https://dx.doi.org/10.1016/j.neunet.2014.11.007

Convex Nonnegative Matrix Factorization with Manifold Regularization

Wenjun Hu [a,b,c,*], Kup-Sze Choi [c], Peiliang Wang [a], Yunliang Jiang [a], Shitong Wang [b]

[a] School of Information Engineering, Huzhou University, Huzhou, Zhejiang, China

[b] School of Digital Media, Jiangnan University, Wuxi, Jiangsu, China

[c] Centre for Smart Heath, School of Nursing, Hong Kong Polytechnic University, Hong Kong, China

Abstract: Nonnegative Matrix Factorization (NMF) has been extensively applied in many areas, including computer vision, pattern recognition, text mining, and signal processing. However, nonnegative entries are usually required for the data matrix in NMF, which limits its application. Besides, while the basis and encoding vectors obtained by NMF can represent the original data in low dimension, the representations do not always reflect the intrinsic geometric structure embedded in the data. Motivated by manifold learning and Convex NMF (CNMF), we propose a novel matrix factorization method called Graph Regularized and Convex Nonnegative Matrix Factorization (GCNMF) by introducing a graph regularization term into CNMF. The proposed matrix factorization technique not only inherits the intrinsic low-dimensional manifold structure, but also allows the processing of mixed-sign data matrix. Clustering experiments on nonnegative and mixed-sign real-world data sets are conducted to demonstrate the effectiveness of the proposed method.

Index Terms—Nonnegative matrix factorization, manifold regularization, convex nonnegative matrix factorization, clustering.

1 INTRODUCTION

Nonnegative Matrix Factorization (NMF) is a popular matrix factorization technique which decomposes a data matrix into the product of two matrices with nonnegative entries [1], [2]. It is a NP-hard problem [5] and was first proposed by Paatero and Tapper [1]. NMF possesses powerful representation of the data and finds many applications, including face and object recognition [14], [15], biomedical applications [16], text mining [17], [18], [19], brain electromagnetic tomography applications [20] and speech signal processing [21].

As the entries are constrained to be nonnegative, NMF is usually interpreted as a *parts-based representation* of the data that only allows additive combinations while prohibiting subtractive combinations. This is a feature that makes NMF distinct from other matrix factorization methods like Singular Value Decomposition (SVD), Principal Component Analysis (PCA) and Independent Component Analysis (ICA) [3], [4].

Multiplicative iterative rules have been developed to solve the NMF problem by considering it as a non-convex programming problem and applying heuristic procedures [2], [6]. Unless convergence to a stationary point can be

[*] Corresponding author. School of Information Engineering, Huzhou University, Huzhou 313000, Zhejiang, China. Tel.: +86 572 2321106.
E-mail address: hoowenjun@foxmail.com (Wenjun Hu).

achieved, the multiplicative iterative rules indeed do not guarantee optimality. Recently, some methods have been presented to solve the NMF problem exactly based on the assumption of separability [7], [8], [9], [10], where regularized terms and/or constraints are introduced to the cost function to develop different variations of NMF. For example, Feng et al. [11] proposed the Local Nonnegative Matrix Factorization (LNMF) by applying subspace method and feature localization to obtain a part-based representation and manifest localized features. Cai et al. proposed the Graph Regularized Nonnegative Matrix Factorization (GNMF) [12] that takes into account the geometrically-based regularizer to determine the low-dimension manifold structure of the data. Smoothing of the encoding vectors is applied to increase the sparseness of the basis vectors. However, these NMF algorithms are only applicable to nonnegative data matrices and the interpretability of the based-parts presentation are weak. To deal with mixed-sign data, semi-NMF, convex-NMF, and cluster-NMF algorithms have been proposed [13]. In particular, convex-NMF algorithms (CNMF) further require that the basis vectors in NMF are convex or linear combinations of the data points. As a result, the basis vectors can better capture the cluster centroids and ensure the sparseness of the encoding vectors. Nevertheless, these methods ignore the importance to preserve the low-dimension manifold in part-based representation, i.e. smoothing of encoding vectors.

Motivated by manifold learning and CNMF [13], we introduce a graph regularized term into CNMF and propose a novel matrix factorization method called Graph Regularized and Convex Nonnegative Matrix Factorization (GCNMF). The proposed approach combines the manifold structure with CNMF so that the encoding vectors obtained by matrix factorization can preserve the low-dimension manifold structure. Like many manifold learning algorithms, the idea of local structure invariant is also employed to reveal the intrinsic manifold structure. Besides, GCNMF can also handle mixed-sign matrix, which extends the application of NMF. As a result, the structure of the data can be interpreted more properly by using GCNMF during the process of matrix factorization and the performance can be guaranteed for nonnegative and mixed-sign data sets.

The rest of the paper is organized as follows. Section 2 provides a brief description of the work related to the proposed GCNMF algorithm. Section 3 introduces the GCNMF algorithm and discusses the solving scheme. Section 4 presents the clustering experiments performed on nonnegative and mixed-sign data sets, which are used to evaluate the performance of the proposed algorithm. Finally, a conclusion is given in Section 5.

2 RELATED WORK

The proposed GCNMF algorithm is closely related to NMF [1], [2], GNMF [12] and CNMF [13]. These matrix factorization techniques are briefly described in this section.

2.1 NMF

NMF is a common matrix factorization technique in numerical linear algebra. It decomposes a data matrix into a product of two matrices whose elements are nonnegative. Let $\mathbf{X}=[\mathbf{x}_1,\cdots,\mathbf{x}_N]$ be a matrix with column vectors $\mathbf{x}_i\in\Re^D$. Then, the NMF algorithm can be expressed as

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

where $\mathbf{U}=[u_{ik}]\in\Re^{D\times K}$ and $\mathbf{V}=[v_{jk}]\in\Re^{N\times K}$ are two matrices with nonnegative entries. The column vectors of $\mathbf{U}$ are called the basis vectors and the column vectors of $\mathbf{V}$ are called the encoding vectors. To measure the quality of NMF in Eq. (1), Paatero et al. proposed two mechanisms based on the measurement of the Euclidean distance and the divergence distance respectively [1]. In this paper, we focus on the former and the corresponding objective function can be formulated as

$$O_1(\mathbf{U},\mathbf{V})=\left\|\mathbf{X}-\mathbf{U}\mathbf{V}^T\right\|^2, \tag{2}$$

where $\|\bullet\|$ denotes the Frobenius norm of a matrix. To minimize the objective function in Eq. (2), Lee and Seung [22] proposed a multiplicative update algorithm, which is given by

$$u_{ik}\leftarrow u_{ik}\frac{(\mathbf{X}\mathbf{V})_{ik}}{(\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik}} \tag{3}$$

and

$$v_{jk}\leftarrow v_{jk}\frac{(\mathbf{X}^T\mathbf{U})_{jk}}{(\mathbf{V}\mathbf{U}^T\mathbf{U})_{jk}}. \tag{4}$$

2.2 GNMF

The GNMF algorithm is developed by combining a geometrically based regularized term with NMF [12]. Here, the approximation in the NMF algorithm in Eq. (1) is considered with the column-wise representation below,

$$\mathbf{x}_j\approx\sum_{k=1}^{K}\mathbf{u}_k v_{jk}, \tag{5}$$

where $\mathbf{u}_k$ is the kth column vector of $\mathbf{U}$. Clearly, the linear combination of the basis vectors and the entries of $\mathbf{V}$ can be used to approximate each data $\mathbf{x}_j$. This implies that $v_{jk},\cdots,v_{jK}$ are the coordinates with respect to the basis $\mathbf{U}$. In other words, we can define a vector $\mathbf{z}_j=[v_{j1},\cdots,v_{jK}]^T$ to represent the original data $\mathbf{x}_j$ under the basis $\mathbf{U}$. A regularized term is introduced into the learning process of NMF to inherit and preserve the underlying manifold structure of the data space in which $\mathbf{X}$ is sampled. The objective function of GNMF [12] can then be expressed as

$$O_2(\mathbf{U},\mathbf{V}) = \left\| \mathbf{X} - \mathbf{U}\mathbf{V}^T \right\|^2 + \lambda \frac{1}{2} \sum_{i,j=1}^{N} \left\| \mathbf{z}_i - \mathbf{z}_j \right\|^2 \mathbf{W}_{ij}, \tag{6}$$

where $\mathbf{W}_{ij}$ is the *ij*th entry of the weight matrix $\mathbf{W}$ and constitutes of the adjacency graph [12], [23], and $\lambda \geq 0$ is a control parameter. The weight matrix $\mathbf{W}$ can take many different forms and two common definitions are given. Let $N(\mathbf{x}_i)$ denote a set of p nearest neighbors of $\mathbf{x}_i$. One of the definitions of $\mathbf{W}$ is the 0-1 weights, which is given by

$$\mathbf{W}_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_j \in N(\mathbf{x}_i) \text{ or } \mathbf{x}_i \in N(\mathbf{x}_j) \\ 0, & \text{otherwise} \end{cases}.$$

