SIAM J. MATRIX ANAL. APPL.
Vol. 39, No. 2, pp. 856–875

ORTHOGONAL NONNEGATIVE MATRIX FACTORIZATION BY SPARSITY AND NUCLEAR NORM OPTIMIZATION*

JUNJUN PAN† AND MICHAEL K. NG‡

Abstract. In this paper, we study orthogonal nonnegative matrix factorization. We demonstrate the coefficient matrix can be sparse and low-rank in the orthogonal nonnegative matrix factorization. By using these properties, we propose to use a sparsity and nuclear norm minimization for the factorization and develop a convex optimization model for finding the coefficient matrix in the factorization. Numerical examples including synthetic and real-world data sets are presented to illustrate the effectiveness of the proposed algorithm and demonstrate that its performance is better than other testing methods.

Key words. orthogonal nonnegative matrix factorization, sparsity, nuclear norm, convex optimization, document clustering, hyperspectral image unmixing

AMS subject classification. 65F30

DOI. 10.1137/16M1107863

1. Introduction.

Nonnegative matrix factorization (NMF) is an important research problem in matrix computation. There are many scientific and engineering applications, for example, clustering [4, 8, 10, 11, 12, 24, 33], text mining [1, 24, 32, 36], image processing [19, 25, 26], face recognition [7, 16, 17, 34, 38], gene expression classification [8, 10, 14, 20, 21, 31, 35], and so on. In [30], Paatero and Tapper used a positive matrix factorization for data analysis arising from environmental applications. In [22, 23], Lee and Seung proposed NMF algorithms for image data analysis and applications. They showed that NMF contains fundamental components from the data decomposition. In [12], Ding et al. further pointed out the connection between NMF and many clustering models.

NMF aims to decompose an input nonnegative matrix $\mathbf{A} \in \mathbb{R}_+^{m\times n}$ into two nonnegative matrices $\mathbf{B} \in \mathbb{R}_+^{m\times r}$ and $\mathbf{C} \in \mathbb{R}_+^{r\times n}$:

$$\mathbf{A} \approx \mathbf{BC}. \tag{1}$$

Here r is smaller than $\min(m, n)$. In the literature, there are many algorithms for finding such NMF. In [22, 23], Lee and Seung proposed to solve NMF by using the multiplicative update algorithm. In [37], Yuan and Oja considered that each data instance/feature vector is represented as a column of $\mathbf{A}$ and studied a projective NMF. Their proposed minimization problem is given as follows:

$$\min_{\mathbf{B}\geq\mathbf{0}} \ \|\mathbf{A} - \mathbf{BB}^T\mathbf{A}\|_F^2, \tag{2}$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix, $\mathbf{B}^T$ is the transpose of $\mathbf{B}$, and $\mathbf{B} \geq \mathbf{0}$ means that each entry of $\mathbf{B}$ is nonnegative. Their basic idea is to find a subspace

*Received by the editors December 13, 2016; accepted for publication (in revised form) by M. W. Berry March 9, 2018; published electronically May 22, 2018.
http://www.siam.org/journals/simax/39-2/M110786.html
Funding: The work of the second author was supported in part by HKRGC GRF 12302715, 12306616, and 12200317 and HKBU RC-ICRS/16-17/03.
†Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong (panjun 009@gmail.com).
‡Corresponding author. Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong (mng@math.hkbu.edu.hk).

and a projection matrix $\mathbf{P}_A$ (e.g., it is equal to $\mathbf{B}\mathbf{B}^T$) that projects $\mathbf{A}$ onto the subspace. The projection matrix should keep $\mathbf{P}_A\mathbf{A}$ to be nonnegative and minimize the difference between $\mathbf{A}$ and $\mathbf{P}_A\mathbf{A}$ as well.

In general, there may be many possible solutions in (1). It is necessary to impose additional constraints for determining NMF. In some applications, sparsity, orthogonality, or smoothness constraints are incorporated into minimization models for determining both $\mathbf{B}$ and $\mathbf{C}$. It is interesting to design suitable optimization techniques to solve the related minimization problems. In particular, the multiplicative update algorithms [9, 10, 12] are modified to tackle the corresponding minimization problems. Among various constraints imposed into NMF, the use of orthogonality constraints can provide a very effective representation of $\mathbf{B}$ and $\mathbf{C}$ which is useful in data clustering applications [10, 12]. The orthogonal nonnegative matrix factorization (ONMF) is given as follows:

$$\min_{\mathbf{B}\geq\mathbf{0},\mathbf{C}\geq\mathbf{0}} \|\mathbf{A}-\mathbf{B}\mathbf{C}\|_F^2 \quad \text{subject to} \quad \mathbf{B}^T\mathbf{B}=\mathbf{I}. \tag{3}$$

Here $\mathbf{A}$ is an m-by-n data matrix. Each row of $\mathbf{A}$ in (3) represents a data instance described by n features. There are m instances for data and cluster analysis. We note that the ONMF in (3) is different from the projective NMF in (2).

In [12], Ding et al. pointed out that the ONMF problem is equivalent to k-means clustering and showed that if $\mathbf{B}$ and $\mathbf{C}$ are the solution of (3), then there exists no other solution matrices unless the matrices can be written as $\mathbf{B}\mathbf{Q}$ and $\mathbf{Q}^T\mathbf{C}$, where $\mathbf{Q}$ is a permutation matrix. Ding et al. further studied an optimization method for solving $\mathbf{B}$ and $\mathbf{C}$ and extended the multiplicative update algorithm for orthogonal nonnegative matrix tri-factorizations. In [9], Choi developed a multiplicative update algorithm based on the gradient calculation in Stiefel manifold. The main issue of the above-mentioned ONMF algorithms is that their solutions can be sensitive to initial guesses.

In this paper, we first study ONMF when $\mathbf{A}$ is decomposed into the coefficient matrix $\mathbf{B}$ and the basis matrix $\mathbf{C}$ exactly; i.e., $\mathbf{A}=\mathbf{B}\mathbf{C}$. As $\mathbf{B}$ is orthogonal, we know that $\mathbf{C}$ is equal to $\mathbf{B}^T\mathbf{A}$, and therefore we obtain $\mathbf{A}=\mathbf{B}\mathbf{B}^T\mathbf{A}$. Instead of solving (3), we propose and consider the following optimization problem:

$$\min_{\mathbf{B}\geq\mathbf{0}} \|\mathbf{A}-\mathbf{B}\mathbf{B}^T\mathbf{A}\|_F^2 \quad \text{subject to} \quad \mathbf{B}^T\mathbf{B}=\mathbf{I}. \tag{4}$$

We remark that each data instance/feature vector is represented as a row of $\mathbf{A}$ in (4), while each data instance/feature vector is represented as a column of $\mathbf{A}$ in (2). The formulation in (4) is different from that in (2).

On the other hand, we show that if $\mathbf{A}$ is orthogonally decomposable, then $\mathbf{B}\mathbf{B}^T$ has a special block structure. We can make use of sparsity and nuclear norm minimization to incorporate such block structure of $\mathbf{K}=\mathbf{B}\mathbf{B}^T$. We formulate a convex optimization problem to determine $\mathbf{K}$. We can employ the alternating direction method of multipliers to solve such an optimization problem quite efficiently. Moreover, the coefficient matrix $\mathbf{B}$ can be recovered from eigenvectors of $\mathbf{K}$, and then $\mathbf{B}$ can be used for clustering analysis. Both synthetic and real-world data sets are used to test the effectiveness of the proposed sparsity and nuclear norm minimization orthogonal nonnegative matrix factorization (SN-ONMF) method. The numerical results are presented that the performance of the proposed model is better than that of other testing methods.

The outline of this paper is given as follows. In section 2, we discuss some properties of $\mathbf{B}$ and $\mathbf{C}$ in ONMF. In section 3, we present the algorithm for the proposed

model. In section 4, numerical examples including synthetic data and real-world data sets are given to demonstrate the performance of the proposed method. It shows better performance than that of other testing methods. Finally, some concluding remarks are given in section 5.

2. The properties of ONMF. In the section, we consider some properties of ONMF for the factorization matrices $\mathbf{B}$ and $\mathbf{C}$. We first establish the results for the coefficient matrix $\mathbf{B}$ when $\mathbf{A}$ is orthogonally decomposable.

LEMMA 1. *Suppose* $\mathbf{B} \geq \mathbf{0}$ *and* $\mathbf{B}$ *is orthogonal. Each row of* $\mathbf{B}$ *has at most one nonzero element.*

Proof. As $\mathbf{B}^T\mathbf{B} = \mathbf{I}$, we have

$$\sum_{i=1}^{m} b_{i,j_1} b_{i,j_2} = 0 \quad \forall 1 \leq j_1 \neq j_2 \leq r. \tag{5}$$

Because $\mathbf{B} \geq \mathbf{0}$, it implies

$$b_{i,j_1} b_{i,j_2} = 0 \quad \forall 1 \leq i \leq m,\ 1 \leq j_1 \neq j_2 \leq r. \tag{6}$$

According to (6), we know that $\mathbf{B}$ must have at most one nonzero element. It is clear that each nonzero element is less than or equal to 1. ∎

Without loss of generality, we consider $\mathbf{B}$ with the following structure:

$$\mathbf{B} = \begin{pmatrix} \mathbf{b}_1 & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{b}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{b}_{r-1} & \mathbf{0} \\ \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{b}_r \end{pmatrix}. \tag{7}$$

We note that if $\mathbf{B}$ is not given in the above form, we make use of the permutation matrices $\mathbf{P}$ and $\mathbf{Q}$ such that $\mathbf{PBQ}$ is in the form of (7) by using the result in Lemma 1. Here $\mathbf{PBQ}$ is still orthogonal. When $\mathbf{A}$ is orthogonal decomposable, we study and obtain the ONMF of $\mathbf{PA}$: $\mathbf{PA} = (\mathbf{PBQ})(\mathbf{Q}^T\mathbf{C})$.

