI dataset contains 192 samples, and the dimension of each sample is 219,307 after data preprocessing. The used fMRI dataset is acquired by utilizing a 3.0-Tesla Siemens MRI scanner with 20 channel head neck coil to scan each orientation. For the fMRI dataset, the label information of 16 samples is available from domain experts, and the labeled samples come from eight different orientations. In order to evaluate the goodness of clustering results, two internal clustering validation measures (i.e., Dunn's index and Silhouette index [47]) are adopted in the experiments, sine there is no fully specified label information for all samples. In particular, Dunn's index utilizes the minimum pairwise distance between samples in different clusters as the intercluster separation and the maximum diameter among all clusters as the intracluster compactness. Silhouette index measures the clustering result based on the pairwise difference of between-cluster and within-cluster distances. Usually, for Dunn's index and Silhouette index, a higher value indicates a better clustering result.

Table 5 shows the clustering performance of DCNMF and the compared methods on the real-world fMRI dataset, in which the best result is highlighted. From this table, one can see that the proposed DCNMF method outperforms these compared methods in terms of Dunn's index and Silhouette index for all situations, which validates the effectiveness of DCNMF in clus-

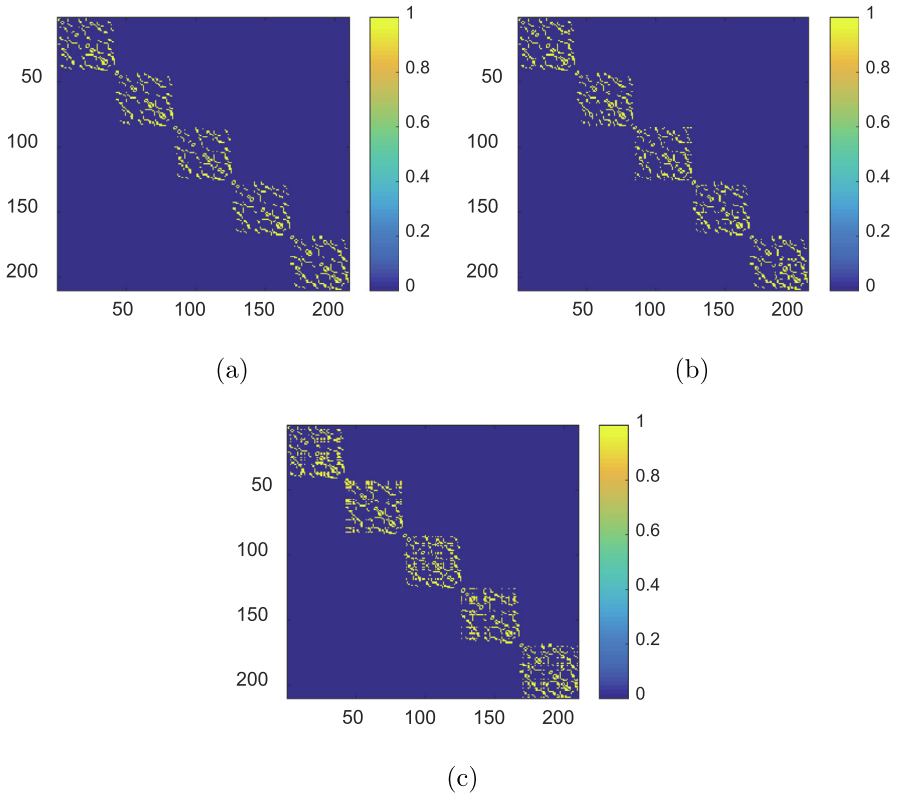

Fig. 5. Visual comparison of the weight matrices on CMU PIE dataset learned by (a) *p*-nearest neighbors (for GDCF, and SODNMF); (b) *p*-nearest neighbors with pairwise constraints (for CNPCF, and PGCNMF); (c) *p*-nearest neighbors with CPA (for CPSNMF, and DCNMF).

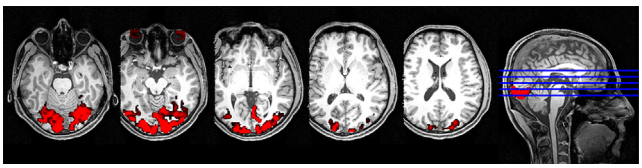

Fig. 6. An example of brain activity map for the real-world fMRI dataset.

Table 5
Clustering results on the real-world fMRI dataset.

Dataset	Dunn's index					
	Convex-NMF	GCNMF	GGSemiNMFD	RFSNP	PGCNMF	DCNMF
fMRI	0.601	0.626	0.631	0.619	0.640	**0.667**
Dataset	Silhouette index					
	Convex-NMF	GCNMF	GGSemiNMFD	RFSNP	PGCNMF	DCNMF
fMRI	0.363	0.417	0.488	0.390	0.523	**0.579**

tering task with a real-world data. Moreover, the semi-supervised methods, i.e., PGCNMF and DCNMF, usually have better clustering performance than other unsupervised methods. This means the limited supervised information can improve the performance in real-world data clustering applications for semi-supervised NMF methods.

5.5. Statistical significance test

In order to analyze whether the proposed DCNMF method is statistically different to other methods in comparison, Friedman test in [48] combination with the Nemenyi post hoc test is used for all the compared methods on the mixed-sign and

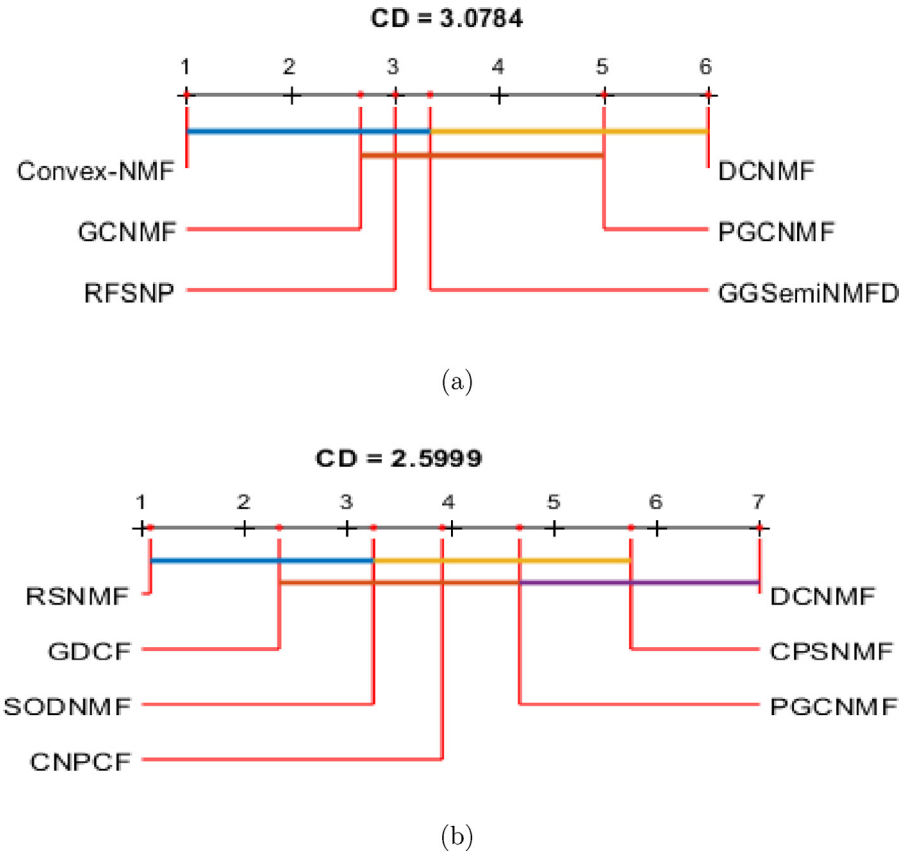

Fig. 7. Statistical comparison of accuracy of all compared methods using the Friedman test in combination with the Nemenyi post hoc test. (a) All compared methods on the mixed-sign datasets. (b) All compared methods on the nonnegative datasets.

