Journal of Computational Mathematics
Vol.36, No.6, 2018, 761–775.

http://www.global-sci.org/jcm
doi:10.4208/jcm.1704-m2016-0657

NONNEGATIVE MATRIX FACTORIZATION WITH BAND CONSTRAINT*

Xiangxiang Zhu, Jicheng Li and Zhuosheng Zhang[1)]
School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China
Email: zhuxiangxiang@stu.xjtu.edu.cn, jcli@mail.xjtu.edu.cn, zszhang@mail.xjtu.edu.cn

Abstract

In this paper, we study a band constrained nonnegative matrix factorization (band NMF) problem: for a given nonnegative matrix Y, decompose it as $Y \approx AX$ with A a nonnegative matrix and X a nonnegative block band matrix. This factorization model extends a single low rank subspace model to a mixture of several overlapping low rank subspaces, which not only can provide sparse representation, but also can capture significant grouping structure from a dataset. Based on overlapping subspace clustering and the capture of the level of overlap between neighbouring subspaces, two simple and practical algorithms are presented to solve the band NMF problem. Numerical experiments on both synthetic data and real images data show that band NMF enhances the performance of NMF in data representation and processing.

Mathematics subject classification: 15A23, 65F30, 90C59.
Key words: Nonnegative matrix factorization, Band structure, Subspace clustering, Sparse representation, Image compression.

1. Introduction

In many large datasets, the relevant information often lies in a low dimensional subspace of the ambient space, which leads to a large interest in representing data with low rank approximations. Particularly, the nonnegative matrix factorization (NMF) for nonnegative data analysis, not only uncovers latent low dimensional structures intrinsic in high dimensional data but also provides a nonnegative, part-based representation of data. With these strengths, NMF has attracted intensive studies in the last decades and various factorization models, algorithms and regularized variants have been developed for different purposes and applications [1,2,5,9,20].

Despite the growing availability of tools for NMF, many techniques ignore an underlying information that the data often contains some type of structure that enables intelligent representation and processing. In computer vision, for example, a collection of images of an object taken under different illuminations has not only a low rank representation [4], but also significant spatial structure relating to the statistics of the scene, such as sparseness on a particular wavelet basis or low total variation [3]. Also, for monaural blind source separation [8], the coefficient matrix X with block diagonal structure indicates where each source is active when there is no training data for the individual sources.

In recent decade, more and more strategies have been given to represent and capture the intrinsic structure implied in the data. Kim et al. [13] introduced a novel formulation of sparse NMFs to get the sparse structure. Cai et al. [12] proposed a graph regularized nonnegative

* Received June 5, 2016 / Revised version received December 8, 2016 / Accepted April 28, 2017 / Published online August 7, 2018 /
1) Corresponding author

matrix factorization (GNMF) model to discover the intrinsic geometrical and discriminating structure of the data space by constructing a nearest neighbor graph. Pei et al. [14] incorporated neighbor isometric regularized constraint in the optimization of the NMF to extract the low rank space that preserves neighbor isometric geometry structure. Similar to these methods, Wu et al. [15] proposed a nonnegative low rank and group sparse matrix factorization (NLRGS) method to capture the grouping structure by simultaneously integrating low rank and group sparse constraints. All these methods, however, are difficult to clearly identify the implicit data structural characteristics for two reasons. First, all these methods are essentially based on the hypothesis that all data is approximately drawn from a low rank subspace. However, a given dataset can seldom be well described by a single subspace. A more reasonable model is to consider samples as a mixture of several overlapping low rank subspaces, as shown in Fig. 1.1 (left). The right is a real image whose pixels is continuously changing, so its pixels are well characterized by its neighbor points, which favors our proposed overlapping subspace model. Second, there are limitations of using regularization method to capture the data structural characteristics. Indeed, when data are from a union of five overlapping subspaces, almost all of the methods which extend the NMF problem formulation to include additional regularization terms on A and/or X cannot capture this overlapping low rank subspace structure (see Fig. 4.3(a-c)).