Suppose the jth column $\mathbf{b}_j$ of $\mathbf{B}$ has l_j nonzero elements, i.e.,

$$\mathbf{b}_j = \begin{pmatrix} b_{s+1,j} \\ b_{s+2,j} \\ \vdots \\ b_{s+l_j,j} \end{pmatrix},$$

where $s = l_1 + l_2 + \cdots + l_{j-1}$. Because of the orthogonality of $\mathbf{B}$, we have

$$\mathbf{b}_j^T \mathbf{b}_j = 1 \quad \forall j = 1, 2, \ldots, r.$$

It is clear that if there is a zero entry in $\mathbf{b}_j$, then all the entries in the corresponding row of $\mathbf{A}$ are zero. By using the partitioning of $\mathbf{B}$ to $\mathbf{A}$, we obtain

$$\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 \\ \mathbf{A}_2 \\ \vdots \\ \mathbf{A}_r \end{pmatrix}, \tag{8}$$

where $\mathbf{A}_j \in \mathbb{R}^{l_j \times n}$,

$$\mathbf{A}_j = \begin{pmatrix} a_{s+1,1} & \cdots & \cdots & a_{s+1,n} \\ a_{s+2,1} & \cdots & \cdots & a_{s+2,n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{s+l_j,1} & \cdots & \cdots & a_{s+l_j,n} \end{pmatrix}, \tag{9}$$

with $s = l_1 + l_2 + \cdots + l_{j-1}$. Next we show that any row of $\mathbf{A}_j$ can be expressed as a constant multiple of the other rows in $\mathbf{A}_j$.

THEOREM 1. *Suppose* $\mathbf{A}$ *is orthogonal decomposable (*$\mathbf{A} = \mathbf{BC}$*) and* $\mathbf{B}$ *is given by* (7) *and* $\mathbf{A}$ *can be partitioned into* r *groups as in* (8)*. Then*

$$\mathbf{a}_{s+t_1}^T = \frac{b_{s+t_1,j}}{b_{s+t_2,j}} \mathbf{a}_{s+t_2}^T \quad \text{and} \quad \tilde{\mathbf{a}}_{p_1} = \frac{c_{j,p_1}}{c_{j,p_2}} \tilde{\mathbf{a}}_{p_2}$$

with $1 \le t_1 \ne t_2 \le l_j$, $1 \le p_1 \ne p_2 \le n$, *and* $s = l_1 + l_2 + \cdots + l_{j-1}$, *where*

$$\mathbf{a}_{s+t}^T = [a_{s+t,1}, \ldots, a_{s+t,n}] \quad \text{and} \quad \tilde{\mathbf{a}}_p = [a_{s+1,p}, \ldots, a_{s+l_j,p}]^T.$$

Moreover, $\|\mathbf{A}_j\|_F^2 = \|\mathbf{c}_j\|_2^2$, *where* $\mathbf{c}_j$ *is the* j*th row of* $\mathbf{C}$ *and* $\|\cdot\|_2$ *is the Euclidean norm of a vector.*

Proof. Since $\mathbf{A}$ is orthogonal decomposable, we obtain

$$\begin{cases} a_{s+1,p} = \displaystyle\sum_{t=1}^{r} b_{s+1,t} c_{t,p} = b_{s+1,j} c_{j,p}, & p = 1, 2, \ldots, n \\ \qquad\qquad \vdots & \qquad\qquad \vdots \\ a_{s+l_j,p} = \displaystyle\sum_{t=1}^{r} b_{s+l_j,t} c_{t,p} = b_{s+l_j,j} c_{j,p}, & p = 1, 2, \ldots, n. \end{cases}$$

By using Lemma 1, it follows that $\mathbf{A}_j$ can be given by

$$\mathbf{A}_j = \begin{pmatrix} b_{s+1,j}c_{j,1} & \cdots & \cdots & b_{s+1,j}c_{j,n} \\ b_{s+2,j}c_{j,1} & \cdots & \cdots & b_{s+2,j}c_{j,n} \\ \vdots & \vdots & \vdots & \vdots \\ b_{s+l_j,j}c_{j,1} & \cdots & \cdots & b_{s+l_j,j}c_{j,n} \end{pmatrix}. \tag{10}$$

For $1 \le t_1 \ne t_2 \le l_j$, we have

$$\mathbf{a}_{s+t_1}^T = [b_{s+t_1,j}c_{j,1}, b_{s+t_1,j}c_{j,2}, \ldots, b_{s+t_1,j}c_{j,n}] = b_{s+t_1,j}[c_{j,1}, c_{j,2}, \ldots, c_{j,n}]$$

and

$$\mathbf{a}_{s+t_2}^T = [b_{s+t_2,j}c_{j,1}, b_{s+t_2,j}c_{j,2}, \ldots, b_{s+t_2,j}c_{j,n}] = b_{s+t_2,j}[c_{j,1}, c_{j,2}, \ldots, c_{j,n}]$$

with $s = l_1 + l_2 + \cdots + l_{j-1}$. Similarly, for $1 \le p_1 \ne p_2 \le n$, we obtain

$$\tilde{\mathbf{a}}_{p_1} = [b_{s+1,j}c_{j,p_1}, b_{s+2,j}c_{j,p_1}, \ldots, b_{s+l_j,j}c_{j,p_1}]^T = [b_{s+1,j}, b_{s+2,j}, \ldots, b_{s+l_j,j}]^T c_{j,p_1}$$

and

$$\tilde{\mathbf{a}}_{p_2} = [b_{s+1,j}c_{j,p_2}, b_{s+2,j}c_{j,p_2}, \ldots, b_{s+l_j,j}c_{j,p_2}]^T = [b_{s+1,j}, b_{s+2,j}, \ldots, b_{s+l_j,j}]^T c_{j,p_2}.$$

Hence, the result follows.

By using (10), we get

$$\sum_{p=1}^{n}\sum_{q=1}^{l_j} a_{s+q,p}^2 = \sum_{p=1}^{n}\sum_{q=1}^{l_j} b_{s+q,j}^2 c_{j,p}^2 = \sum_{p=1}^{n} c_{j,p}^2.$$

The above second equality is established because $\mathbf{b}_j^T\mathbf{b}_j = 1$; hence, $\|\mathbf{A}_j\|_F^2 = \|\mathbf{c}_j\|_2^2$. ∎

According to Theorem 1, we remark that each row of $\mathbf{A}$ is an object represented by n attributes and can be partitioned into one of r groups. This is the reason that ONMF can be used for data clustering. Next we see that the jth column $\mathbf{b}_j$ of B can be determined exactly.

THEOREM 2. *Suppose* $\mathbf{A}$ *is orthogonal decomposable (*$\mathbf{A} = \mathbf{BC}$*) and* $\mathbf{B}$ *is given by* (7) *and* $\mathbf{A}$ *can be partitioned into* r *groups as in* (8)*. If* $a_{s+1,1}$ *is nonzero, then*

$$b_{s+t,j} = \frac{\gamma_t}{\gamma}, \quad 1 \le t \le l_j,$$

with $s = l_1 + l_2 + \cdots + l_{j-1}$,

$$\gamma_t = \frac{a_{s+t,1}}{a_{s+1,1}}, \quad (1 \le t \le l_j), \quad \text{and} \quad \gamma = \sqrt{\sum_{t=1}^{l_j} \gamma_t^2}.$$

Proof. From (10), the first column $\tilde{\mathbf{a}}_1$ of $\mathbf{A}_j$,

$$\tilde{\mathbf{a}}_1 = [a_{s+1,1}, a_{s+2,1}, \ldots, a_{s+l_j,1}]^T = [b_{s+1,j}c_{j,1}, b_{s+2,j}c_{j,1}, \ldots, b_{s+l_j,j}c_{j,1}]^T,$$

we deduce that for $1 \le t \le l_j$,

$$\frac{b_{s+t,j}}{b_{s+1,j}} = \frac{a_{s+t,1}}{a_{s+1,1}}.$$

$\mathbf{B}^T\mathbf{B} = \mathbf{I}$, which implies $\sum_{t=1}^{l_j} b_{s+t,j}^2 = 1$. For $j = 1, \ldots, r$, we can construct $\mathbf{b}_j$ as

$$b_{s+t,j} = \frac{\gamma_t}{\gamma}, \quad 1 \le t \le l_j.$$

The result follows. ∎

In the following, we give some properties of $\mathbf{BB}^T$.

THEOREM 3. *Suppose* $\mathbf{B} \ge \mathbf{0}$*,* $\mathbf{B}$ *is orthogonal and is given by* (7)*. Then* $\mathbf{K} = \mathbf{BB}^T$ *satisfies*

$$\mathbf{K}^T = \mathbf{K} \quad \text{and} \quad \mathbf{K}^2 = \mathbf{K} \quad \text{with} \quad 0 \le k_{i,j} \le 1.$$

Moreover, $\|\mathbf{K}\|_F^2 = r$*,* $\|\mathbf{K}\|_1 \le m$*, where* m *and* r *are the dimensions of* $\mathbf{B}$ *and* $\|\cdot\|_1$ *is the* 1*-norm of a vector. There are* r *repeated eigenvalues of* 1 *for* $\mathbf{K}$*, and their corresponding eigenvectors are given by the columns of* $\mathbf{B}$*.*

Proof. We note that $\mathbf{K}^T = \mathbf{BB}^T = \mathbf{K}$ and $\mathbf{K}^2 = \mathbf{BB}^T\mathbf{BB}^T = \mathbf{BB}^T = \mathbf{K}$. Also, $\|\mathbf{K}\|_F^2 = trace(\mathbf{K}^T\mathbf{K}) = trace(\mathbf{BB}^T\mathbf{BB}^T) = trace(\mathbf{B}^T\mathbf{BB}^T\mathbf{B}) = r$. Moreover, $\mathbf{K}$ has the following block structure:

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{K}_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{K}_r \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1\mathbf{b}_1^T & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{b}_2\mathbf{b}_2^T & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{b}_r\mathbf{b}_r^T \end{pmatrix}, \tag{11}$$

where $\mathbf{K}_j = \mathbf{b}_j\mathbf{b}_j^T \in \mathbb{R}^{l_j \times l_j}$ and $\sum_{j=1}^{r} l_j = m$. Here

$$\mathbf{K}_j = \begin{pmatrix} b_{1,j}^2 & b_{1,j}b_{2,j} & \cdots & b_{1,j}b_{l_j,j} \\ b_{2,j}b_{1,j} & b_{2,j}^2 & \cdots & b_{2,j}b_{l_j,j} \\ \vdots & \vdots & \ddots & \vdots \\ b_{l_j,j}b_{1,j} & b_{l_j,j}b_{2,j} & \cdots & b_{l_j,j}^2 \end{pmatrix}.$$

By Lemma 1, the nonzero element of B is less than or equal to 1. Therefore, $0 \le k_{i,j} \le 1$.