nonnegative datasets, respectively. The statistical comparison of accuracy for the mixed-sign and nonnegative datasets is shown in Fig. 7. It is worth noting that the average rank of each method over all datasets is computed for significance testing, and also is plotted on the horizontal axis in ascending rank order. Specifically, the larger the value of the mean rank is, the better the performance of the method. Furthermore, the connected methods are significantly different at a certain significance level (0.05), when the rank difference between the connected methods is larger than the critical difference (CD). From Fig. 7, one can observe that on the mixed-sign datasets, the rank differences between DCNMF and each of Convex-NMF, GCNMF and RFSNP, respectively are higher than the CD (3.0784). Hence, the performance of DCNMF is considered statistically better than that of the Convex-NMF, GCNMF and RFSNP methods. The proposed DCNMF method is considered statis-

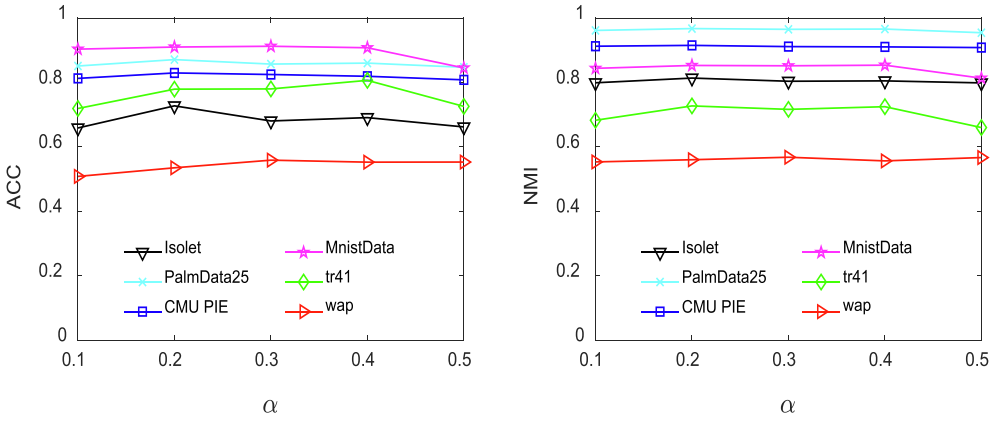

Fig. 8. Clustering performance of DCNMF versus parameter α.

tically similar to GGSemiNMFD and PGCNMF since the rank differences among DCNMF and these two methods are within the CD.Similarly, on the nonnegative datasets, one can see that the performance of DCNMF is considered statistically better than that of the RSNMF, GDCF, SODNMF and CNPCF methods, while statistically similar to PGCNMF and CPSNMF. Although the performance of DCNMF is statistically similiar to that of PGCNMF for both mixed-sign and nonnegative datasets, we notice that for nonnegative datasets, PGCNMF is only the third highest ranked method and it just lies at the edge of the other side of the CD in comparison with DCNMF.

5.6. Parameters selection

The proposed DCNMF method has four essential parameters, i.e., α, p, β, and K, that are required to be set in advance. However, up to now, these still has no widely-accepted universal strategy for the proposed method to select the optimal values for these parameters. In most cases, these parameters are determined empirically. Hence, we test the influence of those parameters with different values for DCNMF on the Isolet, PalmData25, CMU PIE, MnistData, tr41, and wap datasets. Figs. 8–11 illustrate the clustering results of DCNMF varying with $\alpha \in \{0.1, 0.2, 0.3, 0.4, 0.5\}, p \in \{3, 5, 7, 9, 11\}, \beta \in \{10, 50, 100, 500, 1000\}$, and $K \in \{10, 30, 50, 70, 90, 110\}$ respectively. It is remarked that, the parameters α, p, β, and K are respectively fixed to be $0.2, 5, 100$, and N_c, when they do not vary. From Fig. 8, one can observe that, the clustering performance of DCNMF is relatively good and stable, when α varies from 0.2 to 0.4. If the balance parameter α is too small or too large, the performance of DCNMF may decline. The main reason may be that when α is too small, DCNMF cannot propagate the pairwise constraints to the unconstrained data points entirely by using CPA. When α is too large, DCNMF fails to take full advantage of the initial pairwise constraint supervised information. Moreover, since the parameters p and β directly relate to the graph regularization, we discuss the selection of them together. One can find that from Figs. 9 and 10, DCNMF achievers good clustering results when p is set to 5 and β is set in the range of $[50, 500]$ respectively. In fact, the values of $p = 5$ and $\beta = 100$ used in DCNMF have been widely adopted in previous graph regularized NMF based algorithms. According to Fig. 11,

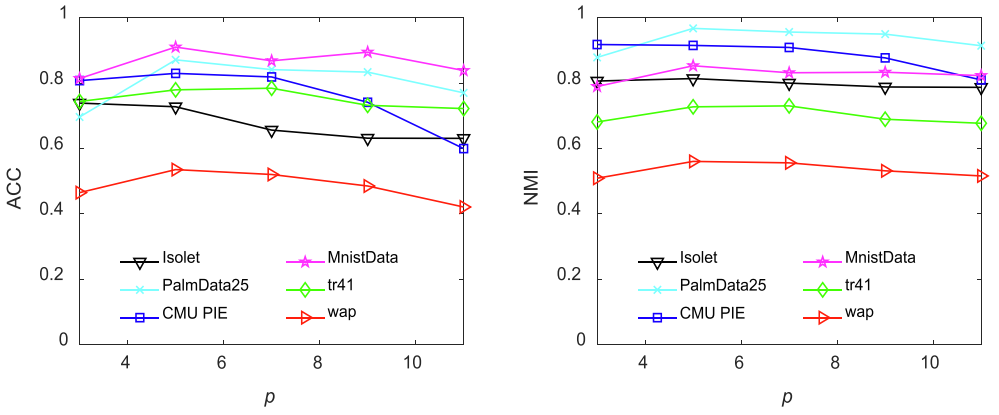

Fig. 9. Clustering performance of DCNMF versus parameter p.

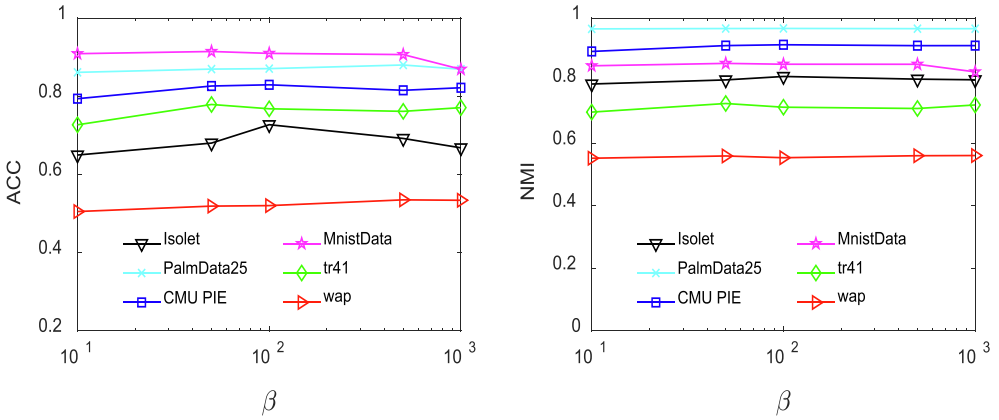

Fig. 10. Clustering performance of DCNMF versus parameter β.

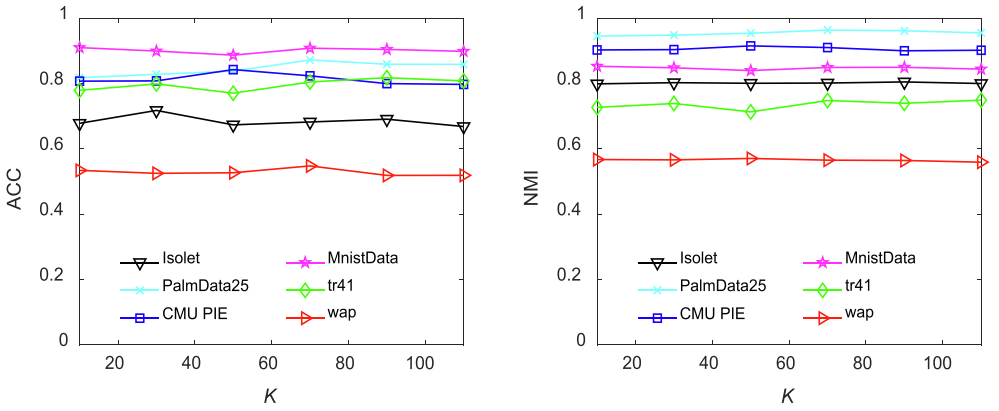

Fig. 11. Clustering performance of DCNMF versus parameter *K*.