The other is the heat kernel weights, which is expressed as

$$\mathbf{W}_{ij} = \begin{cases} \exp(-\frac{\left\| \mathbf{x}_i - \mathbf{x}_j \right\|}{2\sigma^2}), & \text{if } \mathbf{x}_j \in N(\mathbf{x}_i) \text{ or } \mathbf{x}_i \in N(\mathbf{x}_j) \\ 0, & \text{otherwise} \end{cases},$$

where σ is the heat kernel parameter, a constant value. In Eq. (6), the first term on the right-hand side is to increase the accuracy of the approximation of $\mathbf{X}$ by $\mathbf{U}\mathbf{V}^T$. The second term, involving the coordinates with respect to $\mathbf{U}$, is to preserve the manifold structure of the data space. That is, if $\mathbf{x}_i$ and $\mathbf{x}_j$ are close, then $\mathbf{z}_i$ and $\mathbf{z}_j$ will also be close to each other. The GNMF algorithm exhibits good performance in the clustering of image, face and document data sets. Further details of GNMF can be found in [12].

2.3 Convex NMF

NMF and GNMF can only be applied to the factorization of nonnegative data matrix. For better interpretability, Ding et al. proposed the CNMF [13] that the basis vectors are convex or linear combinations of the data points, i.e.

$$\mathbf{X} \approx \tilde{\mathbf{U}}\mathbf{V}^T = \mathbf{X}\mathbf{U}\mathbf{V}^T, \tag{7}$$

where both $\mathbf{U} \in \Re^{N \times K}$ and $\mathbf{V} \in \Re^{N \times K}$ are nonnegative, the column vectors of $\tilde{\mathbf{U}} = \mathbf{X}\mathbf{U}$ are the basis vectors, and the column vectors of $\mathbf{V}$ are the encoding vectors. The restriction of nonnegative data matrix is thus removed in CNMF, making it applicable for both nonnegative and mixed-sign data matrix. Since the basis vectors are restricted to be within the column space of the data matrix, they can better capture the centroids of the cluster. As a result, CNMF has good interpretability of the data.

2.4 Sparseness in NMF

The study of the sparseness is a hot topic in NMF. Whether the basis vectors or the encoding vectors, or both, should be sparse, is dependent on the application of interest. The sparseness in NMF has been widely investigated. Sparse NMF (SNMF) can be performed using explicit methods or implicit methods. The explicit SNMF methods intuitively impose sparseness constraint on the basis and/or encoding vectors. Many SNMF methods are developed based on different sparseness measures. For example, Kim et al. [36] proposed SNMF/L (SNMF/R) by imposing the L_1-norm constraint on the basis and the encoding vectors. Liu et al. [37] minimized the sum of all elements in the encoding vectors to achieve sparseness in NMF. Hoyer [27] developed the Nonnegative Matrix Factorization with Sparseness Constraints (NMFSC) by proposing a new sparseness measure based on the relationship between the L_1-norm and the L_2-norm of the basis vector or the encoding vector (i.e. Eq. (24) in this paper). Besides, Tandon et al. [38] defined a mixed-norm, i.e. $L_{p,q}$-norm, of the basis vectors which was added to the NMF model. Since p and q can take many different values, SNMF models with different sparseness constraints are developed.

For the implicit SNMF methods, the sparseness of the basis vectors and/or the encoding vectors is inherent in the NMF model. For example, in GNMF, the low-dimensional representation points with respect to the basis vectors, i.e. the rows of $\mathbf{V}$, inherit and preserve the underlying manifold structure of the data by introducing the regularized term. The manifold is smooth and thus ensures the sparseness of the basis vectors. Besides, in CNMF, the basis is restricted in the column space of the data matrix, i.e. $\tilde{\mathbf{U}} = \mathbf{XU}$, which enables each basis vector to capture the centroid of the corresponding cluster. Theoretically, the basis vectors should only be linearly combined with the data from the same cluster. That is, the sparseness of the factor $\mathbf{U}$ will be strengthened. Similarly, as the rows of $\mathbf{V}$ are the low-dimensional representation under the cluster centroids, their sparseness will also be strengthened. Note that excessively strengthening the sparseness of the basis and/or encoding vectors may lead to poor performance in the some applications.

3 Convex Nonnegative Matrix Factorization with Manifold Regularization

In this section, the proposed GCNMF method is presented. The method is motivated by CNMF and GNMF, where a graph regularized term is integrated into CNMF to make it applicable for both nonnegative and mixed-sign data matrix, and to reveal the inherited manifold structure.

3.1 GCNMF

Similar to GNMF, the approximation in the CNMF algorithm in Eq. (7) can be represented in column-wise manner as follows,

$$\mathbf{x}_j \approx \sum_{k=1}^{K} \tilde{\mathbf{u}}_k v_{jk} \ , \tag{8}$$

where $\tilde{\mathbf{u}}_k$ is the kth column vector of $\tilde{\mathbf{U}}$. Let $\mathbf{z}_j = [v_{j1}, \cdots, v_{jK}]^T$ $(1 \le j \le N)$ such that the vector $\mathbf{z}_j$ is the low-dimensional representation of the original data $\mathbf{x}_j$ with respect to the basis $\tilde{\mathbf{U}}$. Given an adjacency graph with the weight matrix $\mathbf{W}$, the smoothness of the low-dimensional representation can be measured by using the term $R(\mathbf{V})$ below,

$$\begin{aligned} R(\mathbf{V}) &= \frac{1}{2}\sum_{i,j=1}^{N} \left\| \mathbf{z}_i - \mathbf{z}_j \right\|^2 \mathbf{W}_{ij} \\ &= \sum_{i=1}^{N} \mathbf{z}_i^T \mathbf{z}_i \mathbf{D}_{ii} - \sum_{i,j=1}^{N} \mathbf{z}_i^T \mathbf{z}_j \mathbf{W}_{ij} \quad , \\ &= \mathrm{Tr}(\mathbf{V}^T \mathbf{D} \mathbf{V}) - \mathrm{Tr}(\mathbf{V}^T \mathbf{W} \mathbf{V}) \\ &= \mathrm{Tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) \end{aligned} \tag{9}$$

where $\mathrm{Tr}(\bullet)$ denotes the trace of a matrix, $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the Laplacian matrix [23], [24], [25], [26], and $\mathbf{D}$ is a diagonal matrix whose entries along the diagonal are the column sum of $\mathbf{W}$, i.e. $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$.

We can then obtain the objective function of the proposed GCNMF as follows,

$$O_3(\mathbf{U}, \mathbf{V}) = \left\| \mathbf{X} - \mathbf{X}\mathbf{U}\mathbf{V}^T \right\|^2 + \lambda R(\mathbf{V}), \tag{10}$$

where the parameter $\lambda \ge 0$ is used to control the smoothness of the low-dimensional representation. Before introducing an iterative algorithm to minimize the objective function in Eq. (10), the features of NMF, GNMF, CNMF and GCNMF is summarized with Table 1 and the differences are discussed.