On the other hand, the 1-norm of $\mathbf{K}_j$ is the sum of the magnitude of all the entries of $\mathbf{K}_j$, i.e.,

$$\|\mathbf{K}_j\|_1 = (b_{1,j} + b_{2,j} + \cdots + b_{l_j,j})^2.$$

Here we can study the maximum value of $\|\mathbf{K}_j\|_1$ under the condition that $\mathbf{b}_j^T\mathbf{b}_j = 1$. We know its maximum value can be achieved when $b_{1,j} = b_{2,j} = \cdots = b_{l_j,j} = \frac{1}{\sqrt{l_j}}$, and it is given by $\|\mathbf{K}_j\|_1 = (\sqrt{l_j})^2 = l_j$. The bound of $\|\mathbf{K}\|_1$ can be calculated as follows:

$$\|\mathbf{K}\|_1 = \|\mathbf{K}_1\|_1 + \|\mathbf{K}_2\|_1 + \cdots + \|\mathbf{K}_r\|_1 \le \sum_{j=1}^{r} l_j = m.$$

Moreover, $\mathbf{K} = \mathbf{B}\mathbf{B}^T$ means that $\mathbf{K}\mathbf{B} = \mathbf{B}$. It implies that the columns of $\mathbf{B}$ are the eigenvectors corresponding to the eigenvalue 1. ∎

3. The optimization problem. Based on the results of Theorems 1 and 3, these motivate us to solve the orthogonal decomposition of $\mathbf{A}$ by using $\mathbf{B}\mathbf{B}^T$. More precisely, when $\mathbf{A} = \mathbf{B}\mathbf{C}$, we obtain $\mathbf{C} = \mathbf{B}^T\mathbf{A}$, i.e., $\mathbf{A} = \mathbf{B}\mathbf{B}^T\mathbf{A} = \mathbf{K}\mathbf{A}$. We would like to determine a matrix $\mathbf{K}$ such that $\mathbf{A} \approx \mathbf{K}\mathbf{A}$ and $\mathbf{K}$ satisfies the constraints stated in Theorem 3. However, the constraint $\mathbf{K}^T\mathbf{K} = \mathbf{K}$ is not linear. We drop it in the proposed formulation. Here we propose to use the following convex optimization model to determine $\mathbf{K}$ for the orthogonal decomposition of $\mathbf{A}$:

$$\begin{aligned} &\min_{\mathbf{K}} \ F(\mathbf{K}) = \frac{1}{2}\|\mathbf{A} - \mathbf{K}\mathbf{A}\|_F^2 + \theta\|\mathbf{K}\|_1 + \beta\|\mathbf{K}\|_* \\ &\text{subject to} \quad \mathbf{K} = \mathbf{K}^T, \quad \mathbf{K} \ge 0, \end{aligned} \tag{12}$$

where $\|\mathbf{K}\|_*$ is the nuclear norm of $\mathbf{K}$, $\|\mathbf{K}\|_1$ refers to the summation of the magnitudes of all the entries of $\mathbf{K}$, and θ and β are the positive numbers to control the balance among the three terms in the objective function. The use of $\|\mathbf{K}\|_1$ is to enforce the sparsity of $\mathbf{K}$, and the use of $\|\mathbf{K}\|_*$ is to require the small Frobenius norm of $\mathbf{K}$. We remark from Theorem 3 that $\|\mathbf{K}\|_1$ is bounded by m and the sum of singular values of $\mathbf{K}$ is equal to r when A is orthogonal decomposable.

By introducing two variables $\mathbf{X}$ and $\mathbf{Z}$, the optimization model (12) can be rewritten as

$$\begin{aligned} &\min_{\mathbf{K},\mathbf{X},\mathbf{Z}} \ F(\mathbf{K},\mathbf{X},\mathbf{Z}) = \frac{1}{2}\|\mathbf{A} - \mathbf{K}\mathbf{A}\|_F^2 + \theta\|\mathbf{Z}\|_1 + \beta\|\mathbf{X}\|_* \\ &\text{subject to} \quad \mathbf{K} = \mathbf{K}^T, \quad \mathbf{X} - \mathbf{K} = \mathbf{0}, \quad \mathbf{Z} \ge \mathbf{0}, \quad \mathbf{Z} - \mathbf{K} = \mathbf{0}. \end{aligned} \tag{13}$$

The objective function in (13) is a sum of a function of $\mathbf{K}$ and a function of $(\mathbf{X}, \mathbf{Z})$. The alternating direction method of multipliers can be employed to solve (13). The augmented Lagrangian function of (13) is given as follows:

$$\begin{aligned}&\tilde{L}(\mathbf{K},\mathbf{X},\mathbf{Z},\mathbf{\Lambda}_1,\mathbf{\Lambda}_2,\mathbf{\Lambda}_3)\\&=\frac{1}{2}\|\mathbf{A}-\mathbf{K}\mathbf{A}\|_F^2+\theta\|\mathbf{Z}\|_1+\beta\|\mathbf{X}\|_*+<\mathbf{\Lambda}_1,\mathbf{K}-\mathbf{K}^T>+<\mathbf{\Lambda}_2,\mathbf{X}-\mathbf{K}>\\&\quad+<\mathbf{\Lambda}_3,\mathbf{Z}-\mathbf{K}>+\frac{\mu_1}{2}\|\mathbf{K}-\mathbf{K}^T\|_F^2+\frac{\mu_2}{2}\|\mathbf{X}-\mathbf{K}\|_F^2+\frac{\mu_3}{2}\|\mathbf{Z}-\mathbf{K}\|_F^2,\end{aligned}\tag{14}$$

where $<\cdot,\cdot>$ is the inner product between two matrices; $\mathbf{\Lambda}_1$, $\mathbf{\Lambda}_2$, and $\mathbf{\Lambda}_3$ are the Lagrangian multipliers; and μ_1, μ_2, and μ_3 are the positive penalty parameters. We can further rewrite (14) as follows:

$$\begin{aligned}&L(\mathbf{K},\mathbf{X},\mathbf{Z},\mathbf{\Lambda}_1,\mathbf{\Lambda}_2,\mathbf{\Lambda}_3)\\&=\frac{1}{2}\|\mathbf{A}-\mathbf{K}\mathbf{A}\|_F^2+\theta\|\mathbf{Z}\|_1+\beta\|\mathbf{X}\|_*+\frac{\mu_1}{2}\|\mathbf{K}-\mathbf{K}^T+\frac{\mathbf{\Lambda}_1}{\mu_1}\|_F^2\\&\quad+\frac{\mu_2}{2}\|\mathbf{X}-\mathbf{K}+\frac{\mathbf{\Lambda_2}}{\mu_2}\|_F^2+\frac{\mu_3}{2}\|\mathbf{Z}-\mathbf{K}+\frac{\mathbf{\Lambda}_3}{\mu_3}\|_F^2.\end{aligned}\tag{15}$$

The alternating direction method of multipliers solves (13) by using the following procedure:

$$\mathbf{K}^{(i+1)}=\arg\min_{\mathbf{K}}L(\mathbf{K},\mathbf{X}^{(i)},\mathbf{Z}^{(i)},\mathbf{\Lambda}_1^{(i)},\mathbf{\Lambda}_2^{(i)},\mathbf{\Lambda}_3^{(i)})\tag{16}$$

$$(\mathbf{X}^{(i+1)},\mathbf{Z}^{(i+1)})=\arg\min_{\mathbf{X},\mathbf{Z}}L(\mathbf{K}^{(i+1)},\mathbf{X},\mathbf{Z},\mathbf{\Lambda}_1^{(i)},\mathbf{\Lambda}_2^{(i)},\mathbf{\Lambda}_3^{(i)})\tag{17}$$

$$\mathbf{\Lambda}_1^{(i+1)}=\mathbf{\Lambda}_1^{(i)}+\mu_1(\mathbf{K}^{(i+1)}-\mathbf{K}^{(i+1)T})\tag{18}$$

$$\mathbf{\Lambda}_2^{(i+1)}=\mathbf{\Lambda}_2^{(i)}+\mu_2(\mathbf{X}^{(i+1)}-\mathbf{K}^{(i+1)})\tag{19}$$

$$\mathbf{\Lambda}_3^{(i+1)}=\mathbf{\Lambda}_3^{(i)}+\mu_3(\mathbf{Z}^{(i+1)}-\mathbf{K}^{(i+1)})\tag{20}$$

where $\mathbf{K}^{(i)}$, $\mathbf{X}^{(i)}$, $\mathbf{Z}^{(i)}$, $\mathbf{\Lambda}_1^{(i)}$, $\mathbf{\Lambda}_2^{(i)}$, and $\mathbf{\Lambda}_3^{(i)}$ are the iterates of the unknown variables $\mathbf{K}$, $\mathbf{X}$, $\mathbf{Z}$, $\mathbf{\Lambda}_1$, $\mathbf{\Lambda}_2$, and $\mathbf{\Lambda}_3$, respectively.

We fix $\mathbf{X}^{(i)}$ and $\mathbf{Z}^{(i)}$ and solve for $\mathbf{K}$ in (16) as follows:

$$\begin{aligned}\mathbf{K}^{(i+1)}=\arg\min_{\mathbf{K}}\Bigg\{&\frac{1}{2}\|\mathbf{A}-\mathbf{K}\mathbf{A}\|_F^2+\frac{\mu_1}{2}\|\mathbf{K}-\mathbf{K}^T+\frac{\mathbf{\Lambda}_1^{(i)}}{\mu_1}\|_F^2+\\&\frac{\mu_2}{2}\|\mathbf{X}^{(i)}-\mathbf{K}+\frac{\mathbf{\Lambda}_2^{(i)}}{\mu_2}\|_F^2+\frac{\mu_3}{2}\|\mathbf{Z}^{(i)}-\mathbf{K}+\frac{\mathbf{\Lambda}_3^{(i)}}{\mu_3}\|_F^2\Bigg\}.\end{aligned}\tag{21}$$

It is equivalent to solving the following linear system:

$$\mathbf{K}\mathbf{A}\mathbf{A}^T+\mu_2\mathbf{K}+\mu_3\mathbf{K}=\mathbf{A}\mathbf{A}^T-2\mu_1(\mathbf{K}-\mathbf{K}^T)-(\mathbf{\Lambda}_1^{(i)}-\mathbf{\Lambda}_1^{(i)T})+\mu_2\mathbf{X}^{(i)}+\mathbf{\Lambda}_2^{(i)}+\mu_3\mathbf{Z}^{(i)}+\mathbf{\Lambda}_3^{(i)}.$$

Because μ_2 and μ_3 are positive numbers, the corresponding coefficient matrix is symmetric positive definite.