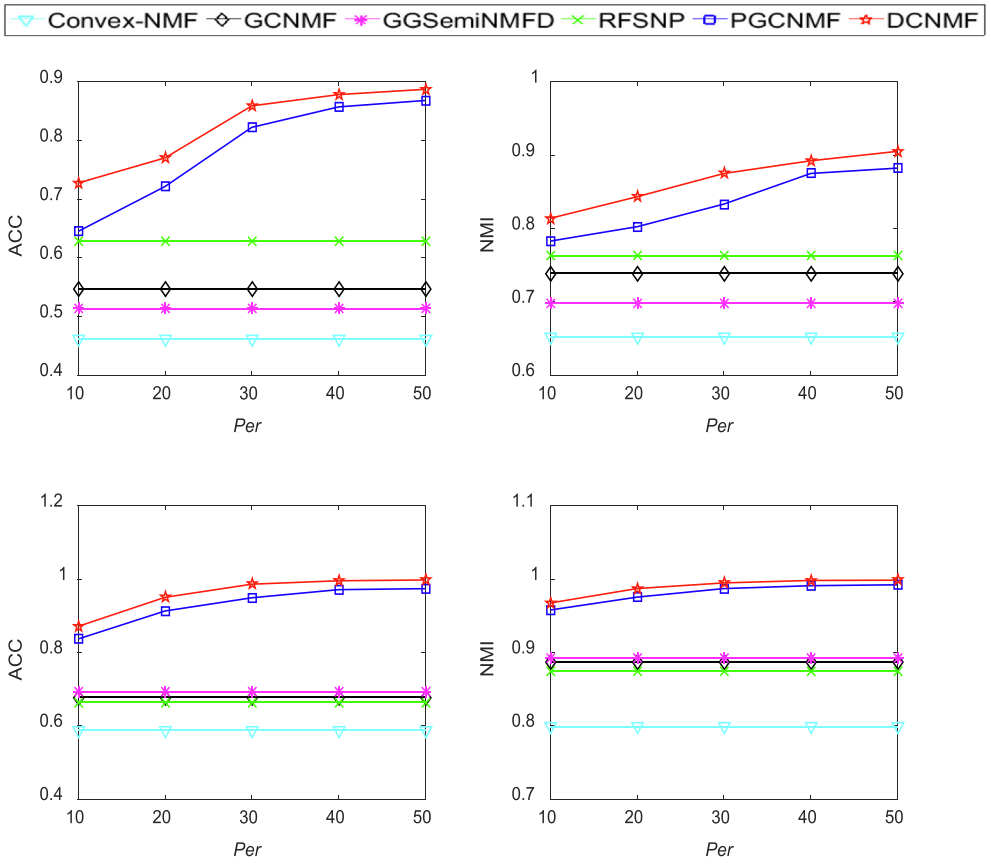

Fig. 12. Clustering performance varied with the percentages of labeled samples on two mixed-sign datasets. The first and second rows are obtained on the Isolet and PalmData25 datasets respectively.

DCNMF has favorable clustering performance when *K* varies from 10 to 110. However, in order to accurately cluster a given dataset which consists of K clusters, it is better to project the data matrix into a K-dimensional subspace. In this situation, each axis corresponds to a particular cluster. Therefore, in this work, *K* is usually set to the number of truth classes for each dataset.

5.7. Effect of supervised information

The effect of supervised information is evaluated on mixed-sign and nonnegative datasets in this subsection. Fig. 12 illustrates the clustering performance of DCNMF and the compared methods with the varied parameter *Per* on mixed-sign data-

sets (i.e., Isolet and PalmData25), where $Per \in \{10, 20, 30, 40, 50\}$ stands for the percentage of labeled samples for each class. Obviously, with the increasing of *Per*, the performance of both PGCNMF and DCNMF consistently improves and outperforms the performance of Convex-NMF, GCNMF, GGSemiNMFD, and RFSNP. It is because PGCNMF and DCNMF are the semi-supervised methods, while other compared methods are the unsupervised methods. The clustering results of the unsupervised methods are unchanged, since the supervised information has no influence on them. Besides, DCNMF has better clustering results than PGCNMF in all cases. The main reason is that DCNMF can utilize the supervised information more fully than PGCNMF. PGCNMF and DCNMF is significantly superior to the unsupervised methods, demonstrating the importance of supervised information to improve the performance of semi-supervised methods.

The experimental results on nonnegative datasets (i.e., CMU PIE, MnistData, tr41 and wap) are shown in Fig. 13. Similarly, from this figure, one can see that the clustering performance of all semi-supervised methods can enhance by increasing the percentages of labeled samples, and the proposed method usually has the best clustering results in terms of ACC and NMI. In

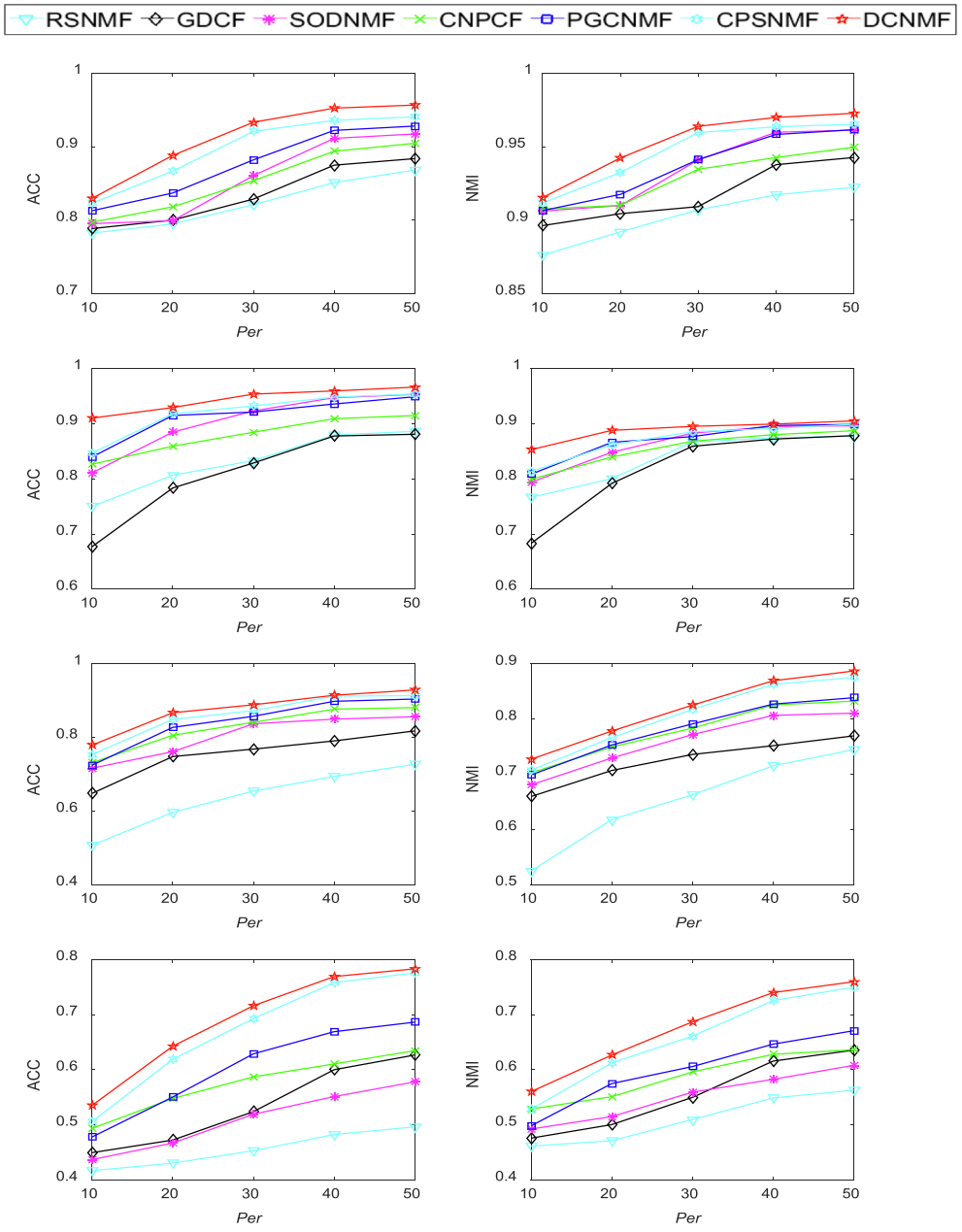

Fig. 13. Clustering performance varied with the percentages of labeled samples on four nonnegative datasets. The first, second, third, and fourth rows are obtained on the CMU PIE, MnistData, tr41, and wap datasets respectively.