Table 1 The features of NMF, GNMF, CNMF and GCNMF

Factorization method	Factorization form	Objective function for solution
NMF	$\mathbf{X}_+ \approx \mathbf{U}_+ \mathbf{V}_+^T$	$\min \left\| \mathbf{X}_+ - \mathbf{U}_+ \mathbf{V}_+^T \right\|^2$
GNMF	$\mathbf{X}_+ \approx \mathbf{U}_+ \mathbf{V}_+^T$	$\min\left(\left\| \mathbf{X}_+ - \mathbf{U}_+ \mathbf{V}_+^T \right\|^2 + \lambda \mathrm{Tr}(\mathbf{V}_+^T \mathbf{L} \mathbf{V}_+) \right)$
CNMF	$\mathbf{X}_\pm \approx \mathbf{X}_\pm \mathbf{U}_+ \mathbf{V}_+^T$	$\min \left\| \mathbf{X}_\pm - \mathbf{X}_\pm \mathbf{U}_+ \mathbf{V}_+^T \right\|^2$
GCNMF	$\mathbf{X}_\pm \approx \mathbf{X}_\pm \mathbf{U}_+ \mathbf{V}_+^T$	$\min\left(\left\| \mathbf{X}_\pm - \mathbf{X}_\pm \mathbf{U}_+ \mathbf{V}_+^T \right\|^2 + \lambda \mathrm{Tr}(\mathbf{V}_+^T \mathbf{L} \mathbf{V}_+) \right)$

The subscripts in Table 1 indicate whether entries of a matrix are of mixed sign (±) or nonnegative (+). The data matrix in NMF and GCNMF is restricted to be nonnegative, while CNMF and GCNMF can be applied to mixed-sign data. In GNMF and GCNMF, the low-dimensional representation, i.e. the row vectors of $\mathbf{V}_+$, detects the underlying manifold structure of the original data by preserving the adjacency relationship. From the view of probability distribution, the underlying manifold in the sampling data matrix comprises of multiple sub-manifolds corresponding to different clusters. For the sampling data, it is indeed difficult to specify the underlying manifold and the sub-manifolds. In GNMF, the approach to preserve the adjacency relation may be sufficient for detecting the underlying manifold but inadequate for releasing the information between the sub-manifolds. On the other hand, the basis vectors (the column vectors of $\mathbf{X}_{\pm}\mathbf{U}_{+}$) in CNMF are restricted to be convex combinations of the column vectors of $\mathbf{X}_{\pm}$ so that they can capture the cluster centroids. As different centroids correspond to different sub-manifolds, CNMF can release the structure information between the sub-manifolds to some extent. In the proposed GCNMF, the underlying manifold and the sub-manifolds are considered simultaneously, which is expected to improve the performance of matrix factorization.

As NMF approximates the data matrix with the product of two matrices, i.e. basis matrix multiplied by encoding matrix, strengthening the sparseness of one of these two matrices can lead to the smoothing of the other. On the other hand, strengthening the sparseness or the smoothing of two matrices can cause poor approximation, or deteriorate the goodness-of-fit of the model for the data. For GNMF and CNMF, it is obvious that overfitting of the model for the data matrix can occur since either sparseness or smoothness is only considered. In the proposed GCNMF, each basis vector is a linear combination of the column vectors of $\mathbf{X}_{\pm}$, i.e. it is represented by part of the data matrix $\mathbf{X}_{\pm}$. Meanwhile, each column vector of $\mathbf{X}_{\pm}$ is represented as the weighted sum of the basis vectors (see Eq. (8)). This approach provides a clearer interpretation of the based-parts presentation.

3.2 Solution to GCNMF

By substituting the Eqs. (9) to (10), we have

$$\begin{aligned} O_3(\mathbf{U},\mathbf{V}) &= \mathrm{Tr}\Big((\mathbf{X}-\mathbf{X}\mathbf{U}\mathbf{V}^T)(\mathbf{X}-\mathbf{X}\mathbf{U}\mathbf{V}^T)^T\Big)+\lambda R(\mathbf{V}) \\ &= \mathrm{Tr}(\mathbf{X}\mathbf{X}^T)-2\mathrm{Tr}(\mathbf{X}\mathbf{U}\mathbf{V}^T\mathbf{X}^T)+\mathrm{Tr}(\mathbf{X}\mathbf{U}\mathbf{V}^T\mathbf{V}\mathbf{U}^T\mathbf{X}^T) \\ &\quad +\lambda\mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) \end{aligned}. \tag{11}$$

Since all the entries of $\mathbf{U}$ and $\mathbf{V}$ are nonnegative, we define the Lagrangian multipliers of $\mathbf{U}$ and $\mathbf{V}$ with $\mathbf{\Theta}=[\theta_{ik}]$ and $\mathbf{\Phi}=[\phi_{jk}]$ respectively. Then, the Lagrangian function is expressed as

$$\begin{aligned} L(\mathbf{U},\mathbf{V}) = {} & \mathrm{Tr}(\mathbf{X}\mathbf{X}^T)-2\mathrm{Tr}(\mathbf{X}\mathbf{U}\mathbf{V}^T\mathbf{X}^T)+\mathrm{Tr}(\mathbf{X}\mathbf{U}\mathbf{V}^T\mathbf{V}\mathbf{U}^T\mathbf{X}^T) \\ & +\lambda\mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V})+\mathrm{Tr}(\mathbf{\Theta}\mathbf{U}^T)+\mathrm{Tr}(\mathbf{\Phi}\mathbf{V}^T) \end{aligned}. \tag{12}$$

Setting the partial derivatives of $L(\mathbf{U},\mathbf{V})$ with respect to the primal variables $\mathbf{U}$ and $\mathbf{V}$ to zero gives the following equations,

$$\frac{\partial L(\mathbf{U},\mathbf{V})}{\partial \mathbf{U}} = -2\mathbf{X}^T\mathbf{X}\mathbf{V} + 2\mathbf{X}^T\mathbf{X}\mathbf{U}\mathbf{V}^T\mathbf{V} + \mathbf{\Theta} = \mathbf{0}, \tag{13}$$

$$\frac{\partial L(\mathbf{U},\mathbf{V})}{\partial \mathbf{V}} = -2\mathbf{X}^T\mathbf{X}\mathbf{U} + 2\mathbf{V}\mathbf{U}^T\mathbf{X}^T\mathbf{X}\mathbf{U} + 2\lambda\mathbf{L}\mathbf{V} + \mathbf{\Phi} = \mathbf{0}. \tag{14}$$

With the Karush–Kuhn–Tucker (KKT) conditions, i.e. $\theta_{ik}u_{ik} = 0$ and $\phi_{jk}v_{jk} = 0$, the equations below can be derived from Eqs. (13) and (14),

$$-(\mathbf{X}^T\mathbf{X}\mathbf{V})_{ik}u_{ik} + (\mathbf{X}^T\mathbf{X}\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik}u_{ik} = 0, \tag{15}$$

$$-(\mathbf{X}^T\mathbf{X}\mathbf{U})_{jk}v_{jk} + (\mathbf{V}\mathbf{U}^T\mathbf{X}^T\mathbf{X}\mathbf{U})_{jk}v_{jk} + \lambda(\mathbf{L}\mathbf{V})_{jk}v_{jk} = 0. \tag{16}$$

Define two nonnegative matrices, $(\mathbf{X}^T\mathbf{X})^+ = \frac{|\mathbf{X}^T\mathbf{X}| + \mathbf{X}^T\mathbf{X}}{2}$ and $(\mathbf{X}^T\mathbf{X})^- = \frac{|\mathbf{X}^T\mathbf{X}| - \mathbf{X}^T\mathbf{X}}{2}$, then $\mathbf{X}^T\mathbf{X}$ can be expressed with its positive and negative parts, i.e. $\mathbf{X}^T\mathbf{X} = (\mathbf{X}^T\mathbf{X})^+ - (\mathbf{X}^T\mathbf{X})^-$. Substituting this expression to Eqs. (15) and (16) gives the following equation,

$$-(((\mathbf{X}^T\mathbf{X})^+ - (\mathbf{X}^T\mathbf{X})^-)\mathbf{V})_{ik}u_{ik} + (((\mathbf{X}^T\mathbf{X})^+ - (\mathbf{X}^T\mathbf{X})^-)\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik}u_{ik} = 0, \tag{15-1}$$

$$-(((\mathbf{X}^T\mathbf{X})^+ - (\mathbf{X}^T\mathbf{X})^-)\mathbf{U})_{jk}v_{jk} + (\mathbf{V}\mathbf{U}^T((\mathbf{X}^T\mathbf{X})^+ - (\mathbf{X}^T\mathbf{X})^-)\mathbf{U})_{jk}v_{jk} + \lambda((\mathbf{D}-\mathbf{W})\mathbf{V})_{jk}v_{jk} = 0. \tag{16-1}$$

i.e.

$$((\mathbf{X}^T\mathbf{X})^-\mathbf{V})_{ik}u_{ik} + ((\mathbf{X}^T\mathbf{X})^+\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik}u_{ik} = ((\mathbf{X}^T\mathbf{X})^+\mathbf{V})_{ik}u_{ik} + ((\mathbf{X}^T\mathbf{X})^-\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik}u_{ik}, \tag{15-2}$$

$$((\mathbf{X}^T\mathbf{X})^-\mathbf{U} + \mathbf{V}\mathbf{U}^T(\mathbf{X}^T\mathbf{X})^+\mathbf{U} + \lambda\mathbf{D}\mathbf{V})_{jk}v_{jk} = ((\mathbf{X}^T\mathbf{X})^+\mathbf{U} + \mathbf{V}\mathbf{U}^T(\mathbf{X}^T\mathbf{X})^-\mathbf{U} + \lambda\mathbf{W}\mathbf{V})_{jk}v_{jk}. \tag{16-2}$$