For the subproblem in (17), $\mathbf{X}$ and $\mathbf{K}$ are separable. Therefore, we solve the optimization problem:

$$\mathbf{X}^{(i+1)}=\arg\min_{\mathbf{X}}\left\{\beta\|\mathbf{X}\|_*+\frac{\mu_2}{2}\left\|\mathbf{X}-\mathbf{K}^{(i+1)}+\frac{\mathbf{\Lambda}_2^{(i)}}{\mu_2}\right\|_F^2\right\}.$$

Suppose the singular value decomposition of matrix $\mathbf{K}^{(i+1)} - \frac{\mathbf{\Lambda}_2^{(i)}}{\mu_2}$ is given by

$$\mathbf{K}^{(i+1)} - \frac{\mathbf{\Lambda}_2^{(i)}}{\mu_2} = \mathbf{U}^{(i+1)}\mathbf{\Sigma}^{(i+1)}\mathbf{V}^{(i+1)T}$$

with $\mathbf{\Sigma}^{(i+1)} = diag(\sigma_j^{(i+1)})$ and we obtain

$$\mathbf{X}^{(i+1)} = \mathbf{U}_\tau^{(i+1)}\mathbf{\Sigma}_\tau^{(i+1)}\mathbf{V}_\tau^{(i+1)T},$$

where $\tau = \frac{\beta}{\mu_2}$, $\mathbf{\Sigma}_\tau^{(i+1)} = diag(\{\sigma_j^{(i+1)} - \tau\}_+)$, $\mathbf{U}_\tau^{(i+1)}$, $\mathbf{V}_\tau^{(i+1)}$ are the singular vectors of $\mathbf{K}^{(i+1)} - \frac{\mathbf{\Lambda}_2^{(i)}}{\mu_2}$ associated with the singular values greater than or equal to τ; see [6]. On the other hand, we consider the following optimization problem for $\mathbf{Z}$:

$$\mathbf{Z}^{(i+1)} = \arg\min_{\mathbf{Z}\geq\mathbf{0}} \left\{ \theta\|\mathbf{Z}\|_1 + \frac{\mu_3}{2}\left\|\mathbf{Z} - \mathbf{K}^{(i+1)} + \frac{\mathbf{\Lambda}_3^{(i)}}{\mu_3}\right\|_F^2 \right\}.$$

The corresponding solution can be given as follows:

$$\mathbf{Z}^{(i+1)} = \max\{sgn(\mathbf{W}^{(i+1)})(|\mathbf{W}^{(i+1)}| - \frac{\theta}{\mu_3}), 0\},$$

where $\mathbf{W}^{(i+1)} = \mathbf{K}^{(i+1)} - \frac{\mathbf{\Lambda}_3^{(i)}}{\mu_3}$, $sgn(\mathbf{W}^{(i+1)})$ is the entrywise sign function on $\mathbf{W}^{(i+1)}$, and $|\mathbf{W}^{(i+1)}|$ is the entrywise magnitude of $\mathbf{W}^{(i+1)}$. More precisely, we have

$$\begin{cases} z_{l,t} = w_{l,t} - \frac{\theta}{\mu_3} & \text{if } w_{l,t} \geq \frac{\theta}{\mu_3}; \\ z_{l,t} = w_{l,t} + \frac{\theta}{\mu_3} & \text{if } w_{l,t} < -\frac{\theta}{\mu_3}; \\ z_{l,t} = 0 & \text{otherwise.} \end{cases}$$

We note that when θ is a very large positive number, $z_{l,t}$ will be forced to be zero.

In the alternating direction method of multipliers, there are only two blocks of variables ($\mathbf{K}$ and $(\mathbf{X}, \mathbf{Z})$) updated in each iteration. The convergence of the algorithm can be guaranteed; see [13, 15].

THEOREM 4. *For $\mu_1 > 0$, $\mu_2 > 0$, $\mu_3 > 0$, the sequence $\{(\mathbf{K}^{(i)}, \mathbf{X}^{(i)}, \mathbf{Z}^{(i)}, \mathbf{\Lambda}_1^{(i)}, \mathbf{\Lambda}_2^{(i)}, \mathbf{\Lambda}_3^{(i)})\}$ generated by Algorithm 1 from any initial point would converge to $\{(\mathbf{K}^*, \mathbf{X}^*, \mathbf{Z}^*, \mathbf{\Lambda}_1^*, \mathbf{\Lambda}_2^*, \mathbf{\Lambda}_3^*)\}$, where $(\mathbf{K}^*, \mathbf{X}^*, \mathbf{Z}^*)$ is a solution of* (13).

Here we summarize the above computational procedure in Algorithm 1.

When $\mathbf{A}$ is orthogonally decomposable, we note that the rank of each block matrix $\mathbf{b}_j\mathbf{b}_j^T$ of $\mathbf{B}\mathbf{B}^T$ is 1; see the results of Theorem 3. This suggests using the best rank-1 approximation of each block $\mathbf{K}_j$ of $\mathbf{K}$ to obtain $\mathbf{B}$. Here we compute eigenvectors corresponding to the r largest eigenvalues of $\mathbf{K}$ and set them to be $\mathbf{B}$. In the next section, we show the computed matrices $\mathbf{K}$ and $\mathbf{B}$ for examples to demonstrate the performance of the proposed method. For convenience, we call the proposed method SN-ONMF.

4. Numerical examples. In this section, to test the performance of the proposed algorithm, we apply it on synthetic data and real-world data. For real-world data, we consider the text data sets, hyperspectral images, and natural scene data set. The results are compared to those generated by other methods. All experiments were implemented by MATLAB R2015a, run on an Intel Core 2 Quad CPU at 2.66 GHZ with 8 GB of RAM.

Algorithm 1. The Determination of $\mathbf{K}$ by Solving (12).

Input: Given $\mathbf{A} \in \mathbb{R}_+^{m\times n}$, the parameters θ, β, μ_1, μ_2, μ_3, initial values $\mathbf{K}^{(1)} \in \mathbb{R}_+^{m\times m}$, $\mathbf{X}^{(1)} \in \mathbb{R}_+^{m\times m}$, $\mathbf{Z}^{(1)} \in \mathbb{R}_+^{m\times m}$, $\mathbf{\Lambda}_1^{(1)} \in \mathbb{R}^{m\times m}$, $\mathbf{\Lambda}_2^{(1)} \in \mathbb{R}^{m\times m}$, $\mathbf{\Lambda}_3^{(1)} \in \mathbb{R}^{m\times m}$, and the stopping criterion ϵ.

Output: $\mathbf{K}$

for $i = 1, 2, \ldots$ **do**
  $\tilde{\mathbf{\Lambda}}_1^{(i)} = 2\mu_1(\mathbf{K}^{(i)} - \mathbf{K}^{(i)T}) + \mathbf{\Lambda}_1^{(i)} - \mathbf{\Lambda}_1^{(i)T}$
  $\mathbf{K}^{(i+1)} = (\mathbf{A}\mathbf{A}^T - \tilde{\mathbf{\Lambda}}_1^{(i)} + \mu_2\mathbf{X}^{(i)} + \mathbf{\Lambda}_2^{(i)} + \mu_3\mathbf{Z}^{(i)} + \mathbf{\Lambda}_3^{(i)})(\mathbf{A}\mathbf{A}^T + \mu_2\mathbf{I} + \mu_3\mathbf{I})^{-1}$
  $\mathbf{\Lambda}_1^{(i+1)} = \mathbf{\Lambda}_1^{(i)} + \mu_1(\mathbf{K}^{(i+1)} - \mathbf{K}^{(i+1)T})$
  $\mathbf{H}^{(i+1)} = \mathbf{K}^{(i+1)} - \frac{\mathbf{\Lambda}_2^{(i)}}{\mu_2}$
  $[\mathbf{U}^{(i+1)}, \mathbf{\Sigma}^{(i+1}, \mathbf{V}^{(i+1)}] = svd(\mathbf{H}^{(i+1)})$
  $\tau = \frac{\beta}{\mu_2}$
  $\mathbf{X}^{(i+1)} = \mathbf{U}_\tau^{(i+1)}\mathbf{\Sigma}_\tau^{(i+1)}\mathbf{V}_\tau^{(i+1)T}$
  $\mathbf{\Lambda}_2^{(i+1)} = \mathbf{\Lambda}_2^{(i)} + \mu_2(\mathbf{X}^{(i+1)} - \mathbf{K}^{(i+1)})$
  $\mathbf{W}^{(i+1)} = \mathbf{K}^{(i+1)} - \frac{\mathbf{\Lambda}_3^{(i)}}{\mu_3}$
  $\mathbf{Z}^{(i+1)} = \max\{sgn(\mathbf{W}^{(i+1)})(|\mathbf{W}^{(i+1)}| - \frac{\theta}{\mu_3}), 0\}$
  $\mathbf{\Lambda}_3^{(i+1)} = \mathbf{\Lambda}_3^{(i)} + \mu_3(\mathbf{Z}^{(i+1)} - \mathbf{K}^{(i+1)})$
  $nor1 = \|\mathbf{K}^{(i+1)} - \mathbf{K}^{(i)}\|_F$, $nor2 = \|\mathbf{X}^{(i+1)} - \mathbf{K}^{(i+1)}\|_F$, $nor3 = \|\mathbf{X}^{(i+1)} - \mathbf{K}^{(i+1)}\|_F$
  if $nor1$, $nor2$ and $nor3$ are less than ϵ **then**
    break
  end if
end for

4.1. Synthetic data. In this subsection, we consider three synthetic data examples to test the performance of the proposed algorithm. We also compare our results with those generated by UONMF and BiOR-NM3F [12]. The UONMF uses the multiplicative updates of $\mathbf{B}$ and $\mathbf{C}$ as follows:

$$(\mathbf{B})_{i,j} := (\mathbf{B})_{i,j}\sqrt{\frac{(\mathbf{A}\mathbf{C}^T)_{i,j}}{(\mathbf{B}\mathbf{B}^T\mathbf{A}\mathbf{C}^T)_{i,j}}} \quad \text{and} \quad (\mathbf{C})_{i,j} := (\mathbf{C})_{i,j}\frac{(\mathbf{A}^T\mathbf{B})_{i,j}}{(\mathbf{C}^T\mathbf{B}^T\mathbf{B})_{i,j}}.$$