addition, DCNMF and CPSNMF obviously outperform other compared semi-supervised methods in most situations. That indicates the pairwise constraint propagation has larger contributions than the pairwise constraint without propagation and the single label information to enhance the performance in clustering applications.

Furthermore, in order to show that DCNMF is robust to batch effects and observation bias, some additional experiments have also been done in this subsection, where for each dataset, instead of selecting 10% from each class earlier, 5% labeled samples are randomly selected from half of the classes and 25% labeled samples are randomly selected from the other half of the classes. The experimental results are demonstrated in Fig. 14, in which D1-D6 stand for the Isolet, PalmData25, CMU PIE, MnistData, tr41 and wap datasets respectively. One can see from this figure that when 5% labeled samples are randomly selected from half of the classes for each dataset and 25% labeled samples are randomly selected from the other half of the classes, the clustering results of the proposed DCNMF method are slightly better on the tr41 and wap datasets than that of DCNMF with picking 10% labeled samples from each class, but slightly worse on other datasets. Therefore, DCNMF has

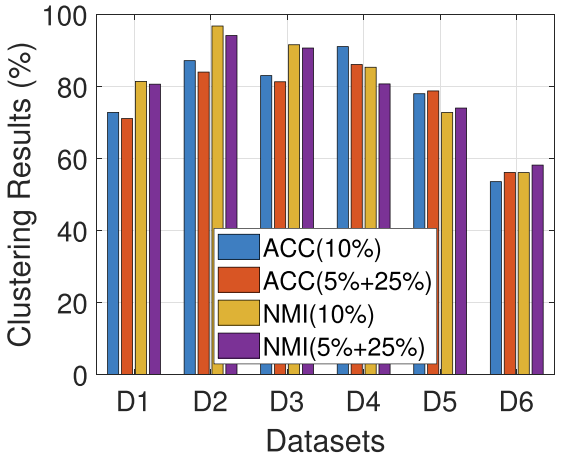

Fig. 14. Clustering performance of DCNMF varied with the percentages of labeled samples on six datasets, where D1-D6 denote the Isolet, PalmData25, CMU PIE, MnistData, tr41 and wap datasets respectively.

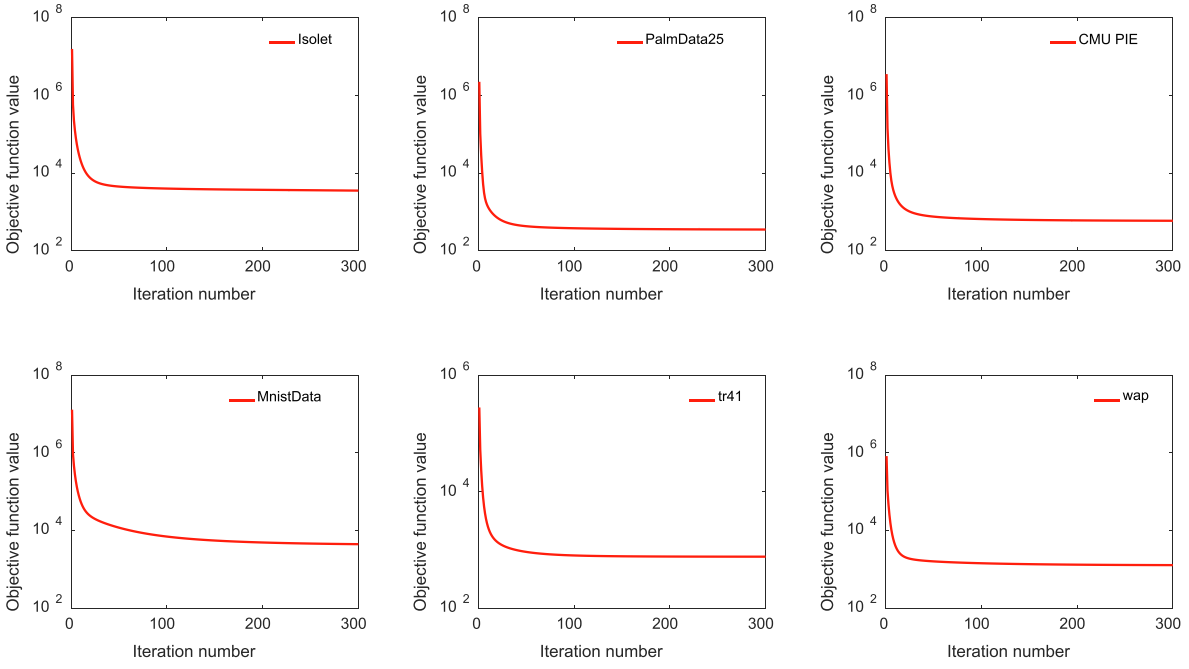

Fig. 15. Convergence curves of DCNMF on six datasets.

Table 6
Average running time (second) on the mixed-sign datasets.

Methods	Convex-NMF	GCNMF	GGSemiNMFD	RFSNP	PGCNMF	DCNMF
Breast	2.36	1.97	0.29	0.47	0.99	2.42
Control	0.51	0.43	0.09	0.17	0.22	0.91
Isolet	1008.24	794.25	105.02	216.68	578.22	1387.11
PalmData25	276.14	238.27	29.97	61.43	82.62	320.54
USPS	1424.52	1027.42	102.12	208.65	815.49	1619.17
Waveform	59.76	47.17	1.19	2.05	31.66	84.77

Table 7
Average running time (second) on the nonnegative datasets.

Methods	RSNMF	GDCF	SODNMF	CNPCF	PGCNMF	CPSNMF	DCNMF
MSRA25	18.72	11.74	109.78	74.98	39.97	32.47	90.08
CMU PIE	44.12	183.25	289.30	241.71	165.33	74.89	253.54
UMIST	9.87	13.73	27.05	18.14	10.82	10.31	19.13
Mpeg7	16.85	21.84	56.35	28.68	7.71	29.32	32.16
COIL100	167.72	685.29	1713.07	1114.68	718.99	521.12	1550.99
MnistData	33.98	278.69	426.44	367.85	251.56	377.55	418.51
fbis	30.78	123.36	277.81	497.65	326.29	138.78	569.87
k1a	115.77	202.68	276.21	511.79	289.23	216.76	470.70
la12	871.75	1127.34	3461.77	5426.96	6776.67	1044.83	6402.46
tr41	34.42	41.24	77.56	94.21	46.27	37.77	81.01
tr45	7.65	10.54	13.35	26.05	4.71	11.52	19.02
wap	26.36	80.72	115.34	151.73	106.79	68.74	146.86

relatively stable clustering performance when the different percentages of labeled samples for each class are used. This indicates that the proposed DCNMF method can be robust to batch effects and observation bias.

5.8. Convergence study

In this subsection, the convergence of DCNMF is validated experimentally. Fig. 15 demonstrates the convergence curves of DCNMF on six datasets. Particularly, for each subfigure, the x-axis and y-axis respectively stand for the iteration number and the value of the objective function $\hat{\mathcal{O}}_{DC}$. One can see from Fig. 15 that the DCNMF method with multiplicative update rules converges before 200 iterations, illustrating that DCNMF is effective and converges quickly.