Then, the two iterative update rules below can be obtained,

$$u_{ik} \leftarrow u_{ik}\frac{((\mathbf{X}^T\mathbf{X})^+\mathbf{V} + (\mathbf{X}^T\mathbf{X})^-\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik}}{((\mathbf{X}^T\mathbf{X})^-\mathbf{V} + (\mathbf{X}^T\mathbf{X})^+\mathbf{U}\mathbf{V}^T\mathbf{V})_{ik}}, \tag{17}$$

$$v_{jk} \leftarrow v_{jk}\frac{((\mathbf{X}^T\mathbf{X})^+\mathbf{U} + \mathbf{V}\mathbf{U}^T(\mathbf{X}^T\mathbf{X})^-\mathbf{U} + \lambda\mathbf{W}\mathbf{V})_{jk}}{((\mathbf{X}^T\mathbf{X})^-\mathbf{U} + \mathbf{V}\mathbf{U}^T(\mathbf{X}^T\mathbf{X})^+\mathbf{U} + \lambda\mathbf{D}\mathbf{V})_{jk}}. \tag{18}$$

In fact, the objective function $O_3(\mathbf{U},\mathbf{V})$ in Eq. (10) is non-increasing based on the updating rules in Eqs. (17) and (18), as shown in the proof of GNMF (see appendix A in [12] for more details). Clearly, the iterative rules can be implemented by using the multiplicative algorithms for nonnegative matrix factorization discussed in [22], [6]. Note that the solution to the minimization of the objective function in Eq. (10) is not unique. For a given solution $\mathbf{U}$ and $\mathbf{V}$, it is easy to verify that $\mathbf{U}\mathbf{H}$ and $\mathbf{V}\mathbf{H}^{-1}$ are also the solution of the objective function for any positive diagonal matrix $\mathbf{H}$. To obtain a unique solution, a feasible technique is to normalize the Euclidean length of the column vectors of $\mathbf{U}$

to unity while $\mathbf{UV}^T$ remains unchanged [27]. In this paper, we use the L_1-norm to normalize the column vectors of $\mathbf{U}$, i.e.

$$u_{ik} \leftarrow \frac{u_{ik}}{\sum_{i=1}^{N} |u_{ik}|}, \tag{19}$$

$$v_{jk} \leftarrow v_{jk} \sum_{i=1}^{N} |u_{ik}|, \tag{20}$$

4 EXPERIMENT RESULTS

The clustering experiments carried out to investigate the effectiveness of the proposed GCNMF algorithm are presented in this section. The sparseness of the basis vectors and the encoding vectors in the GCNMF algorithm is also studied. A total of six algorithms are involved in the clustering experiments, including K-means clustering in original space (KM) [12], Normalized Cut (NCut) [28], NMF-based clustering [1], [2], GNMF-based clustering [12], CNMF-based clustering [13] and the proposed GCNMF-based clustering. Among these six algorithms, NMF and GNMF require that all entries of the data matrix should be nonnegative, while the other algorithms can be applied to both nonnegative and mixed-sign data matrix. The K-means clustering method is employed to evaluate and compare the performance of the six algorithms.

4.1 Data Preparation

The group of NMF techniques, i.e. NMF, GNMF, CNMF and GCNMF, is a powerful tool for image clustering. Three image data sets are thus prepared for the clustering experiments. In addition, the multiple feature data set in [29] is also employed in the experiments. In these four image data sets, two of them involve data matrix of nonnegative entries, which are to be used for the NMF and GNMF algorithms. The other two contain mixed-sign entries in the data matrix. The details are discussed as follows.

The first data set is obtained from the PIE face database of the Carnegie Mellon University (CMU) (downloadable from http://www.cad.zju.edu.cn/home/dengcai). The face images are created under different poses, illuminations and expressions. The database contains 41,368 images of 68 subjects. The image size is 32×32 pixels, with 256 grey levels. 1428 images under different illumination conditions are selected for the clustering experiment. The second data set is obtained from the COIL20 image library of the Columbia University (downloadable from http://www1.cs.columbia.edu/CAVE/software/softlib/coil-20.php). It contains 1440 images generated from 20 objects. Each image is represented by a 1024-dimensional vector, and the size is 32×32 pixels with 256 grey levels per pixel. All the data of PIE and COIL20 are nonnegative. Hence, they both can be used to evaluate the six algorithms.

The third data set is obtained from the USPS handwritten digits dataset (USPS) (downloadable from http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets). It contains 7291 training images and 2007 testing images of handwritten digits. In the experiments, the training images are adopted, where the number of samples for the digits ‘0’ to ‘9’ is 1194, 1005, 731, 658, 652, 556, 664, 645, 542 and 644 respectively. The size of each image is 16×16 pixels, with 256 grey levels per pixel. Further details about this dataset can be found in [30]. The fourth data set is the multiple feature data set (MFD) [29], consisting of features of handwritten digits (‘0’ to ‘9’) extracted from a collection of Dutch utility maps. 200 patterns per class (for a total of 2,000 patterns) are digitized into binary images. Each digit is represented by a 649-dimensional vector in terms of six feature sets: Fourier coefficients of the character shapes, profile correlations, Karhunen-Love coefficients, pixel averages in 2×3 windows, Zernike moments and morphological features. Since the data of USPS and MFD are of mixed-sign, they can only be used to evaluate the KM [12], NCut [28], CNMF [13] and GCNMF algorithms.

4.2 Evaluation Metrics

The clustering performance of the six algorithms is evaluated by comparing the label mapped to each data point with the label provided. The procedure is as follows. First, the algorithms under comparison (except KM) are executed respectively to obtain a new representation of each data point. The K-means clustering method is then applied to these new representations to get the clustering labels. Finally, the clustering labels are mapped to the equivalent labels provided by the data sets using the Kuhn-Munkres algorithm [31].

Two metrics, the clustering accuracy (AC) and the normalized mutual information (NMI), are used to evaluate the clustering performance of the six algorithms under comparison. Details about these two metrics and definitions can be found in [18], [32]. Besides, for the group of NMF algorithms, the sparseness on the basis vectors and/or the encoding vectors is usually used to evaluate the power of the parts-based representation. Here, we measure the sparseness of the basis vectors and the encoding vectors based on the relationship between the L_1 and L_2 norm of a given vector using the sparseness metric in [20], [27] as follows,

$$Sparseness(\mathbf{y}) = \frac{\sqrt{Q} - \sum |y_i| \Big/ \sqrt{\sum y_i^2}}{\sqrt{Q} - 1}, \tag{24}$$

where Q is the dimensionality of the vector $\mathbf{y}$. This sparseness metric quantifies the energy of a vector that is packed into a few components only. The metric is unity 1 if and only if $\mathbf{y}$ contains only a single nonzero component, and takes a value of 0 if and only if all the components are equal, interpolating smoothly between the two extremes. In our experiments, we consistently use the column vectors of $\mathbf{U}$ and $\mathbf{V}$ to compute their sparseness.

4.3 Nonnegative Data Sets: PIE and COIL20

In this section, the clustering experiments conducted using the six algorithms on the two nonnegative data sets PIE and COIL20 are discussed.

4.3.1 Numerical Results

In the clustering experiments, the number of the nearest neighbors p for constructing the adjacency graph and the parameter λ in both GNMF and GCNMF are empirically fixed at 5 and 100 respectively. For each data set, the experiments are conducted repeatedly with different number of clusters K. For the PIE data set, K takes the values in the grid $\{10, 20, \cdots, 60, 68\}$. For the COIL20 data set, K takes the values in the grid $\{2, 4, \cdots, 20\}$. For a given value of K, the experimental process is described as follows:

1) Select K classes randomly from the data set;
2) Run the corresponding algorithm (except KM);
3) Execute K-means clustering algorithm for 20 times with different initialization settings and record the best results;
4) Repeat steps 1) to 3) for 20 times (except when K reaches the maximum value, i.e. the entire data set are chosen);
5) Compute the mean and standard error for the given value of K;
6) Repeat steps 1) to 5) with another value of K, until all the values of K have been selected.