Here $(\cdot)_{i,j}$ refers to the (i,j)th entry of a matrix. The BiOR-NM3F considers the following problem:

$$\min_{\mathbf{B},\mathbf{S},\mathbf{C}} \|\mathbf{A} - \mathbf{B}\mathbf{S}\mathbf{C}\|_F^2 \quad \text{subject to} \quad \mathbf{B}^T\mathbf{B} = I, \quad \mathbf{C}^T\mathbf{C} = I, \quad \mathbf{B} \geq 0, \quad \mathbf{S} \geq 0, \quad \mathbf{C} \geq 0.$$

The multiplicative updates of $\mathbf{B}$, $\mathbf{S}$ and $\mathbf{C}$ are given as follows:

$$(\mathbf{B})_{i,j} := (\mathbf{B})_{i,j}\sqrt{\frac{(\mathbf{A}\mathbf{C}^T\mathbf{S}^T)_{i,j}}{(\mathbf{B}\mathbf{B}^T\mathbf{A}\mathbf{C}^T\mathbf{S}^T)_{i,j}}}, \quad (\mathbf{S})_{i,j} := (\mathbf{S})_{i,j}\sqrt{\frac{(\mathbf{B}^T\mathbf{A}\mathbf{C}^T)_{i,j}}{(\mathbf{B}^T\mathbf{B}\mathbf{S}\mathbf{C}\mathbf{C}^T)_{i,j}}},$$

$$\mathbf{C}_{i,j} := \mathbf{C}_{i,j}\sqrt{\frac{(\mathbf{A}^T\mathbf{B}\mathbf{S})_{i,j}}{(\mathbf{C}^T\mathbf{C}\mathbf{A}^T\mathbf{B}\mathbf{S})_{i,j}}}.$$

To evaluate the performance of the algorithm, we define

$$error(\mathbf{K}_c) = \|\mathbf{K}_c - \mathbf{B}\mathbf{B}^T\|_F, \quad residual(\mathbf{K}_c) = \frac{\|\mathbf{A} - \mathbf{K}_c\mathbf{A}\|_F}{\|\mathbf{A}\|_F},$$

$$residual(\mathbf{B}_c) = \frac{\|\mathbf{A} - \mathbf{B}_c\mathbf{B}_c^T\mathbf{A}\|_F}{\|\mathbf{A}\|_F}, \quad \text{and} \quad orthogonal(\mathbf{B}_c) = \|\mathbf{B}_c^T\mathbf{B}_c - \mathbf{I}\|_F.$$

Here $\mathbf{K}_c$ and $\mathbf{B}_c$ refer to the computed matrices for $\mathbf{K}$ and $\mathbf{B}$ by the algorithm. For UONMF and BiOR-NM3F, the computed matrix $\mathbf{K}_c$ can be used by constructing $\mathbf{B}_c\mathbf{B}_c^T$. We see that the quantity "*error*" is used to measure how good the computed matrix $\mathbf{K}_c$ is, and the quantity "*residual*" is used to measure how good the fitting is. Also, the quantity "*orthogonal*" is used to measure the orthogonality of the computed matrix. The stopping criteria of UONMF and BiOR-NM3F are (i) that the difference of the iterates is small enough, i.e., $\|\mathbf{B}^{(i+1)} - \mathbf{B}^{(i)}\|_F < \epsilon$, or (ii) that the maximum number of iterations is set to be $imax$. The stopping criteria of Algorithm 1 are (i) that the quantities $nor1$, $nor2$, and $nor3$ are less than ϵ or (ii) that the maximum number of iterations is set to be $imax$. In the following synthetic data examples, we set $\epsilon = 10^{-6}$ and $imax = 10000$.

4.1.1. Example 1. We construct a 100-by-10 orthogonal nonnegative matrix $\mathbf{B}$. The structures of $\mathbf{B}$ and $\mathbf{B}\mathbf{B}^T$ are shown in Figure 1. Next we generate a 10-by-30 random nonnegative matrix $\mathbf{C}$ and obtain a 100-by-30 orthogonal decomposable matrix $\mathbf{A}$. By using the proposed algorithm SN-ONMF with $\theta = 10^{-4}$ and $\beta = 10^{-7}$, the computed matrix $\mathbf{K}_c$ and the calculated matrix $\mathbf{B}_c$ are shown in Figure 2. We see from Figures 1 and 2 that the proposed algorithm can recover both matrices quite well. We compare the results obtained by the other two algorithms and display them in Table 1. We see from the table that the quality of the solution computed by the proposed algorithm SN-ONMF is better than that computed by UONMF and BiOR-NM3F. The quantities $error(\mathbf{K}_c)$, $residual(\mathbf{K}_c)$, and $residual(\mathbf{B}_c)$ by the proposed method are smaller than those computed by UONMF and BiOR-NM3F. Also, the orthogonality of the computed matrices by UONMF and BiOR-NM3F is not well kept in the calculation.

4.1.2. Example 2. In this example, we use $\mathbf{A}$ generated in Example 1, and a noise is added to $\mathbf{A}$ such that $\mathbf{A}$ is not orthogonal decomposable. The added noise is based on MATLAB command $0.01 \times rand(100, 30)$. The magnitude of the noise

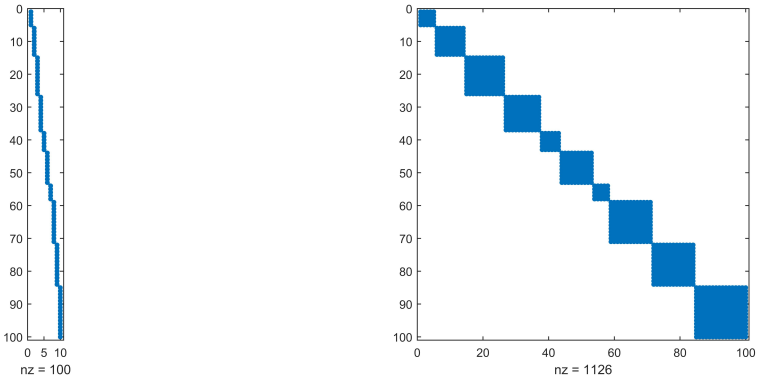

FIG. 1. *The structures of* $\mathbf{B}$ *(left) and* $\mathbf{B}\mathbf{B}^T$ *(right) in Example* 1.

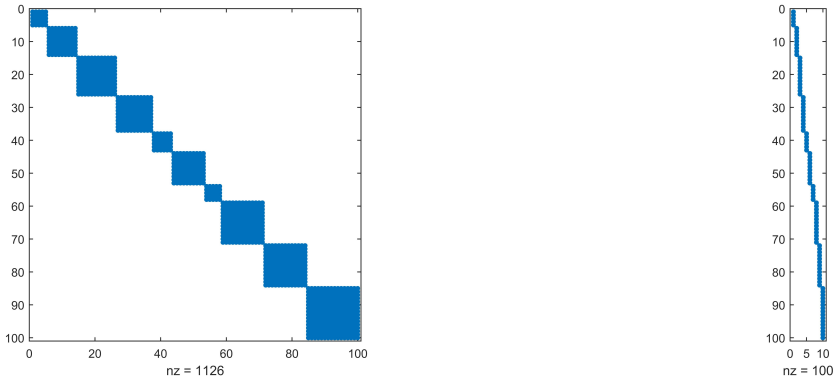

FIG. 2. *The computed solutions of* $\mathbf{K}_c$ *(left) and* $\mathbf{B}_c$ *(right) in Example* 1.

TABLE 1
The performance of different methods in Example 1.

	SN-ONMF	UONMF	BiOR-NM3F
$error(\mathbf{K}_c)$	1.1966×10^{-4}	0.0059	0.0029
$residual(\mathbf{K}_c)$	3.5336×10^{-5}	0.0018	8.5045×10^{-4}
$residual(\mathbf{B}_c)$	4.5285×10^{-10}	0.0018	8.5045×10^{-4}
$orthogonal(\mathbf{B}_c)$	4.4409×10^{-16}	0.0029	0.0017

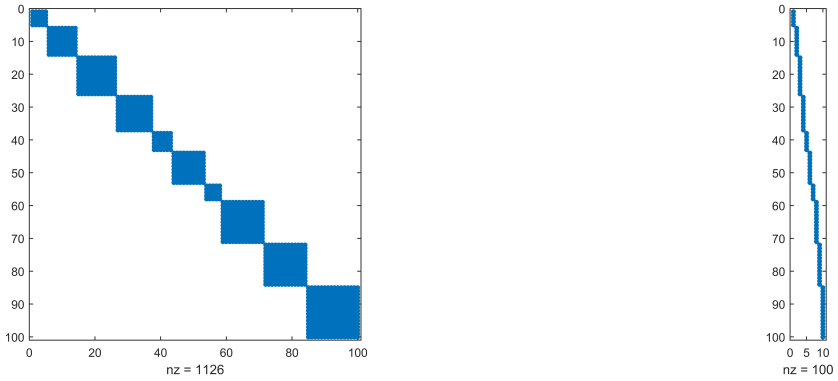

FIG. 3. *The computed solutions of* $\mathbf{K}_c$ *(left) and* $\mathbf{B}_c$ *(right) in Example* 2.

component is about 0.2752. The computed matrix $\mathbf{K}_c$ and the calculated matrix $\mathbf{B}_c$ by the proposed algorithm with the parameters used in Example 1 are shown in Figure 3. We see from Figures 1 and 3 that SN-ONMF can recover both matrices quite well. To evaluate the performance of the proposed algorithm, we show in Table 2 the quantities $error(\mathbf{K}_c)$, $residual(\mathbf{K}_c)$, $residual(\mathbf{B}_c)$, and $orthogonal(\mathbf{B}_c)$ by SN-ONMF, UONMF, and BiOR-NM3F. It is clear that the results by SN-ONMF are better than those by the other two comparison methods. Indeed, we see that these metrics for the solution given by UONMF are significantly affected by noise. These results demonstrate that the proposed algorithm SN-ONMF may be quite robust under the noise setting. Again the orthogonality of the computed matrices by UONMF

TABLE 2
The performance of different methods in Example 2.