5.9. Comparison of running time

The running time of different compared methods is shown in this subsection, where the results are respectively illustrated in Table 6 and Table 7 for the mixed-sign and nonnegative sign datasets. Especially, the execution time of the decomposition and clustering steps is computed for all methods on an Intel Core i9-10,900 K 3.70 GHz Window Server with 64 G memory. From the two tables, one can observe that the proposed DCNMF method requires the longest running time on the mixed-sign datasets, and the second longest time on the nonnegative datasets. It can be seen from Tables 5,6 that the proposed method may demand more computational time than most other compared methods in clustering applications. Generally speaking, for some big datasets (e.g., COIL100, Mpeg7, k1a, and la12), the CPSNMF method may be more efficient than DCNMF. However, CPSNMF is only applied for the nonnegative datasets, which limits the range of applications.

6. Conclusion

In this paper, the dual semi-supervised convex nonnegative matrix factorization (DCNMF) is proposed, which uses the pointwise constraints and the corresponding pairwise constraints as the dual semi-supervised information in convex NMF to obtain the good low-dimensional data representation, while only limited labeled samples are required. Due to the property of convex NMF that enforces no constraint on the base matrix, the proposed method enlarges the applicable range for dealing with the mixed-sign data. Furthermore, the convergence and computational complexity of DCNMF are analyzed, and the relationships between DCNMF and some representative methods are also discussed. Finally, extensive experiments on the mixed-sign and nonnegative datasets have shown the effectiveness of the proposed method as compared to the related NMF based methods in clustering applications.

Although the proposed DCNMF method has shown good performance in clustering tasks, it still has a limitation of high computational load. In the future, to reduce the computational cost and the running time, investigation will be conducted to exploit a theoretically feasible and practically efficient fast optimization algorithm for DCNMF. Furthermore, as an important

dimensionality reduction technique, NMF has been widely used as a preprocessing step for various practical tasks. In the future, applying DCNMF to more concrete scenarios e.g., community detection in social networks will be an interesting work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported in part by the National Nature Science Foundation of China (No. U1701266, 91648208, 61976175), and Guangdong Intellectual Property Big Data Key Laboratory (No. 2018B030322016).

Appendix A. Proof of Theorem 1

In order to prove Theorem 1, we first define an auxiliary function, a relevant lemma, and two propositions.

Definition: $\mathcal{G}(v, v')$ is an auxiliary function for $\mathcal{F}(v)$ if the following conditions

$$\mathcal{G}(v, v') \geqslant \mathcal{F}(v), \quad \mathcal{G}(v, v) = \mathcal{F}(v) \tag{19}$$

are satisfied.

Lemma 1. *If $\mathcal{G}$ is an auxiliary function for F, then $\mathcal{F}$ decreases monotonically under the following update rule from iteration t to $t+1$:*

$$v^{t+1} = \arg\min_{v} \mathcal{G}(v, v^t) \tag{20}$$

Proof.

$$\mathcal{F}(v^{t+1}) \leqslant \mathcal{G}(v^{t+1}, v^t) \leqslant \mathcal{G}(v^t, v^t) = \mathcal{F}(v^t)$$

clearly, if the update rules for minimizing $\mathcal{G}$ are exactly the update rules (14) and (15), then we conclude that $\mathcal{O}_{DC}$ decreases monotonically under the update rules (14) and (15). □

Proposition 1. *For any matrices $\mathbf{G}, \mathbf{G}', \mathbf{H} \in \mathbb{R}_{\geqslant 0}^{N\times K}$, then the following inequalities hold*

$$Tr(\mathbf{G}^T\mathbf{H}) \geqslant \sum_{ik} \mathbf{H}_{ik}\mathbf{G}'_{ik}\left(1 + \log\frac{\mathbf{G}_{ik}}{\mathbf{G}'_{ik}}\right) \tag{21}$$

$$Tr(\mathbf{G}^T\mathbf{H}) \leqslant \sum_{ik} \mathbf{H}_{ik}\frac{\mathbf{G}_{ik}^2 + \mathbf{G}_{ik}'^2}{2\mathbf{G}'_{ik}} \tag{22}$$

Proposition 2. *For any matrices $\mathbf{G}, \mathbf{G}' \in \mathbb{R}_{\geqslant 0}^{N\times K}, \mathbf{H} \in \mathbb{R}_{\geqslant 0}^{N\times N}, \mathbf{J} \in \mathbb{R}_{\geqslant 0}^{K\times K}$, with $\mathbf{H}$ and $\mathbf{J}$ symmetric, then the following inequalities hold:*

$$Tr(\mathbf{G}^T\mathbf{H}\mathbf{G}\mathbf{J}) \geqslant \sum_{ijkl} \mathbf{H}_{ij}\mathbf{G}'_{jk}\mathbf{J}_{kl}\mathbf{G}'_{il}\left(1 + \log\frac{\mathbf{G}_{jk}\mathbf{G}_{il}}{\mathbf{G}'_{jk}\mathbf{G}'_{il}}\right) \tag{23}$$

$$Tr(\mathbf{G}^T\mathbf{H}\mathbf{G}\mathbf{J}) \leqslant \sum_{ik} \frac{(\mathbf{H}\mathbf{G}'\mathbf{J})_{ik}\mathbf{G}_{ik}^2}{\mathbf{G}'_{ik}} \tag{24}$$

The detailed steps for Proposition 1 and 2 have been proved in [16,49]. Hence we omit them in this paper. Next we prove Theorem 1:

Proof. We first prove part 1). When $\mathbf{Z}$ is fixed, the objective function $\mathcal{O}_{DC}$ can be rewritten as follows:

$$\begin{aligned}\mathcal{O}_{DC}(\mathbf{W}) &= Tr\left(-2\mathbf{Z}^T\mathbf{A}^T\mathbf{X}^T\mathbf{X}\mathbf{W} + \mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W}\mathbf{Z}^T\mathbf{A}^T\mathbf{A}\mathbf{Z}\right) = Tr\left(-2\mathbf{V}^T\mathbf{X}^T\mathbf{X}\mathbf{W} + \mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W}\mathbf{V}^T\mathbf{V}\right)\\ &= Tr(-2\mathbf{W}^T\mathbf{M}^+\mathbf{V} + 2\mathbf{W}^T\mathbf{M}^-\mathbf{V} + \mathbf{W}^T\mathbf{M}^+\mathbf{W}\mathbf{V}^T\mathbf{V} - \mathbf{W}^T\mathbf{M}^-\mathbf{W}\mathbf{V}^T\mathbf{V})\end{aligned} \tag{25}$$

Then we define the following lemma:

Lemma 2. *Given the objective function $\mathcal{O}_{DC}(\mathbf{W})$ defined as in (25), where all matrices are nonnegative, the following function*

$$\begin{aligned}\mathcal{G}_1(\mathbf{W},\mathbf{W}') = &-\sum_{ik} 2(\mathbf{M}^+\mathbf{V})_{ik}\mathbf{W}'_{ik}\left(1+\log\frac{\mathbf{W}_{ik}}{\mathbf{W}'_{ik}}\right)+\sum_{ik}2(\mathbf{M}^-\mathbf{V})_{ik}\frac{\mathbf{W}_{ik}^2+\mathbf{W}_{ik}'^2}{2\mathbf{W}'_{ik}}\\ &+\sum_{ik}\frac{(\mathbf{M}^+\mathbf{W}'\mathbf{V}^T\mathbf{V})_{ik}\mathbf{W}_{ik}^2}{\mathbf{W}'_{ik}}-\sum_{ijkl}\mathbf{M}_{ij}^-\mathbf{W}'_{jk}(\mathbf{V}^T\mathbf{V})_{kl}\mathbf{W}'_{il}\left(1+log\frac{\mathbf{W}_{jk}\mathbf{W}_{il}}{\mathbf{W}'_{jk}\mathbf{W}'_{il}}\right)\end{aligned} \tag{26}$$

is an auxiliary function for $\mathcal{O}_{DC}(\mathbf{W})$.