The clustering results are reported respectively in Tables 2 and 3. The findings are highlighted as follows:

1) GCNMF significantly outperforms CNMF, which demonstrates the importance of geometrical structure in the discovery of hidden information.
2) Among the six algorithms, GCNMF, GNMF and NCut use the geometrical structure to reveal the hidden information. The experimental results show that these three algorithms are able to achieve better results than the rest, i.e. KM, NMF and CNMF. This finding again indicates that the geometrical structure plays an important role in the clustering process.
3) Compared with GNMF, the proposed GCNMF exhibits better performance for a majority of the K values and so does the total average performance (Av.), which validates that it is beneficial for the clustering process to restrict the basis vectors in the space of the data set.

Table 2 Clustering Results on PIE

K	Accuracy (%)					
	KM	NCut	NMF	GNMF	CNMF	GCNMF
5	38.81±6.64	**97.64±6.10**	55.45±6.00	84.67±13.89	48.90±3.15	89.83±10.91
10	29.64±3.43	**90.31±9.07**	49.69±5.76	85.69±8.01	42.21±2.70	85.75±9.58
20	28.12±2.82	79.46±4.84	44.43±4.21	**82.58±4.37**	36.48±2.05	80.80±6.02
30	26.63±1.31	75.17±2.68	42.45±2.87	79.06±4.55	37.51±1.96	**79.19±3.20**
40	25.77±1.37	72.93±3.87	41.31±2.49	77.90±3.79	34.08±1.78	**78.04±4.13**

50	25.18±1.33	69.83±2.57	39.76±2.13	76.61±3.32	32.02±2.04	**77.57±3.54**
60	24.14±0.94	68.10±2.63	39.32±1.57	**75.45±1.64**	30.57±1.72	74.98±2.52
68	23.14	68.21	39.08	74.19	28.12	**77.59**
Av.	27.68±2.23	77.71±3.97	43.94±3.13	79.52±4.95	36.24±1.93	**80.47±4.99**
K	Normalized Mutual Information (%)					
	KM	NCut	NMF	GNMF	CNMF	GCNMF
5	27.92±9.10	**96.99±4.02**	48.63±6.45	85.63±10.55	37.42±7.08	88.29±10.31
10	35.06±5.77	**92.58±5.39**	58.30±5.02	88.96±4.75	47.15±3.89	88.96±5.50
20	45.16±2.78	87.77±1.97	64.33±3.19	**90.39±1.93**	51.49±2.19	89.63±2.82
30	48.61±2.18	86.40±1.65	66.36±2.24	**89.51±1.98**	54.70±1.87	89.23±1.31
40	50.56±1.62	85.07±1.54	67.44±2.09	89.00±1.25	53.69±1.55	**89.12±1.62**
50	52.00±1.42	83.50±1.68	67.99±1.45	88.86±1.19	53.38±1.41	**89.07±1.18**
60	52.90±0.99	82.79±1.57	69.00±1.19	**88.51±0.56**	52.99±1.15	88.26±0.92
68	52.69	82.25	68.20	87.68	52.01	**88.58**
Av.	45.61±2.98	87.17±2.23	63.78±2.70	88.57±2.78	50.35±2.39	**88.89±2.96**

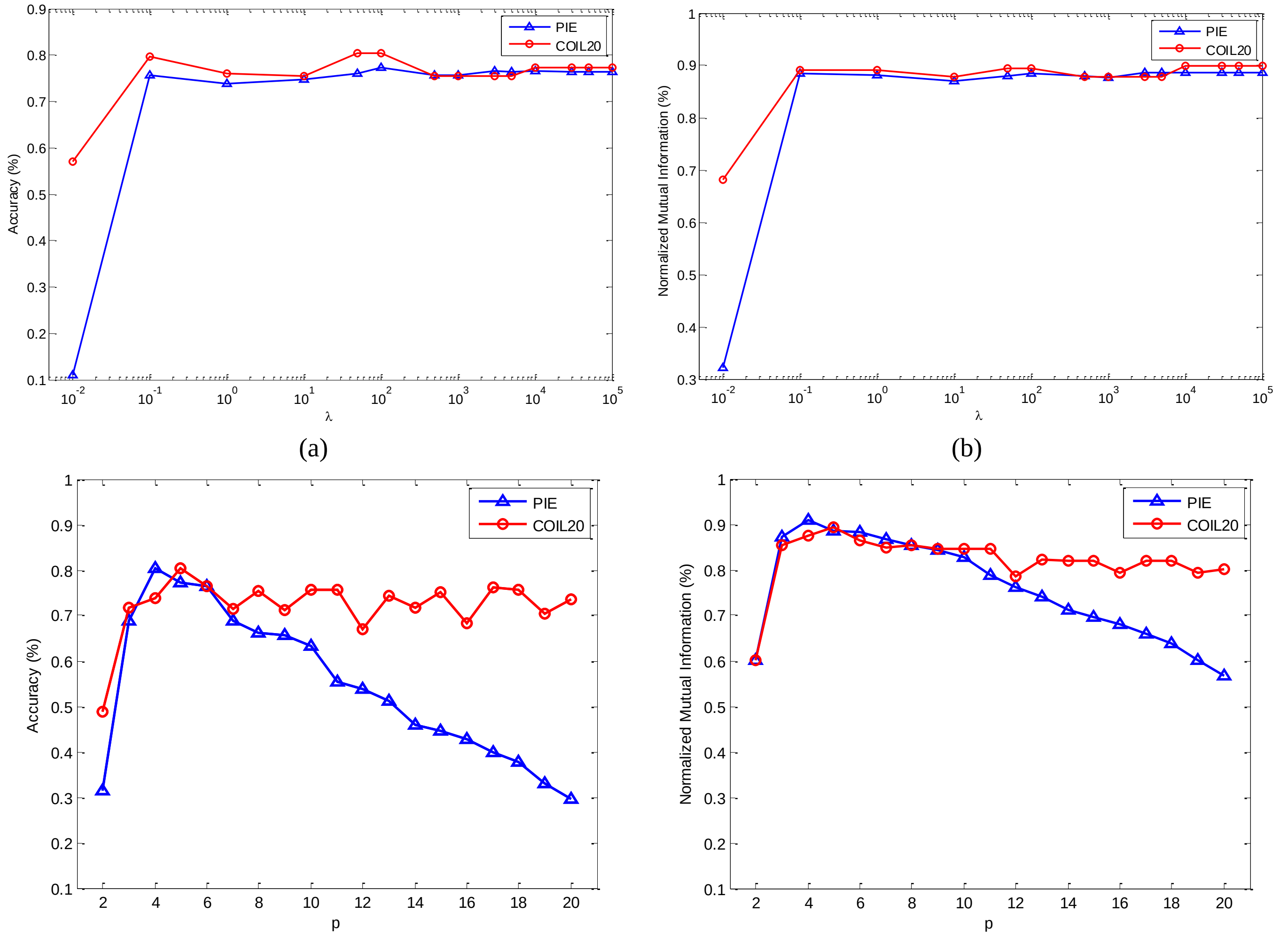
Table 3 Clustering Results on COIL20

K	Accuracy (%)					
	KM	NCut	NMF	GNMF	CNMF	GCNMF
2	90.76±12.19	**95.31±12.07**	90.66±12.15	94.13±12.27	90.59±11.70	94.93±14.24
4	87.69±9.52	85.38±16.62	79.93±15.00	89.46±13.29	83.78±12.44	**91.55±11.96**
6	79.27±10.17	84.73±11.50	77.95±9.42	93.72±8.71	74.33±10.07	**93.74±8.99**
8	70.34±5.32	74.33±8.51	68.45±7.69	**83.08±7.53**	61.25±6.49	82.80±7.68
10	69.94±6.63	74.67±7.19	70.13±9.27	**86.89±8.21**	57.17±7.25	85.60±7.78
12	63.75±6.16	71.79±5.54	64.69±4.12	77.63±6.22	50.63±4.61	**79.47±7.16**
14	68.96±5.33	74.59±6.67	68.23±5.40	83.42±5.62	52.92±5.22	**83.53±5.12**
16	64.49±5.90	71.96±5.94	64.13±4.92	78.78±4.20	47.66±4.41	**78.82±4.99**
18	62.55±3.35	70.19±4.94	63.13±2.54	78.77±4.48	44.28±3.01	**80.09±4.77**
20	62.71	69.24	63.61	**81.60**	45.97	77.78
Av.	72.05±6.46	77.22±7.90	71.09±7.05	84.75±7.05	60.86±6.52	**84.83±7.27**
K	Normalized Mutual Information (%)					
	KM	NCut	NMF	GNMF	CNMF	GCNMF
2	72.05±29.25	88.47±28.19	71.72±29.26	84.22±29.22	70.15±29.40	**88.61±30.09**
4	79.76±14.01	88.01±12.01	70.44±17.43	**88.54±11.01**	73.70±15.42	88.22±14.35
6	77.56±8.36	90.49±6.67	75.38±8.16	92.55±8.64	69.87±9.30	**93.64±6.93**
8	69.81±5.03	82.84±5.83	68.72±5.86	**85.78±5.21**	60.68±6.07	85.49±5.20
10	73.58±5.48	85.32±4.10	72.35±7.40	**90.66±4.99**	62.46±6.63	89.93±5.16
12	69.95±4.90	83.92±3.21	69.45±3.77	86.43±4.33	58.00±4.86	**86.47±4.75**
14	75.69±4.36	86.44±3.66	74.16±4.04	89.45±3.30	61.01±4.46	**90.08±2.61**
16	73.37±4.17	84.41±3.23	72.41±4.00	87.90±2.50	57.21±3.75	**88.01±2.46**
18	73.86±2.19	84.19±2.67	72.75±2.11	88.47±2.22	55.79±2.59	**89.07±2.34**
20	74.55	83.23	71.77	**89.71**	59.18	89.66
Av.	74.02±7.78	85.73±6.96	71.92±8.20	88.37±7.14	62.81±8.25	**88.92±7.39**