	SN-ONMF	UONMF	BiOR-NM3F
$error(\mathbf{K})$	1.1796×10^{-4}	1.2493	0.0029
$residual(\mathbf{K})$	3.5580×10^{-5}	0.2851	8.5053×10^{-4}
$residual(\mathbf{B})$	9.9867×10^{-6}	0.2851	8.5053×10^{-4}
$orthogonal(\mathbf{B})$	3.3307×10^{-16}	0.6381	0.0017

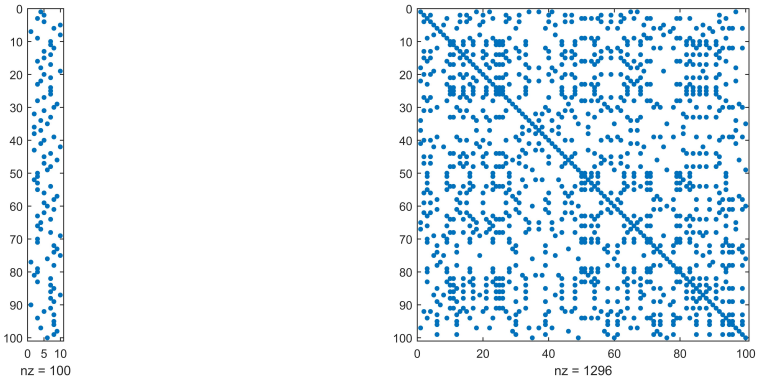

FIG. 4. *The structures of* $\mathbf{B}$ *(left) and* $\mathbf{BB}^T$ *(right) in Example* 3.

and BiOR-NM3F are not well kept in the calculation, while SN-ONMF can keep the orthogonality of the computed matrix very well.

4.1.3. Example 3. In this example, we generate a 100-by-10 orthogonal nonnegative matrix $\mathbf{B}$, and the nonzero entries of $\mathbf{B}$ are distributed randomly. The structures of $\mathbf{B}$ and $\mathbf{BB}^T$ are shown in Figure 4. Next we generate a 10-by-30 random nonnegative matrix $\mathbf{C}$ and obtain $\mathbf{A}$. Similar to Example 2, the added noise is based on MATLAB command $\alpha \times rand(100, 30)$. Here we choose $\alpha = 0.001, 0.01, 0.025, 0.05, 0.1$, and 0.5; the comparison results are shown in Table 3. From Table 3, it is clear that the evaluation metrics of SN-ONMF are better than those by UONMF and BiOR-NM3F. We also show the computed solutions $\mathbf{K}_c$ and $\mathbf{B}_c$ by the proposed method when $\alpha = 0.01$ in Figure 5.

4.2. Document data sets. In this subsection, we apply the proposed algorithm SN-ONMF to find clustering results for documents in terms data sets derived from the three data sources: TDT2, Reuters, and Newsgroup.

4.2.1. Data sets and evaluation measures. The document data sets are described in Table 4.

The TDT2 corpus consists of data collected during the first half of 1998 and taken from six sources, including two newswires (APW, NYT), two radio programs (VOA, PRI), and two television programs (CNN, ABC). It consists of 11201 topic documents which are classified into 96 semantic categories. In this paper, we adopt the data set tested in [5]. The documents appearing in two or more categories are the removed.

TABLE 3
The performance of different methods in Example 3.

		SN-ONMF	UONMF	BiOR-NM3F
$A + 0.001 \times rand(100, 30)$	$error(\mathbf{K})$	1.3990×10^{-4}	0.0063	0.0036
	$residual(\mathbf{K})$	4.2542×10^{-5}	0.0020	0.0012
	$residual(\mathbf{B})$	9.5232×10^{-7}	0.0020	0.0012
	$orthogonal(\mathbf{B})$	6.6613×10^{-16}	0.0032	0.0016
$A + 0.01 \times rand(100, 30)$	$error(\mathbf{K})$	1.2492×10^{-4}	0.0064	0.0036
	$residual(\mathbf{K})$	3.8706×10^{-5}	0.0021	0.0012
	$residual(\mathbf{B})$	1.0141×10^{-5}	0.0021	0.0012
	$orthogonal(\mathbf{B})$	4.4409×10^{-16}	0.0033	0.0016
$A + 0.025 \times rand(100, 30)$	$error(\mathbf{K})$	1.7032×10^{-4}	0.0062	0.0036
	$residual(\mathbf{K})$	2.7010×10^{-5}	0.0020	0.0012
	$residual(\mathbf{B})$	2.6443×10^{-5}	0.0020	0.0012
	$orthogonal(\mathbf{B})$	4.4409×10^{-16}	0.0030	0.0016
$A + 0.05 \times rand(100, 30)$	$error(\mathbf{K})$	1.2462×10^{-4}	0.0063	0.0036
	$residual(\mathbf{K})$	5.4503×10^{-5}	0.0020	0.0012
	$residual(\mathbf{B})$	5.3330×10^{-5}	0.0020	0.0012
	$orthogonal(\mathbf{B})$	4.4409×10^{-16}	0.0031	0.0016
$A + 0.1 \times rand(100, 30)$	$error(\mathbf{K})$	3.6772×10^{-4}	1.2133	0.0036
	$residual(\mathbf{K})$	1.1128×10^{-4}	0.2380	0.0012
	$residual(\mathbf{B})$	1.0881×10^{-4}	0.2380	0.0012
	$orthogonal(\mathbf{B})$	2.2204×10^{-16}	0.7378	0.0016
$A + 0.5 \times rand(100, 30)$	$error(\mathbf{K})$	6.9000×10^{-3}	1.1552	0.0037
	$residual(\mathbf{K})$	5.5627×10^{-4}	0.1848	0.0013
	$residual(\mathbf{B})$	5.4586×10^{-4}	0.1848	0.0013
	$orthogonal(\mathbf{B})$	4.4409×10^{-16}	0.6089	0.0016

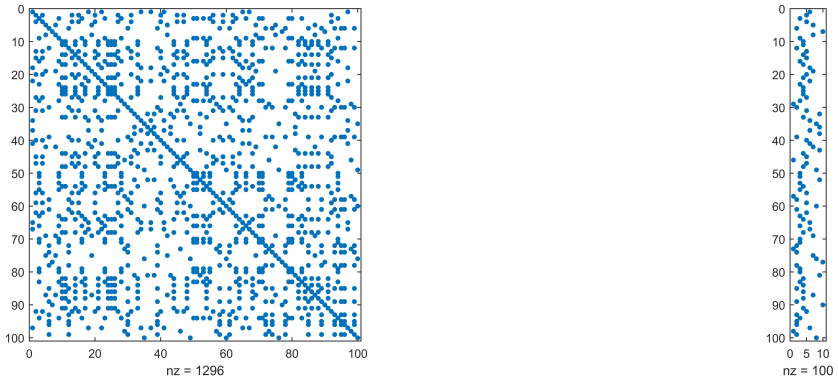

FIG. 5. *The computed solutions of* $\mathbf{K}_c$ *(left) and* $\mathbf{B}_c$ *(right) in Example* 3 *when* $\alpha = 0.01$.

TABLE 4
Summary of document data sets.

Data sets	Number of document	Data sets	Number of document
TDT2-Last10	652	TDT2-Last20	1937
TDT2-Top10	752	TDT2-Top20	884
Reuters-Top10	1001	Reuters-Top20	2424
Newsgroup-Top10	969	Newsgroup-Top10	1892

We use the documents coming from the 30 categories which are the largest collections in the data set and divide them into several subsets:

- TDT2-Last10: In this subset, we use all documents in the 10 categories with the smallest number of documents in the refined data set.
- TDT2-Last20: In this subset, we use all documents in the 20 categories with the smallest number of documents in the data set.
- TDT2-Top10: We consider the documents in the 10 categories with the largest number of documents in the data set and then construct the subset containing 10 percents of each category.
- TDT2-Top20: We consider the documents in the 20 categories with the largest number of documents in the data set and then construct the subset containing 10 percents of each category.

The Reuters-21578 data set is a collection of documents from Reuters newswire in 1987. It contains 21578 documents. They can be divided into 135 categories. In this data set, we use the following subsets for testing [5]:

- Reuters-Top10: From 10 categories with largest number of documents in Reuters data set, we collect 5 percents from the 1st category with the largest number documents and 10 percents from the 2nd category with the 2nd largest number documents, and from the 3rd to 20th categories, we choose 40 percents of each category.
- Reuters-Top20: Similarly, from 20 categories with largest number of documents in Reuters data set, we collect 5 percents from the 1st category with the largest number documents and 10 percents from the 2nd category with the 2nd largest number documents, and from the 3rd to 20th categories, we choose all documents. These documents construct the subset.

The Newsgroup contains 20000 newsgroup documents. It has 20 different newsgroups.

- Newsgroup-Top10: We built the subset in the top 10 categories with largest number of documents in Newsgroup data set. These documents are given by collecting 10 percents of each category.
- Newsgroup-Top20: The subset is constructed by collecting 10 percents documents from each category in top 20 categories with largest number of documents in Newsgroup data set.

To show the effectiveness of document clustering results, we use two evaluation measures, purity and entropy,

$$Purity = \sum_{i=1}^{r} \frac{max_j(n_i^j)}{n}, \quad Entropy = -\frac{1}{n \log m} \sum_{i=1}^{r} \sum_{j=1}^{r_0} n_i^j \log \frac{n_i^j}{n_i},$$

where n_i is the size of ith cluster, n_i^j is the number of documents of the jth input class that is assigned to the ith cluster, r is the number of clusters, r_0 is the number of original labels, and n is the total number of documents. Purity measures the dominance of the largest class per cluster. The larger the value of purity is, the better the clustering solution is. Entropy is a measure on uncertainty about the distribution of clustering results. The smaller the entropy value is, the better the clustering quality is.

In these real data sets, $\mathbf{A}$ may not be orthogonally decomposable. The recovery of $\mathbf{B}$ from eigenvectors of $\mathbf{K}$ directly may not be effective. Here the objective is to conduct document clustering, and we can make use of similarity values in $\mathbf{K}$ to achieve this purpose. We remark that the entry of $\mathbf{K}$ can be viewed as a link between two documents. When $(\mathbf{K})_{i,j}$ is large (small), it indicates that both ith and jth

TABLE 5
The clustering Results of different methods.