Proof. Evidently, by Propositions 1 and 2, we have

$$Tr(\mathbf{W}^T\mathbf{M}^+\mathbf{V}) \geqslant \sum_{ik}(\mathbf{M}^+\mathbf{V})_{ik}\mathbf{W}'_{ik}\left(1+\log\frac{\mathbf{W}_{ik}}{\mathbf{W}'_{ik}}\right) \tag{27}$$

$$Tr(\mathbf{W}^T\mathbf{M}^-\mathbf{V}) \leqslant \sum_{ik}(\mathbf{M}^-\mathbf{V})_{ik}\frac{\mathbf{W}_{ik}^2+\mathbf{W}_{ik}'^2}{2\mathbf{W}'_{ik}} \tag{28}$$

$$Tr(\mathbf{W}^T\mathbf{M}^+\mathbf{W}\mathbf{V}^T\mathbf{V}) \leqslant \sum_{ik}\frac{(\mathbf{M}^+\mathbf{W}'\mathbf{V}^T\mathbf{V})_{ik}\mathbf{W}_{ik}^2}{\mathbf{W}'_{ik}} \tag{29}$$

$$Tr(\mathbf{W}^T\mathbf{M}^-\mathbf{W}\mathbf{V}^T\mathbf{V}) \geqslant \sum_{ijkl}\mathbf{M}_{ij}^-\mathbf{W}'_{jk}(\mathbf{V}^T\mathbf{V})_{kl}\mathbf{W}'_{il}\left(1+\log\frac{\mathbf{W}_{jk}\mathbf{W}_{il}}{\mathbf{W}'_{jk}\mathbf{W}'_{il}}\right) \tag{30}$$

Based on (27), (28), (29), and (30), we derive that

$$\mathcal{G}_1(\mathbf{W},\mathbf{W}') \geqslant \mathcal{O}_{DC}(\mathbf{W}), \quad \mathcal{G}_1(\mathbf{W},\mathbf{W}) \geqslant \mathcal{O}_{DC}(\mathbf{W})$$

According to Definition 1, $\mathcal{G}_1(\mathbf{W},\mathbf{W}')$ is the auxiliary function for the objective function $\mathcal{O}_{DC}(\mathbf{W})$. Assuming $\mathbf{W}_{ik}$ (or w_{ik}) to be any entry in the matrix $\mathbf{W}$, the first order partial derivatives for $\mathcal{O}_{DC}(\mathbf{W})$ with respect to $\mathbf{W}_{ik}$ is:

$$\frac{\partial\mathcal{G}_1(\mathbf{W},\mathbf{W}')}{\partial\mathbf{W}_{ik}} = -2(\mathbf{M}^+\mathbf{V})_{ik}\frac{\mathbf{W}'_{ik}}{\mathbf{W}_{ik}}+2(\mathbf{M}^-\mathbf{V})_{ik}\frac{\mathbf{W}_{ik}}{\mathbf{W}'_{ik}}+2\frac{(\mathbf{M}^+\mathbf{W}'\mathbf{V}^T\mathbf{V})_{ik}\mathbf{W}_{ik}}{\mathbf{W}'_{ik}}-2\frac{(\mathbf{M}^-\mathbf{W}'\mathbf{V}^T\mathbf{V})_{ik}\mathbf{W}'_{ik}}{\mathbf{W}_{ik}} \tag{31}$$

Since $\mathcal{G}_1(\mathbf{W},\mathbf{W}')$ is convex with respect to $\mathbf{W}$, by $\frac{\partial\mathcal{G}_1(\mathbf{W},\mathbf{W}')}{\partial\mathbf{W}_{ik}}=0$, we have the optimal $\mathbf{W}_{ik}$ for minimizing $\mathcal{G}_1(\mathbf{W},\mathbf{W}')$:

$$\mathbf{W}_{ik} = \mathbf{W}'_{ik}\sqrt{\frac{(\mathbf{M}^+\mathbf{V})_{ik}+(\mathbf{M}^-\mathbf{W}'\mathbf{V}^T\mathbf{V})_{ik}}{(\mathbf{M}^-\mathbf{V})_{ik}+(\mathbf{M}^+\mathbf{W}'\mathbf{V}^T\mathbf{V})_{ik}}}$$

Due to $\mathbf{V}=\mathbf{AZ}$, the update rule for $\mathbf{W}$ is derived as follows:

$$w_{ik} \leftarrow w_{ik}\sqrt{\frac{\left(\mathbf{M}^+\mathbf{AZ}+\mathbf{M}^-\mathbf{W}\mathbf{Z}^T\mathbf{A}^T\mathbf{AZ}\right)_{ik}}{\left(\mathbf{M}^-\mathbf{AZ}+\mathbf{M}^+\mathbf{W}\mathbf{Z}^T\mathbf{A}^T\mathbf{AZ}\right)_{ik}}} \tag{33}$$

which is similar to (14). Based on Lemma 1, we conclude that, when fixing $\mathbf{Z}$, the objective function $\mathcal{O}_{DC}$ decreases monotonically under the update rule (14). The proof for the part 1) of Theorem 1 is completed. For the part 2), when $\mathbf{W}$ is fixed, and the constant terms nonrelated to $\mathbf{W}$ are discarded in $\mathcal{O}_{DC}$, we can rewrite the objective function $\mathcal{O}_{DC}$ as follows:

$$\begin{aligned}\mathcal{O}_{DC}(\mathbf{Z}) &= Tr\left(-2\mathbf{Z}^T\mathbf{A}^T\mathbf{X}^T\mathbf{X}\mathbf{W}+\mathbf{Z}^T\mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W}\right)+\beta Tr(\mathbf{Z}^T\mathbf{A}^T\widetilde{L}\mathbf{AZ})\\ &= Tr(-2\mathbf{Z}^T\mathbf{A}^T\mathbf{M}^+\mathbf{W}+2\mathbf{Z}^T\mathbf{A}^T\mathbf{M}^-\mathbf{W}+\mathbf{Z}^T\mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{M}^+\mathbf{W}-\mathbf{Z}^T\mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{M}^-\mathbf{W})\\ &\quad+\beta Tr(\mathbf{Z}^T\mathbf{A}^T\widetilde{\mathbf{D}}\mathbf{AZ})-\beta Tr(\mathbf{Z}^T\mathbf{A}^T\widetilde{\mathbf{S}}\mathbf{AZ})\end{aligned} \tag{34}$$

Then Lemma 3 is defined as follows:

Lemma 3. Given the objective function $\mathcal{O}_{DC}(\mathbf{Z})$ defined as in (34), where all matrices are nonnegative, the following function

$$\begin{aligned}\mathcal{G}_2(\mathbf{Z},\mathbf{Z}') = &-\sum_{jk}2(\mathbf{A}^T\mathbf{M}^+\mathbf{W})_{jk}\mathbf{Z}'_{jk}\left(1+\log\frac{\mathbf{Z}_{jk}}{\mathbf{Z}'_{jk}}\right)+\sum_{jk}2(\mathbf{A}^T\mathbf{M}^-\mathbf{W})_{jk}\frac{\mathbf{Z}_{jk}^2+\mathbf{Z}_{jk}'^2}{2\mathbf{Z}'_{jk}}+\sum_{jk}\frac{(\mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{M}^+\mathbf{W})_{jk}\mathbf{Z}_{jk}^2}{\mathbf{Z}'_{jk}}\\&-\sum_{ijkl}(\mathbf{A}^T\mathbf{A})_{ij}\mathbf{Z}'_{jk}(\mathbf{W}^T\mathbf{M}^-\mathbf{W})_{kl}\mathbf{Z}'_{il}\left(1+\log\frac{\mathbf{Z}_{jk}\mathbf{Z}_{il}}{\mathbf{Z}'_{jk}\mathbf{Z}'_{il}}\right)\\&+\beta\sum_{jk}\frac{(\mathbf{A}^T\tilde{\mathbf{D}}\mathbf{A}\mathbf{Z})_{jk}\mathbf{Z}_{jk}^2}{\mathbf{Z}'_{jk}}-\beta\sum_{ijk}(\mathbf{A}^T\tilde{\mathbf{S}}\mathbf{A})_{ij}\mathbf{Z}'_{jk}\mathbf{Z}\prime_{ik}\left(1+\log\frac{\mathbf{Z}_{jk}\mathbf{Z}_{ik}}{\mathbf{Z}'_{jk}\mathbf{Z}'_{ik}}\right)\end{aligned}\tag{35}$$

is an auxiliary function for $\mathcal{O}_{DC}(\mathbf{Z})$.