4.3.2 Parameter Selection

For the proposed GCNMF algorithm, it is necessary to set the control parameter λ and the number of nearest neighbors p for constructing the adjacency graph. They are empirically set to 100 and 5 respectively as in the previous experiments. The effect of these two parameters on the clustering performance is investigated in this section.

In the experiments, the GCNMF algorithms are executed on the entire data set of PIE and COIL20 respectively. The 0-1 weights are adopted. The value of p is set to 5 when the effect of λ on the clustering performance is investigated, while λ is set to 100 when effect of p is studied. Fig. 1 shows the variation in performance of GCNMF with λ and p. In Fig 1(a) and (b), λ takes the values in the grid {1e-2, 1e-1, 1e+0, 1e+1, 5e+1, 1e+2, 5e+2, 1e+3, 3e+3, 5e+3, 1e+4, 3e+4, 5e+4, 1e+5}, where a base 10 logarithmic scale is used for the x-axis. In Fig. 1(c) and (d), p takes the values in the grid {2, 3, …, 20}. It can been seen that for both the PIE and COIL20 data sets, GCNMF is able to achieve good performance over a wide range of λ (from 1e-1 to 1e+5), which demonstrates that GCNMF is insensitive to this control parameter. Especially, the performance is the best when λ is set 50 or 100. However, the number of nearest neighbors *p* has to be selected from the range between 3 and 7 in order to maintain high accuracy.



(a) (b)

(c) (d)

Fig. 1 Performance of GCNMF on PIE and COIL20 under different parameter setting: (a) AC versus λ, (b) NMI versus λ, (c) AC versus p, (d) NMI versus p.

4.3.3 Sparseness

In this section, the sparseness of the basis vectors (the column vectors of $\mathbf{U}$) and the encoding vectors (the column vectors of $\mathbf{V}$) is studied to evaluate the power of the parts-based representation for the group of NMF algorithms.

In the experiments, all classes of the corresponding data set are used, which means that the column number of $\mathbf{U}$ and $\mathbf{V}$ is equal to the class number of the data set. For example, with the PIE data set containing 41,368 images of 68 subjects, there are 68 column vectors in $\mathbf{U}$ and $\mathbf{V}$. For each column vector, Eq. (24) is used to evaluate the sparseness. As a result, there are 68 sparseness values of the basis vectors and the encoding vectors respectively. An average value, namely, “average sparseness”, is then used to evaluate their sparseness. It is defined as the average of the sparseness values obtained over all the basis vectors and encoding vectors. Table 4 shows the experimental results. The findings of the experiments are discussed as follows.

1) In all the data sets, both NMF and CNMF exhibit better sparseness for the encoding vectors, but their clustering performance is relatively inferior (see the fourth and sixth columns in Tables 2 and 3). This demonstrates the importance of the smoothness of the encoding vectors, and highlights the manifold regularization ability in GNMF and GCNMF.
2) While the encoding vectors obtained by CNMF and GCNMF are smooth, they are also able to achieve better clustering performance (see the fifth and seventh columns in Tables 2 and 3). The results show that it is important for the clustering process to preserve the geometrical structure of the sample data in low-dimension representation, i.e. the row vectors of $\mathbf{V}$.
3) For NMF, sparseness of both the basis vectors and the encoding vectors can be achieved with the two data sets. For CNMF, this can only be achieved for the COIL20 data set. However, their clustering performance is relatively poor (see the fourth and sixth columns in Tables 2 and 3), indicating that strengthening the sparseness of both the basis and encoding vectors deteriorates the goodness-of-fit of the model for the data.

Table 4 Sparseness of basis and encoding vectors on PIE and COIL20

Data set	Method	Average sparseness of basis vectors	Average sparseness of encoding vectors
PIE	NMF	0.4642	0.3930
	GNMF	0.2498	0.0018
	CNMF	0.0160	0.5080
	GCNMF	0.1412	0.0021
COIL20	NMF	0.5280	0.4126

GNMF	0.3891	0.0018
CNMF	0.4330	0.6165
GCNMF	0.2025	0.0080

4.4 Mixed-sign Data Sets: USPS and MFD

In this section, clustering experiments are performed using KM, NCut, CNMF and GCNMF respectively with the two mixed-sign data sets, USPS and MFD. The two algorithms NMF and GNMF are excluded because they can only work for nonnegative data matrix.

4.4.1 Numerical Results

Similar to the setting described in section 4.3, the parameters p and λ are also set to 5 and 100 respectively in this experiment. For each data set, the experiment is conducted repeatedly with different number of clusters K. The value K for both the USPS and MFD data sets take the values in the grid $\{2, 3, \cdots, 10\}$. The experimental process is the same as that described in section 4.3.1. Tables 5 and 6 give the experimental results respectively for the USPS and MFD data sets, from which the following observations can be made:

1) The proposed GCNMF outperforms CNMF considerably for both the USPS and MFD data sets, demonstrating the importance of geometrical structure in revealing the hidden information.
2) Regardless of the data sets, GCNMF and NCut achieve the best clustering performance. This result once again shows that geometrical structure plays an important role in revealing the hidden information. Besides, NCut generally performs better than the proposed GCNMF. This is due to the fact that the parameters p and λ are not optimized. The issue will be discussed further in next section.

Table 5 Clustering Results on USPS

K	Accuracy (%)				Normalized Mutual Information (%)			
	KM	NCut	CNMF	GCNMF	KM	NCut	CNMF	GCNMF
2	94.02±7.89	**99.45±0.53**	56.52±4.53	97.61±7.82	75.89±20.11	**95.46±3.71**	1.88±1.95	91.25±17.08
3	90.97±4.82	91.61±15.53	44.36±4.05	**91.69±13.62**	72.09±8.91	**87.35±11.33**	5.73±3.65	85.32±13.75
4	78.36±14.32	90.49±13.44	39.66±4.20	**95.14±7.81**	67.13±10.54	87.67±8.01	11.03±5.23	**89.35±5.68**
5	75.14±10.41	80.43±11.93	35.59±4.16	**86.27±13.98**	66.97±7.03	84.13±4.19	13.96±6.17	**84.18±7.44**
6	70.42±8.30	78.55±11.73	33.31±3.72	**81.27±13.28**	65.06±5.38	**84.26±5.12**	16.98±4.80	84.16±6.80
7	71.79±6.98	78.34±12.32	31.11±2.32	**80.73±9.45**	64.72±4.81	**83.96±4.35**	16.51±3.06	83.04±3.58
8	70.46±4.85	71.57±9.46	29.11±2.89	**79.60±7.69**	64.32±3.29	81.53±3.72	16.81±3.05	**84.17±2.69**
9	68.67±1.79	69.92±6.28	27.82±2.11	**76.28±7.33**	63.44±2.51	81.69±1.60	17.55±2.71	**82.70±2.79**
10	68.37	**68.74**	26.31	68.56	62.92	**81.16**	16.23	79.57
Av.	76.47±6.60	81.01±9.02	35.98±3.11	**84.13±9.00**	66.95±6.95	**85.25±4.67**	12.96±3.40	84.86±6.65

Table 6 Clustering Results on MFD

K	Accuracy (%)	Normalized Mutual Information (%)