	SN-ONMF			BiOR-NM3F			k-means		
Data sets	Purity	Entropy	Time	Purity	Entropy	Time	Purity	Entropy	Time
TDT2-Last10	**0.9847**	**0.0293**	92.03	0.7209	0.3243	80.28	0.3972	0.6588	154.86
TDT2-Last20	**0.9504**	**0.0479**	2622.4	0.6460	0.3153	151.21	0.2967	0.7317	798.51
TDT2-Top10	**0.8059**	**0.2639**	130.91	0.7646	0.3180	743.29	0.5465	0.5649	303.71
TDT2-Top20	**0.7613**	**0.2304**	230.81	0.6697	0.3574	949.06	0.4932	0.5307	455.06
Reuters-Top10	**0.8002**	**0.2812**	986.39	0.5904	0.4983	288.76	0.3926	0.7522	288.82
Reuters-Top20	**0.7001**	**0.3305**	1154.62	0.5644	0.4695	1347.15	0.3614	0.6558	1365.98
Newsgroup-Top10	**0.4954**	**0.6297**	28.15	0.2322	0.8837	371.59	0.1197	0.9815	39.92
Newsgroup-Top20	**0.4117**	**0.6254**	236.11	0.1105	0.9519	352.16	0.0729	0.9761	123.83

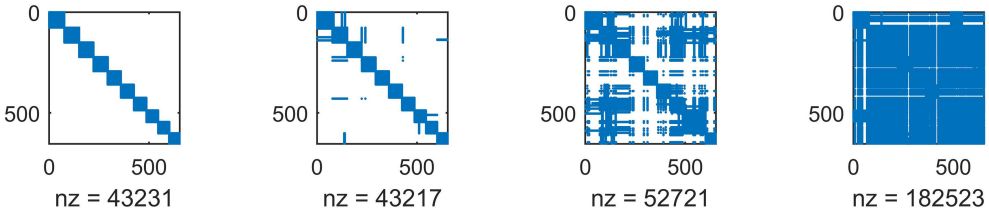

FIG. 6. *The clustering matrices for TDT2-Last* 10 *data set by different methods. From left to right: true clustering, SN-ONMF, BiOR-NM3F, k-means.*

documents are associated strongly (weakly) together. Therefore, we suggest using a spectral clustering algorithm[1] to obtain clustering of documents into r clusters. The entry $(\mathbf{B})_{i,k}$ is equal to one if the ith document belongs to the kth cluster; otherwise, the value is zero. In the following experiments, we compare the performance of the proposed algorithm SN-ONMF with BiOR-NM3F [12] and k-means algorithm [18, 28]. According to the results in [12], the performance of UONMF is not good. Therefore, we only compare the proposed method SN-ONMF with BiOR-NM3F for clustering the above data sets. The stopping criteria are similar to that of synthetic data set; here we set $\epsilon = 10^{-3}$ and $imax = 1000$.

4.2.2. Document clustering results. In this subsection, we compare the proposed algorithm SN-ONMF with BiOR-NM3F and k-means algorithms for the above data sets. We summarize the clustering results in Table 5. We see from the table that the proposed method SN-ONMF performs very well. The clustering results by the proposed method SN-ONMF have larger purity and smaller entropy than those by BiOR-NM3F and k-means. This comparison shows that the proposed method outperforms the other two methods on clustering. The computational time (in seconds) required by each method is also shown in Table 5. The computational results show that the proposed method SN-ONMF is also competitive in efficiency with the other two methods.

To see it better, for TDT2-Last10, we show the structure of clustering matrix that can be constructed from $\mathbf{BB}^T$, where $\mathbf{B}$ contains the ground truth classes of different documents. In the meantime, we display the computed cluster structure by different methods for comparison in Figure 6. Similarly, the cluster structure matrices for TDT-Last20 are shown in Figure 7. We can see from the figures that the cluster structure obtained by SN-ONMF looks better than those by the other two methods.

[1]It is equivalent to compute eigenvectors corresponding to the smallest r eigenvalues of the normalized Laplacian matrix of $\mathbf{K}$, and then k-means is used to determine r different clusters [27].

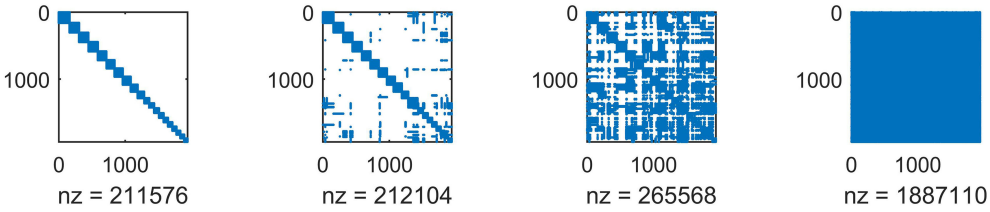

FIG. 7. *The clustering matrices for TDT2-Last 20 data set by different methods. From left to right: true clustering, SN-ONMF, BiOR-NM3F, k-means.*

4.3. Hyperspectral image unmixing. Hyperspectral imaging has wide applications in agriculture, mineralogy, physics, environment, and many other fields. It collects information from the object by taken at different wavelengths. The images are obtained by measuring the percentage of the light hitting a material, which is called reflectance. The aim of hyperspectral unmixing is to classify the pixels to different clusters; each one stands for a material. In this subsection, we use the Samson data set [39]. In the image, there are 952×952 pixels, and each pixel is recorded at 156 channels that cover the wavelengths from 401 to 889 nm. We use a region of 95×95 pixels starting from $(252, 332)$ pixel in the original image. In the region, there are three different materials, "Tree," "Rock," and "Water." We apply SN-ONMF, UONMF, BiOR-NM3F, and k-means on the image, respectively. Here, we compute three eigenvectors corresponding to three largest eigenvalues of $\mathbf{K}$ and then employ three eigenvectors for three clusters evaluation. The stopping criteria are $\epsilon = 10^{-3}$ and $imax = 1000$. The ground truth and the computational results are displayed in Figure 8.

According to the computational results, SN-ONMF shows a good clustering performance. It is clear that "Tree," "Rock," and "Water" can be extracted suitably. For the other two methods, they do not perform well, as they are not able to classify three materials separately. To evaluate the clustering results quantitatively, the following accuracy metric is employed:

$$Accuracy = \frac{1}{r} \sum_{i=1}^{r} \frac{< \mathbf{b}_i, \mathbf{g}_i >}{\|\mathbf{b}_i\|_2 \|\mathbf{g}_i\|_2},$$

where $\{\mathbf{b}_i\}_{i=1}^r$ are the extracted features and $\{\mathbf{g}_i\}_{i=1}^r$ are the ground truth features. It describes the similarity of the ground truth feature space and the computed feature space. The larger the value is, the better the results we obtain. The clustering accuracy and computational time (in seconds) are listed in Table 6. We see from the table that SN-ONMF requires more computational time, but its accuracy is higher than those of the other methods.

4.4. Multilabel natural scene clustering. In natural scene clustering, each scene image may belong to several image classes simultaneously. We apply our method on scene classification in this subsection. The experimental data set is studied in the literature [3] and can be downloaded from [29]. It consists of 2407 natural scene images with a set of label assigned to each image. Table 7 gives the detailed descriptions. In this data set, each color image is converted to the CIE Luv space and then divided into 49 blocks using a 7×7 grid. The first and second moments (mean and variance) of each band are computed, corresponding to a low-resolution image and to computationally inexpensive texture features, respectively. Each image is finally transformed to a $49 \times 2 \times 3$-dimensional (i.e., 294-dimensional) feature vector.

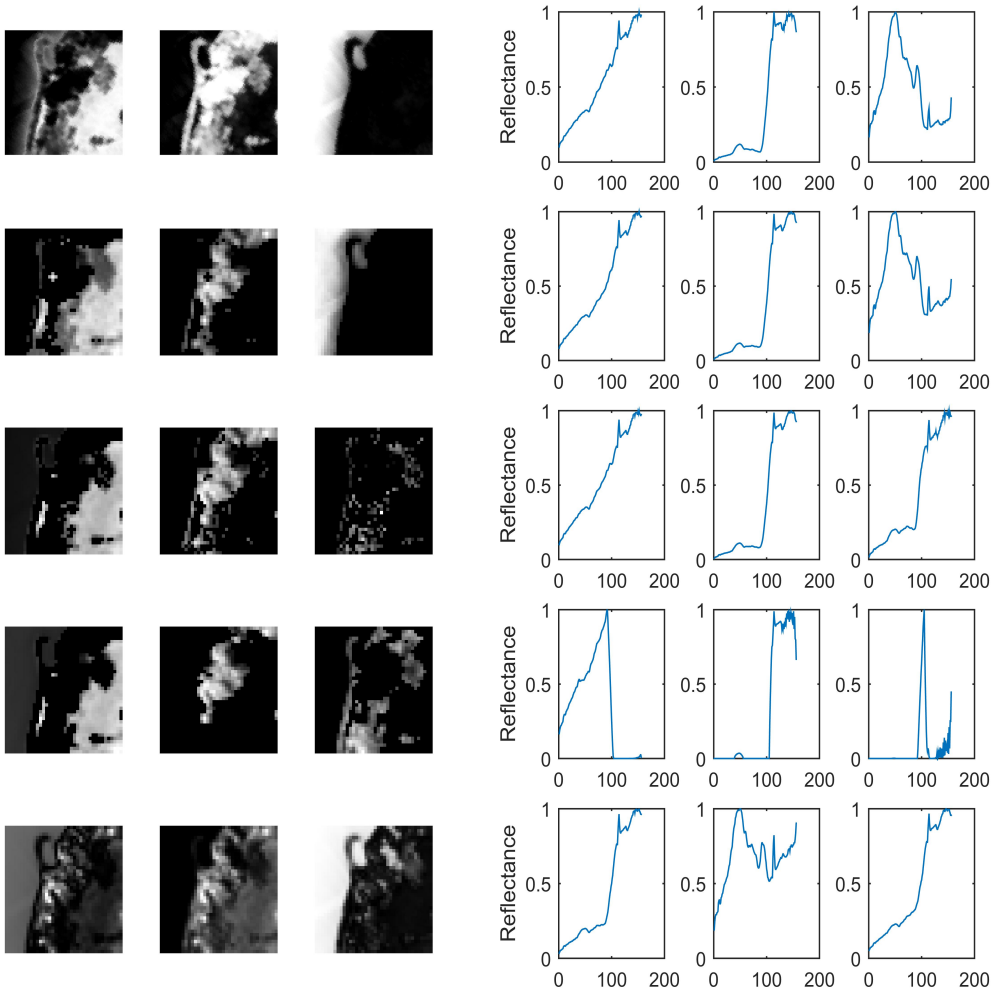

FIG. 8. *Left: Rock, Tree, Water; Right: reflectance of Rock, Tree, Water. From the top to bottom: ground truth, SN-ONMF, UONMF, BiOR-NM3F, k-means.*

TABLE 6
The hyperspectral unmixing results of different methods.