Proof. Similarly, based on Propositions 1, we derive

$$Tr(\mathbf{Z}^T\mathbf{A}^T\mathbf{M}^+\mathbf{W})\geqslant\sum_{jk}(\mathbf{A}^T\mathbf{M}^+\mathbf{W})_{jk}\mathbf{Z}'_{jk}\left(1+\log\frac{\mathbf{Z}_{jk}}{\mathbf{Z}'_{jk}}\right)\tag{36}$$

$$Tr(\mathbf{Z}^T\mathbf{A}^T\mathbf{M}^-\mathbf{W})\leqslant\sum_{jk}(\mathbf{A}^T\mathbf{M}^-\mathbf{W})_{jk}\frac{\mathbf{Z}_{jk}^2+\mathbf{Z}_{jk}'^2}{2\mathbf{Z}'_{jk}}\tag{37}$$

Furthermore, by using Propositions 2 and $\beta>0$, we have

$$Tr(\mathbf{Z}^T\mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{M}^+\mathbf{W})\leqslant\sum_{jk}\frac{(\mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{M}^+\mathbf{W})_{jk}\mathbf{Z}_{jk}^2}{\mathbf{Z}'_{jk}}\tag{38}$$

$$Tr(\mathbf{Z}^T\mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{M}^-\mathbf{W})\geqslant\sum_{ijkl}(\mathbf{A}^T\mathbf{A})_{ij}\mathbf{Z}'_{jk}(\mathbf{W}^T\mathbf{M}^-\mathbf{W})_{kl}\mathbf{Z}'_{il}\left(1+\log\frac{\mathbf{Z}_{jk}\mathbf{Z}_{il}}{\mathbf{Z}'_{jk}\mathbf{Z}'_{il}}\right)\tag{39}$$

$$Tr(\mathbf{Z}^T\mathbf{A}^T\tilde{\mathbf{D}}\mathbf{A}\mathbf{Z})\leqslant\sum_{jk}\frac{(\mathbf{A}^T\tilde{\mathbf{D}}\mathbf{A}\mathbf{Z})_{jk}\mathbf{Z}_{jk}^2}{\mathbf{Z}'_{jk}}\tag{40}$$

$$Tr(\mathbf{Z}^T\mathbf{A}^T\tilde{\mathbf{S}}\mathbf{A}\mathbf{Z})\geqslant\sum_{ijk}(\mathbf{A}^T\tilde{\mathbf{S}}\mathbf{A})_{ij}\mathbf{Z}'_{jk}\mathbf{Z}'_{ik}\left(1+\log\frac{\mathbf{Z}_{jk}\mathbf{Z}_{ik}}{\mathbf{Z}'_{jk}\mathbf{Z}'_{ik}}\right)\tag{41}$$

Based on (36), (37), (38), (39), (40), and (41), we easily derive that

$$\mathcal{G}_2(\mathbf{Z},\mathbf{Z}')\geqslant\mathcal{O}_{DC}(\mathbf{Z}),\quad\mathcal{G}_2(\mathbf{Z},\mathbf{Z})\geqslant\mathcal{O}_{DC}(\mathbf{Z})$$

which proves $\mathcal{G}_2(\mathbf{Z},\mathbf{Z}')$ is the auxiliary function for the objective function $\mathcal{O}_{DC}(\mathbf{Z})$. Assuming $\mathbf{Z}_{jk}$ (or z_{jk}) to be any entry in the matrix $\mathbf{Z}$, the first order partial derivatives for $\mathcal{O}_{DC}(\mathbf{Z})$ with respect to $\mathbf{Z}_{jk}$ is:

$$\begin{aligned}\frac{\partial\mathcal{G}_1(\mathbf{Z},\mathbf{Z}')}{\partial\mathbf{Z}_{jk}}=&-2(\mathbf{A}^T\mathbf{M}^+\mathbf{W})_{jk}\frac{\mathbf{Z}'_{jk}}{\mathbf{Z}_{jk}}+2(\mathbf{A}^T\mathbf{M}^-\mathbf{W})_{jk}\frac{\mathbf{Z}_{jk}}{\mathbf{Z}'_{jk}}+2\frac{(\mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{M}^+\mathbf{W})_{jk}\mathbf{Z}_{jk}}{\mathbf{Z}'_{jk}}\\&-2\frac{(\mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{M}^-\mathbf{W})_{jk}\mathbf{Z}'_{jk}}{\mathbf{Z}_{jk}}+2\beta\frac{(\mathbf{A}^T\tilde{\mathbf{D}}\mathbf{A}\mathbf{Z})_{jk}\mathbf{Z}_{jk}}{\mathbf{Z}'_{jk}}-2\beta\frac{(\mathbf{A}^T\tilde{\mathbf{S}}\mathbf{A}\mathbf{Z})_{jk}\mathbf{Z}'_{jk}}{\mathbf{Z}_{jk}}\end{aligned}\tag{42}$$

Due to the fact that $\mathcal{G}_2(\mathbf{Z},\mathbf{Z}')$ is convex with respect to $\mathbf{Z}$, we obtain the optimal $\mathbf{Z}_{jk}$ by $\frac{\partial\mathcal{G}_2(\mathbf{Z},\mathbf{Z}')}{\partial\mathbf{Z}_{jk}}=0$,:

$$\mathbf{Z}_{jk}\leftarrow\mathbf{Z}'_{jk}\sqrt{\frac{\left(\mathbf{A}^T\mathbf{M}^+\mathbf{W}+\mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{M}^-\mathbf{W}+\beta\mathbf{A}^T\tilde{\mathbf{S}}\mathbf{A}\mathbf{Z}\right)_{jk}}{\left(\mathbf{A}^T\mathbf{M}^-\mathbf{W}+\mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{M}^+\mathbf{W}+\beta\mathbf{A}^T\tilde{\mathbf{D}}\mathbf{A}\mathbf{Z}\right)_{jk}}}\tag{43}$$

Therefore the update rule for $\mathbf{Z}$ is:

$$z_{jk} \leftarrow z_{jk} \sqrt{\frac{\left(\mathbf{A}^T\mathbf{M}^+\mathbf{W} + \mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{M}^-\mathbf{W} + \beta\mathbf{A}^T\tilde{\mathbf{S}}\mathbf{A}\mathbf{Z}\right)_{jk}}{\left(\mathbf{A}^T\mathbf{M}^-\mathbf{W} + \mathbf{A}^T\mathbf{A}\mathbf{Z}\mathbf{W}^T\mathbf{M}^+\mathbf{W} + \beta\mathbf{A}^T\tilde{\mathbf{D}}\mathbf{A}\mathbf{Z})\right)_{jk}}} \tag{44}$$

which is similar to (15). Based on Lemma 1, we conclude that, when fixing $\mathbf{W}$, the objective function $\mathcal{O}_{DC}$ decreases monotonically under the update rule (15). The proof for the part 2) of Theorem 1 is completed. □