	KM	NCut	CNMF	GCNMF	KM	NCut	CNMF	GCNMF
2	98.61±0.96	**99.10±0.94**	69.26±15.99	98.76±1.36	90.26±5.88	**93.47±5.60**	23.29±36.85	91.85±7.41
3	97.47±1.98	**98.93±0.64**	67.23±9.06	92.61±11.43	89.76±5.99	**94.88±2.50**	37.54±16.84	85.40±12.74
4	94.71±2.17	**97.48±1.21**	60.38±9.42	95.91±2.94	84.32±4.78	**91.62±3.40**	39.04±10.57	89.05±5.06
5	94.44±1.96	**97.91±1.11**	59.68±6.07	95.54±3.64	85.79±3.87	**93.69±2.79**	40.66±7.86	89.64±5.66
6	93.47±1.69	**97.07±0.92**	51.04±5.06	94.89±4.49	84.42±3.00	**92.08±2.18**	36.78±5.86	88.99±4.87
7	91.26±4.28	**96.44±0.92**	49.18±2.87	93.91±3.02	83.17±2.75	**91.42±1.90**	36.13±3.05	88.55±3.12
8	90.75±3.77	**95.20±4.32**	45.78±3.58	92.95±5.39	83.31±2.19	**91.51±2.31**	37.20±2.39	89.48±2.90
9	87.15±6.46	**96.05±0.57**	42.85±2.76	91.16±6.10	81.46±3.04	**91.49±1.02**	36.26±1.82	88.65±3.21
10	77.60	95.75	42.40	**96.25**	77.63	91.29	36.61	**92.26**
Av.	91.72±2.59	**97.10±1.18**	54.20±6.09	94.66±4.26	84.46±3.50	**92.38±2.41**	35.95±9.47	89.32±5.00

4.4.2 Parameter Selection

Next, with the USPS and MFD data sets, we investigate the effect of λ and p in GCNMF on the clustering performance. Fig. 3 shows the experimental results, where λ and p take the values in the grids {1e-2, 1e-1, 1e+0, 1e+1, 5e+1, 1e+2, 5e+2, 1e+3, 3e+3, 5e+3, 1e+4, 3e+4, 5e+4, 1e+5} and {2, 3, …, 20} respectively. The results show that GCNMF is very stable when $\lambda \geq 10$ while being sensitive to λ when $\lambda \leq 1$. In fact, increasing λ will improve the smoothness of the basis and encoding vectors for the low-dimensional representation (see Eqs. (9) and (10)), which implies that manifold regularization can stabilize the process of matrix factorization in NMF. However, the performance is found to decrease with p due to the fact that as p increases, the local invariant is not likely to be preserved when the local geometrical structure of the data manifold is captured. From the perspective of accuracy, it can be seen from Fig. 2(a) and (c) that the best values of the parameters are $p = 2$, $\lambda = 1$ for USPS, and $p = 3$, $\lambda = 1\text{e-}1$ for MFD. However, from the perspective of normalized mutual information, Fig. 2(b) and (d) show that the best values of the parameters are $p = 3$, $\lambda = 1\text{e-}2$ for USPS, and $p = 3$, $\lambda = 1\text{e-}1$ for MFD. For simplicity, p and λ are experimentally set to 5 and 100, which may be the reason why NCut generally performs better in the numerical experiments.

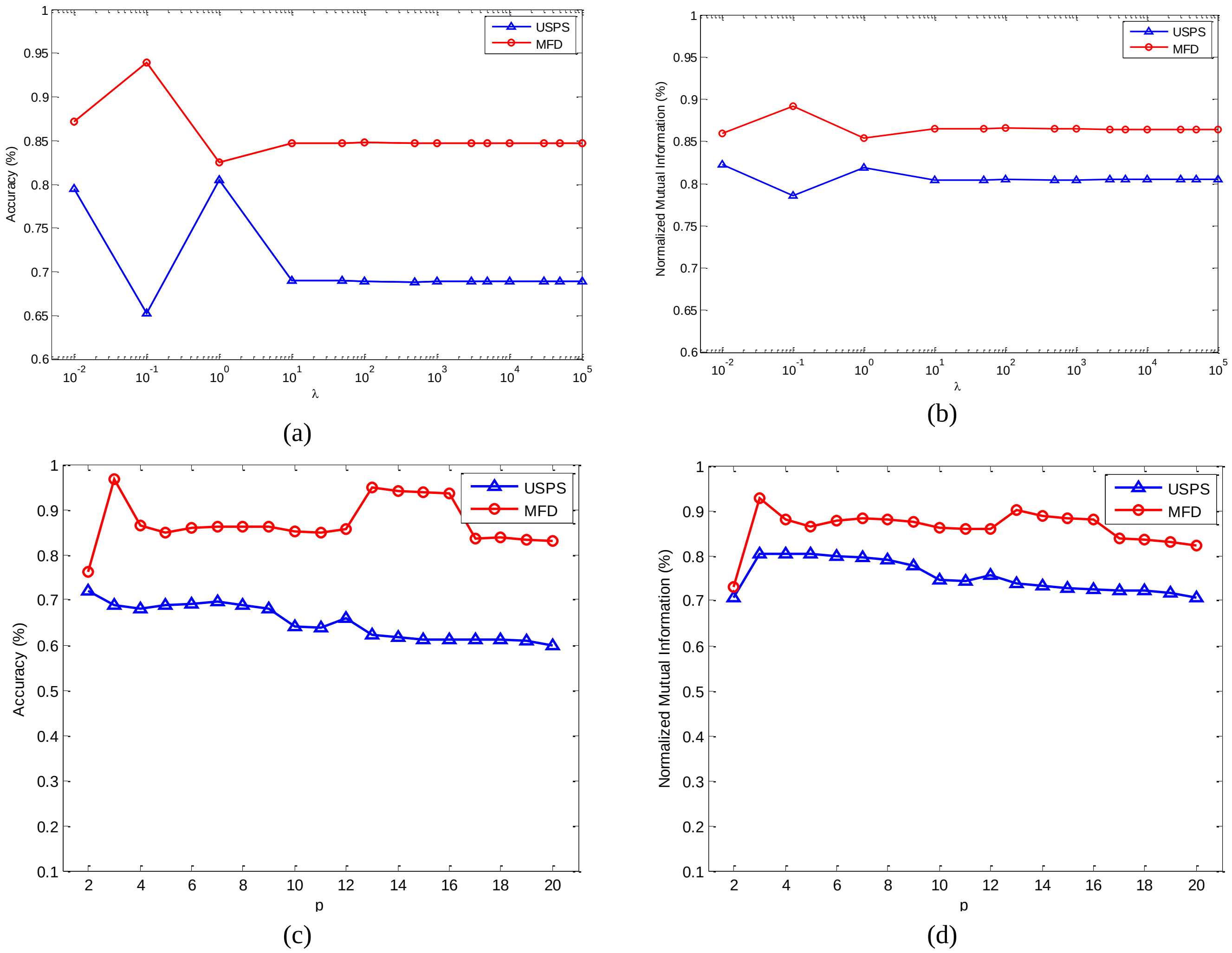


Fig. 2 Performance of GCNMF on USPS and MFD under different parameter setting: (a) AC versus λ, (b) NMI versus λ, (c) AC versus p, (d) NMI versus p.

4.4.3 Sparseness

The USPS and MFD data sets of mixed-sign are used to investigate the sparseness of CNMF and GCNMF in this section. The experimental process is that same as that mentioned in section 4.3.3 and the results are shown in Table 7. The following observations are made from the experiments.

1) The basis vectors obtained by GCNMF are sparser than that by CNMF, while the encoding vectors obtained by CNMF are sparser than that by GCNMF. This verifies that strengthening the sparseness of one vector, basis vector or encoding vector, will affect the smoothness of the other vector.
2) In general, both the basis vectors and the encoding vectors obtained in CNMF have good sparseness. However, it leads to inferior clustering results (see the second line from the bottom in Tables 5 and 6). This not only shows that the geometrical structure plays an important role in the clustering process, but also demonstrates that strengthening the sparseness of the two vectors will deteriorate the goodness-of-fit of the model of the data.