	SN-ONMF	UONMF	BiOR-NM3F	k-means
Accuracy	0.7631	0.4834	0.4607	0.7018
Computational time	898.30	68.68	71.17	0.06

In the data set, there are 2407 images with 6 different class labels, i.e., Beach, Fall Foliage, Field, Mountain, Sunset, and Urban. We employ SN-ONMF, UONMF, BiOR-NM3F, and soft k-means (fuzzy c-means) [2] on scene data set, respectively. The stopping criteria here are $\epsilon = 10^{-3}$ and $imax = 1000$. In the proposed method SN-ONMF, we obtain 2047-by-6 class-indicator matrix $\mathbf{P}$ from the computed matrix $\mathbf{K}$ by using soft k-means algorithm. In the soft clustering procedure, each image can be classified into six possible classes where $(\mathbf{P})_{i,j}$ refers to the degree of association of the ith image to the jth class.

TABLE 7
Summary of the natural scene data set.

Number of labels	Label set	Number of images
1	Beach	369
	Fall Foliage	360
	Field	327
	Mountain	405
	Sunset	364
	Urban	405
2	Beach+Field	1
	Beach+Mountain	38
	Beach+Urban	19
	Fall Foliage+Field	23
	Fall Foliage+Mountain	13
	Field+Mountain	75
	Field+Urban	6
	Mountain+Urban	1
3	Field+Fall Foliage+Mountain	1

TABLE 8
The scene clustering results of different methods.

	SN-ONMF	UONMF	BiOR-NM3F	Soft k-means
Precision	0.5508	0.4525	0.4309	0.4340
Computational time	1211.32	136.61	237.43	0.18

To evaluate the performance of different methods, we compare the ground truth classes and the assigned classes. For example, when the number of ground truth classes is ℓ, we choose the largest ℓ associations of classes from $\mathbf{P}$ as the assigned classes by the method. The precision can be calculated based on the matching between ground truth classes and the assigned classes. In Table 8, we show the average precision of different methods. We see from the table that the precision of SN-ONMF is better than that of the other testing methods. The computational time required by SN-ONMF is larger than those by the other testing methods.

5. Concluding remarks. In the paper, we study ONMF, which arises in many data analysis and applications. The main contribution of this paper is to show that the coefficient matrix can be sparse and low-rank in the ONMF. Then we propose to use a sparsity and nuclear norm minimization for the factorization and develop a convex optimization model for finding the coefficient matrix in the factorization. To illustrate the effectiveness of the proposed algorithm, numerical examples including synthetic and real-world data sets are presented, and they demonstrate that the clustering performance of the proposed method SN-ONMF is better than that of the other testing methods in the paper.

REFERENCES

[1] M. W. BERRY AND J. KOGAN, *Text Mining: Applications and Theory*, John Wiley and Sons, West Sussex, UK, 2010.
[2] J. C. BEZDEK, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Springer Science and Business Media, New York, 2013.
[3] M. R. BOUTELL, J. B. LUO, X. P. SHEN, AND C. M. BROWN, *Learning multi-label scene classification*, Pattern Recogn., 37 (2004), pp. 1757–1771.
[4] D. CAI, X. HE, J. HAN, AND T. S. HUANG, *Graph regularized nonnegative matrix factorization for data representation*, IEEE Trans. Pattern Anal. Mach. Intell., 33 (2011), pp. 1548–1560.

[5] D. Cai, Q. Mei, J. Han, and C. Zhai, *Modeling hidden topics on document manifold*, in Proceedings of the ACM Conference on Information and Knowledge Management, 2008, pp. 911–920.
[6] J. F. Cai, E. J. Candés, and Z. Shen, *A singular value thresholding algorithm for matrix completion*, SIAM J. Optim., 20 (2010), pp. 1956–1982 .
[7] M. Chen, W. S. Chen, B. Chen, and B. Pan, *Non-negative sparse representation based on block NMF for face recognition*, in Biometric Recognition, Springer International Publishing, New York, 2013, pp. 26–33.
[8] H. Cho, I. S. Dhillon, Y. Guan, and S. Sra, *Minimum sum squared residue based co-clustering of gene expression data*, in Proceedings of the 4th SIAM International Conference on Data Mining (SDM), 2004, pp. 114–125.
[9] S. Choi, *Algorithms for orthogonal nonnegative matrix factorization*, in Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE, New York, 2008, pp. 1828–1832.
[10] A. Cichocki, R. Zdunek, A. H. Phan, and S. I. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*, John Wiley and Sons, New York, 2009.
[11] C. Ding, X. He, and H. D. Simon, *On the equivalence of nonnegative matrix factorization and spectral clustering*, in Proceedings of the SIAM International Conference on Data Mining (SDM'05), 2005, pp. 606–610.
[12] C. Ding, T. Li, W. Peng, and H. Park, *Orthogonal nonnegative matrix tri-factorizations for clustering*, in Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD06), ACM Press, New York, 2006, pp. 126–135.
[13] D. Gabay and B. Mercier, *A dual algorithm for the solution of nonlinear variational problems via finite-element approximations*, Comput. Math. Appl., 2 (1976), pp. 17–40.
[14] Y. Gao and G. Church, *Improving molecular cancer class discovery through sparse non-negative matrix factorization*, Bioinformatics, 21 (2005), pp. 3970–3975.
[15] R. Glowinski and A. Marrocco, *Sur l' approximation par áláments finis d'ordre un, et la résolution par pénalisation-dualité d'une classe de problèmes de Dirichlet non lineéaires*, RAIRO Anal. Number, 9 (1975), pp. 41–76.
[16] D. Guillamet and J. Vitria, *Non-negative matrix factorization for face recognition*, in Topics in Artificial Intelligence, Springer, Berlin, 2002, pp. 336–344.
[17] D. Guillamet, J. Vitria, and B. Schiele, *Introducing a weighted nonnegative matrix factorization for image classification*, Pattern Recogn. Lett., 24 (2003), pp. 2447–2454.
[18] J. A. Hartigan, *Clustering Algorithms*, John Wiley and Sons, New York, 1975.
[19] L. P. Jing, C. Zhang, and M. K. Ng, *SNMFCA: Supervised NMF-based image classification and annotation*, IEEE Trans. Image Process., 21 (2012), pp. 4508–4521.
[20] P. M. Kim and B. Tidor, *Subsystem identification through dimensionality reduction of large-scale gene expression data*, Genome Res., 13 (2003), pp. 1706–1718.
[21] H. Kim and H. Park, *Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis*, Bioinformatics, 23 (2007), pp. 1495–1502.
[22] D. D. Lee and H. S. Seung, *Learning of the parts of objects by non-negative matrix factorization*, Nature, 401 (1999), pp. 788–791.
[23] D. D. Lee and H. S. Seung, *Algorithms for Nonnegative Matrix Factorization*, Vol. 13, MIT Press, Cambridge, MA, 2001.
[24] T. Li and C. Ding, *The relationships among various nonnegative matrix factorization methods for clustering*, in Proceedings of the 6th International Conference on Data Mining (ICDM06), IEEE Computer Society, Washington, DC, 2006, pp. 362–371.
[25] H. Liu, Z. Wu, D. Cai, and T. S. Huang, *Constrained nonnegative matrix factorization for image representation*, IEEE Trans. Pattern Anal. Mach. Intell., 34 (2012), pp. 1299–1311.
[26] Y. Liu, L. P. Jing, and M. K. Ng, *Robust and non-negative collective matrix factorization for text-to-image transfer learning*, IEEE Trans. Image Process., 24 (2015), pp. 4701–4714.
[27] U. V. Luxburg, *A tutorial on spectral clustering*, Stat. Comput., 17 (2007), pp. 395–416.
[28] J. B. MacQueen, *Some methods for classification and analysis of multivariate observations*, in Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability, University of California Press, Berkeley, 1967, pp. 281–297.
[29] mulan.sourceforge.net/datasets-mlc.html.
[30] P. Paatero and U. Tapper, *Positive matrix factorization: A nonnegative factor model with optimal utilization of error estimates of data values*, Environmetrics, 5 (1994), pp. 111–126.

[31] A. Pascual-Montano, P. Carmona-Saez, M. Chagoyen, F. Tirado, J. M. Carazo, and R. Pacual-Marqui, *bioNMF: A versatile tool for non-negative matrix factorization in biology*, BMC Bioinformatics, 7 (2006).

[32] V. P. Pauca, F. Shahnaz, M. W. Berry, and R. J. Plemmons, *Text mining using non-negative matrix factorizations*, in Proceedings of the SIAM International Conference on Data Mining, Vol. 4, 2004, pp. 452–456.

[33] F. Shahnaz, M. Berry, P. Pauca, and R. Plemmons, *Document clustering using non-negative matrix factorization*, J. Info. Process. Manag., 42 (2006), pp. 373–386.

[34] Y. Wang, Y. Jia, C. Hu, and M. Turk, *Non-negative matrix factorization framework for face recognition*, Int. J. Pattern Recognit. Artif. Intell., 19 (2005), pp. 495–511.

[35] G. Wang, A. V. Kossenkov, and M. F. Ochs, *LS-NMF: A modified non-negative matrix factorization algorithm utilizing uncertainty estimates*, BMC Bioinformatics, 7 (1975), 2006.

[36] W. Xu, X. Liu, and Y. Gong, *Document clustering based on non-negative matrix factorization*, in Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 03), ACM Press, New York, 2003, pp. 267–273.

[37] Z. Yuan and E. Oja, *Projective Nonnegative Matrix Factorization for Image Compression and Feature Extraction*, in Scandinavian Conference on Image Analysis, Springer, Berlin, 2005, pp. 333–342.

[38] D. Zhang, S. Chen, and Z.-H. Zhou, *Two-dimensional non-negative matrix factorization for face representation and recognition*, Lect. Notes Comput. Sci., 3723 (2005), pp. 350–363.

[39] F. Zhu, Y. Wang, B. Fan, S. Xiang, G. Meng, and C. Pan, *Spectral unmixing via data-guided sparsity*, IEEE Trans. Image Process., 23 (2014), pp. 5412–5427.