References

[1] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural Comput. 15 (6) (2003) 1373–1396.
[2] L. De Lathauwer, B. De Moor, J. Vandewalle, A multilinear singular value decomposition, SIAM J. Matrix Anal. Appl. 21 (4) (2000) 1253–1278.
[3] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[4] N. Zhou, H. Cheng, J. Qin, Y. Du, B. Chen, Robust high-order manifold constrained sparse principal component analysis for image representation, IEEE Trans. Circuits Syst. Video Technol. 29 (7) (2018) 1946–1961.
[5] A. Hyvärinen, E. Oja, Independent component analysis: algorithms and applications, Neural Networks 13 (4–5) (2000) 411–430.
[6] D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (6755) (1999) 788–791.
[7] D.D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, in: Advances in Neural Information Processing Systems, 2001, pp. 556–562..
[8] C. Peng, Z. Zhang, Z. Kang, C. Chen, Q. Cheng, Nonnegative matrix factorization with local similarity learning, Inf. Sci. 562 (2021) 325–346.
[9] W. Wu, Y. Jia, S. Kwong, J. Hou, Pairwise constraint propagation-induced symmetric nonnegative matrix factorization, IEEE Trans. Neural Networks Learn. Syst. 29 (12) (2018) 6348–6361.
[10] W. Xu, X. Liu, Y. Gong, Document clustering based on non-negative matrix factorization, in: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, 2003, pp. 267–273.
[11] W. Wu, S. Kwong, J. Hou, Y. Jia, H.H.S. Ip, Simultaneous dimensionality reduction and classification via dual embedding regularized nonnegative matrix factorization, IEEE Trans. Image Process. 28 (8) (2019) 3836–3847.
[12] C. Ding, X. He, H.D. Simon, On the equivalence of nonnegative matrix factorization and spectral clustering, in, in: Proceedings of the 2005 SIAM international conference on data mining, SIAM, 2005, pp. 606–610.
[13] Y.-X. Wang, Y.-J. Zhang, Nonnegative matrix factorization: A comprehensive review, IEEE Trans. Knowl. Data Eng. 25 (6) (2012) 1336–1353.
[14] D. Cai, X. He, J. Han, T.S. Huang, Graph regularized nonnegative matrix factorization for data representation, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2010) 1548–1560.
[15] Y. Yi, J. Wang, W. Zhou, C. Zheng, J. Kong, S. Qiao, Non-negative matrix factorization with locality constrained adaptive graph, IEEE Trans. Circuits Syst. Video Technol. 30 (2) (2019) 427–441.
[16] C.H. Ding, T. Li, M.I. Jordan, Convex and semi-nonnegative matrix factorizations, IEEE Trans. Pattern Anal. Mach. Intell. 32 (1) (2008) 45–55.
[17] J. Ye, Z. Jin, Dual-graph regularized concept factorization for clustering, Neurocomputing 138 (2014) 120–130.
[18] W. Hu, K.-S. Choi, P. Wang, Y. Jiang, S. Wang, Convex nonnegative matrix factorization with manifold regularization, Neural Networks 63 (2015) 94–103.
[19] H. Liu, Z. Wu, X. Li, D. Cai, T.S. Huang, Constrained nonnegative matrix factorization for image representation, IEEE Trans. Pattern Anal. Mach. Intell. 34 (7) (2011) 1299–1311.
[20] D. Wang, X. Gao, X. Wang, Semi-supervised nonnegative matrix factorization via constraint propagation, IEEE Trans. Cybern. 46 (1) (2015) 233–244.
[21] Y. Jia, S. Kwong, J. Hou, W. Wu, Semi-supervised non-negative matrix factorization with dissimilarity and similarity regularization, IEEE Trans. Neural Networks Learn. Syst. 31 (7) (2019) 2510–2521.
[22] X. Zhang, L. Zong, X. Liu, J. Luo, Constrained clustering with nonnegative matrix factorization, IEEE Trans. Neural Networks Learn. Syst. 27 (7) (2015) 1514–1526.
[23] S. Peng, W. Ser, B. Chen, Z. Lin, Robust semi-supervised nonnegative matrix factorization for image clustering, Pattern Recogn. 111 (2021) 107683.
[24] Y. Jia, H. Liu, J. Hou, S. Kwong, Semisupervised adaptive symmetric non-negative matrix factorization, IEEE Trans. Cybern. (2020).
[25] S. Wang, A. Huang, Penalized nonnegative matrix tri-factorization for co-clustering, Expert Syst. Appl. 78 (2017) 64–73.
[26] N. Zhou, Y. Xu, H. Cheng, Z. Yuan, B. Chen, Maximum correntropy criterion-based sparse subspace learning for unsupervised feature selection, IEEE Trans. Circuits Syst. Video Technol. 29 (2) (2017) 404–417.
[27] P. He, X. Xu, J. Ding, B. Fan, Low-rank nonnegative matrix factorization on Stiefel manifold, Inf. Sci. 514 (2020) 131–148.
[28] X. Peng, D. Chen, D. Xu, Hyperplane-based nonnegative matrix factorization with label information, Inf. Sci. 493 (2019) 1–19.
[29] H. Li, J. Zhang, G. Shi, J. Liu, Graph-based discriminative nonnegative matrix factorization with label information, Neurocomputing 266 (2017) 91–100.
[30] H. Li, J. Zhang, J. Hu, C. Zhang, J. Liu, Graph-based discriminative concept factorization for data representation, Knowl.-Based Syst. 118 (2017) 70–79.
[31] M. Lu, L. Zhang, X.-J. Zhao, F.-Z. Li, Constrained neighborhood preserving concept factorization for data representation, Knowl.-Based Syst. 102 (2016) 127–139.
[32] Z. Li, J. Tang, X. He, Robust structured nonnegative matrix factorization for image representation, IEEE Trans. Neural Networks Learn. Syst. 29 (5) (2017) 1947–1960.
[33] Y. Meng, R. Shang, L. Jiao, W. Zhang, S. Yang, Dual-graph regularized non-negative matrix factorization with sparse and orthogonal constraints, Eng. Appl. Artif. Intell. 69 (2018) 24–35.
[34] P. Luo, J. Peng, Group sparsity and graph regularized semi-nonnegative matrix factorization with discriminability for data representation, Entropy 19 (12) (2017) 627.
[35] K. Allab, L. Labiod, M. Nadif, A semi-nmf-pca unified framework for data clustering, IEEE Trans. Knowl. Data Eng. 29 (1) (2016) 2–16.
[36] G. Li, X. Zhang, S. Zheng, D. Li, Semi-supervised convex nonnegative matrix factorizations with graph regularized for image representation, Neurocomputing 237 (2017) 1–11.
[37] H. Cai, B. Liu, Y. Xiao, L. Lin, Semi-supervised multi-view clustering based on orthonormality-constrained nonnegative matrix factorization, Inf. Sci. 536 (2020) 171–184.
[38] Y. Jia, H. Liu, J. Hou, S. Kwong, Pairwise constraint propagation with dual adversarial manifold regularization, IEEE Trans. Neural Networks Learn. Syst. 31 (12) (2020) 5575–5587.
[39] Z. Lu, H.H. Ip, Constrained spectral clustering via exhaustive and efficient constraint propagation, in, European Conference on Computer Vision, Springer (2010) 1–14.
[40] S. Peng, W. Ser, B. Chen, L. Sun, Z. Lin, Correntropy based graph regularized concept factorization for clustering, Neurocomputing 316 (2018) 34–48.
[41] D. Cai, X. He, J. Han, Locally consistent concept factorization for document clustering, IEEE Trans. Knowl. Data Eng. 23 (6) (2010) 902–913.
[42] W. Xu, Y. Gong, Document clustering by concept factorization, in: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, 2004, pp. 202–209.
[43] W. Zhang, T. Yoshida, X. Tang, A comparative study of tf* idf, lsi and multi-words for text classification, Expert Syst. Appl. 38 (3) (2011) 2758–2765.

[44] X. Zhang, H. Gao, G. Li, J. Zhao, J. Huo, J. Yin, Y. Liu, L. Zheng, Multi-view clustering based on graph-regularized nonnegative matrix factorization for object recognition, Inf. Sci. 432 (2018) 463–478.
[45] X. Peng, D. Xu, D. Chen, Robust distribution-based nonnegative matrix factorizations for dimensionality reduction, Inf. Sci. 552 (2021) 244–260.
[46] L. Lovász, M.D. Plummer, Matching theory, vol. 367, American Mathematical Soc., 2009..
[47] C.C. Aggarwal, C.K. Reddy, Data clustering, Algorithms and applications. Chapman&Hall/CRC Data mining and Knowledge Discovery series, Londra, 2014..
[48] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[49] Z. Yang, E. Oja, Linear and nonlinear projective nonnegative matrix factorization, IEEE Trans. Neural Networks 21 (5) (2010) 734–749.