Table 7 Sparseness of basis and encoding vectors on USPS and MFD

Data set	Method	Average sparseness of basis vectors	Average sparseness of encoding vectors
USPS	CNMF	0.3637	0.4703
	GCNMF	0.7123	0.0002
MFD	CNMF	0.6923	0.6054
	GCNMF	0.9294	0.0021

5 CONCLUSION AND FUTURE WORK

The novel nonnegative matrix factorization technique GCNMF is proposed in this paper. The method extends the application of NMF by enabling it to deal with mixed-sign data. Besides, the basis and encoding vectors obtained by GCNMF have better representation power because the proposed method takes into account the geometric structure of the data manifold. In comparison with other clustering methods, including KM, NCut, NMF, GNMF and CNMF, the results of the experiments performed on four real-world data sets validate that the performance of GCNMF is significantly better.

Like many manifold learning algorithms, GCNMF requires the construction of an adjacency graph to reveal the intrinsic structural information. The construction of the graph in turn requires the selection of the number of nearest neighbors p in order to match the local structure, and also the value of the λ to control the tradeoff between the approximation error of matrix factorization and the geometric structure information. However, the selection of suitable values for these two parameters in a theoretical way remains an issue. Although the experimental results show that the GCNMF method is not sensitive to the value of λ, further research effort is still required to confirm this finding. Finally, the proposed method is achieved by introducing a manifold regularized term into the CNMF method, which is based on the column-wise representation of the approximation in the CNMF algorithm (see Eq. (7)), with the basis and encoding vectors given by the linear combination of the data points. In fact, the NMF problem can be approached by representing the basis vectors and the encoding vectors in other ways. For example, the column vectors of the basis vectors can be used to indicate the cluster centroids while the encoding vectors can serve as cluster membership indicators. This implies that the regularized technique in this paper can be further optimized from other perspectives. Some methods have been proposed recently to improve NMF based on the assumption of separability in [10], [34] [35], e.g. Linear Programming (LP) model, which may be used to solve the model of our method in a more effective way. Ongoing work is being conducted along this line of investigation.

ACKNOWLEDGEMENTS

This work was supported in part by General Research Fund of the Hong Kong Research Grants Council under Grant PolyU

5134/12E; the National Natural Science Foundation of China under Grants 61370173, 61272210, 61202311, by the Natural Science Foundation of Jiangsu Province under Grants BK2011003, BK2011417, by the Natural Science Foundation of Zhejiang Province under Grants LY13F020011, LY12F03008, LY14F010010, LY14F020009, by Key Innovative Research Team of Huzhou: Internet of Things-Technology & Application under Grant 2012KC04, and by the Humanities and Social Science Foundation of Ministry of Education of China under Grant 13YJAZH084. Thank reviewers for their valuable comments and we would like to offer the source program if interested researchers need it.

REFERENCES

[1] P. Paatero, U. Tapper, "Positive matrix factorization—a nonnegative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111-126, 1994.

[2] D. D. Lee, H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, pp. 788-791, Oct. 1999.

[3] A. Hyvärinen, E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411-430, Dec. 2000.

[4] A. Hyvärinen, J. Karhunen, E. Oja, Independent Component Analysis. John Wiley and Sons, 2001.

[5] S. Vavasis, "On the complexity of non-negative matrix factorization," *SIAM Journal on Optimization*, vol. 20, no. 3, pp.1364-1377, 2009.

[6] C. J. Lin, "On the convergence of multiplicative update algorithms for non-negative matrix factorization," *IEEE Transactions on Neural Networks*, vol. 18, no. 6, pp. 1589-1596, Nov. 2007.

[7] S. Arora, R. Ge, R. Kannan, and A. Moitra, "Computing a nonnegative matrix factorization – provably," *Proceedings of the 44th annual ACM symposium on Theory of computing (STOC 2012)*, pp. 145-162, 2012.

[8] D. Donoho, and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts? " *In Advances in Neural Information Processing Systems (NIPS 2003)*, 2003.

[9] E. Esser, M. Möller, S. Osher, G. Sapiro, and J. Xin, "A convex model for non-negative matrix factorization and dimensionality reduction on physical space," *IEEE Transactions on Image Processing*, vol. 21, no. 10, pp. 3239-3252, 2012.

[10] A. Kumar, V. Sindhwani, and P. Kambadur, "Fast conical hull algorithms for near-separable non-negative matrix factorization," *Proceedings of the 30th International Conference on Machine Learning*, S. Dasgupta and D. McAllester eds.,vol. 28, no. 1, pp. 231-239, 2013.

[11] T. Feng, S. Z. Li, H. Y. Shum, H. Zhang, "Local nonnegative matrix factorization as a visual representation," *Proceedings of the 2nd International Conference on Development and Learning*, 2002.

[12] D. Cai, X. He, J. Han, T. Huang, "Graph regularized non-negative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548-1560, Aug. 2011.

[13] C. Ding, T. Li, M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 45-55, Jan. 2010.

[14] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, pp. 281-297, 1967.

[15] D. Guillamet, J. Vitria, “Nonnegative matrix factorization for face recognition,” *Proceedings of the 5th Catalonian Conference on AI: Topics in Artificial Intelligence*, pp. 336-344, 2002.

[16] J. P. Brunet, P. Tamayo, T. R. Golub, J. P. Mesirov, “Metagenes and molecular pattern discovery using matrix factorization,” *Proceeding in National Academy of Science of USA*, vol. 101, pp. 4164-4169, Mar. 2004.

[17] W. Xu, X. Liu, Y. Gong, “Document clustering based on non-negative matrix factorization,” *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 267-273, 2003.

[18] F. Shahnaza, M.W. Berrya, V. Paucab, and R.J. Plemmonsb, “Document clustering using nonnegative matrix factorization,” *Information Processing and Management*, vol. 42, no. 2, pp. 373-386, Mar. 2006.

[19] V. P. Pauca, F. Shahnaz, M. Berry, R. Plemmons, “Text mining using non-negative matrix factorization,” *Proceeding in International Conference on Data Mining*, pp. 452-456, 2004.

[20] A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmann, R. D. Pascual-Marqui, “Nonsmooth nonnegative matrix factorization (nsNMF),” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 3, pp. 403-415, Mar. 2006.

[21] B. Gao, W. L. Woo, S. S. Dlay, “Variational regularized 2-D nonnegative matrix factorization,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 5, pp. 703-716, May 2012.

[22] D. D. Lee, H. S. Seung, “Algorithms for non-negative matrix factorization,” *Advances in Neural Information Processing Systems 13*, MIT Press, 2001.

[23] X. He, P. Niyogi, “Locality preserving projections,” *Proceeding in Conference Advances in Neural Information Processing Systems (NIPS)*, 2003.

[24] M. Belkin, P. Niyogi, “Laplacian eigenmaps and spectral techniques for embedding and clustering,” *Advances in Neural Information Processing Systems 14*, pp. 585-591, MIT Press, 2001.

[25] Fan R. K. Chung, Spectral Graph Theory, *Regional Conference Series in Mathematics*, no. 92, 1997.

[26] X. He, D. Cai, S. Yan, H. Zhang, “Neighborhood preserving embedding,” *Proceeding in International Conference on Computer Vision (ICCV)*, 2005.

[27] P. O. Hoyer, “Nonnegative matrix factorization with sparseness constraints,” *The Journal of Machine Learning Research*, vol. 5, pp. 1457-1469, 2004.

[28] J. Shi, J. Malik, “Normalized cuts and image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, Aug. 2000.

[29] A. Frank, A. Asuncion (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[30] J. J. Hull, “A database for handwritten text recognition research,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 550-554, May 1994.

[31] L. Lovasz, M. Plummer, Matching Theory. Akadé miai Kiadó, 1986.

[32] D. Cai, X. He, J. Han, “Document clustering using locality preserving indexing,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 12, pp. 1624-1637, Dec. 2005.

[33] X. He, D. Cai, P. Niyogi, “Laplacian score for feature selection,” *Advances in Neural Information Processing Systems 18*, MIT Press, 2005.

[34] V. Bittorf, B. Recht, E. Ré, and J. A. Tropp, “Factoring nonnegative matrices with linear programs,” *Advances in Neural Information Processing Systems 25*, pp. 1223-1231, 2012.

[35] N. Gillis, R. Lucey, “Robust near-separable nonnegative matrix factorization using linear optimization,” *Journal of Machine Learning Research*, vol.15, pp. 1249-1280, 2014.

[36] H. Kim, H. Park, “Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis,” *Bioinformatics*, vol.23, no.12, pp. 1495-1502, 2007.

[37] W. Liu, N. Zheng, X. Lu, “Nonnegative matrix factorization for visual coding,” *Proceeding in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003)*, 2003.

[38] R. Tandon, S. Sra, “Sparse nonnegative matrix approximation: new formulations and algorithms,” *Max Planck Institute for Biological Cybernetics*, Technical Report, no.193, 2010.