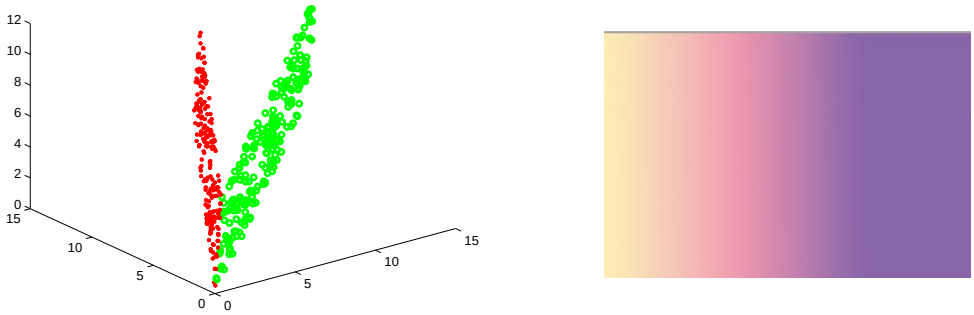


Fig. 1.1. A display of overlapping subspace model.

Besides, low rank representation (LRR) [10] and its various modified methods [16,17] incorporated the low rank constraint to represent each sample as a linear combination of other samples. However, these low rank methods neither derive the projection subspace of original examples nor get the block band structure of the coefficient matrix when data are overlapping. More specifically, LRR and its variants can only obtain the block diagonal structure of the coefficient matrix approximatively although some subspaces share a few bases (see Fig. 4.3(d)).

To overcome the aforementioned deficiencies, this paper proposes a band constrained nonnegative matrix factorization model (named band NMF). Particularly, band NMF extends a single low rank subspace model to a mixture of several overlapping low rank subspaces, which not only can provide sparse representation but also can capture significant grouping structure from a dataset. Additionally, the coefficient matrix X with band structure is also considered as a filtering matrix that performs continuity and removes slowly changing trends from the data representation. Moreover, the block band matrices allow for convenient storage, which is very

significant to represent data in the practical application. Benefiting from these properties, experimental results substantiate that band NMF has a better performance in data representation and processing than the state-of-the-art NMF models.

The remainder of this paper is organized as follows. In Section 2, we give a brief description of related conceptions and results. In Section 3, we state the band constrained nonnegative matrix factorization problem and two algorithms are proposed to solve this problem approximately. The experimental results are highlighted in Section 4. Finally, the conclusions are given in Section 5.

The following notation is used. $R_+^{m\times n}$ denotes the set of nonnegative real matrices with order m by n. $[X]_{ij}$ or x_{ij} denotes the (i,j) th element of the matrix X, X_{ij} denotes the (i,j) th block of the block matrix $X=(X_{ij})_{m\times n}$, and $X(:,j)$ $(X(j,:))$ the j th column (row) of the matrix X. X^T denotes the transpose of X, and $X\geq 0$ means the elements of X are nonnegative. span(U) denotes the linear space spanned by the columns of the matrix U.

2. Related Conceptions and Known Results

In this section, we first briefly review an important special matrix–block band matrix and its some properties. Next, the nonnegative matrix factorization and related algorithms are presented. Finally, we introduce the subspace clustering by low rank representation.

2.1. Block band matrix

Definition 2.1. *The block matrix $X=(X_{ij})_{m\times n}$, $X_{ij}\in R^{m_i\times n_j}$, is called a $\{c_1,c_2\}$ block band matrix if, for nonnegative integers c_1 and c_2, its blocks satisfy*

$$X_{ij}=0 \quad when \quad i>j+c_1 \ or \ j>i+c_2.$$

The reasonably small nonnegative integers c_1 and c_2 are called the lower and upper bandwidth, respectively. There are many special block band matrices that occur frequently, such as $m\times n$ block upper triangular matrices ($\{0,n-1\}$ block band matrices), $m\times n$ block tridiagonal matrices ($\{1,1\}$ block band matrices) and $m\times n$ block lower Hessenberg matrices ($\{m-1,1\}$ block band matrices).

Since the block band matrices are special sparse matrices, many efficient methods for storing sparse matrices, such as hierarchical storage format [24] and arithmetical-coding-based format [25], can be used for the storage of the block band matrices. Particularly, for a $\{c_1,c_2\}$ block band matrix $X=(X_{ij})_{m\times n}$, when $X_{ij}\in R^{m_r\times m_j}$ (that is, each block X_{ij} has the same number of rows), then such a matrix can be stored in a $(c_1+c_2+1)\times n$ block matrix BX with the rule that

$$BX(i-j+c_2+1,j)=X_{ij},$$

for all (i,j) that fall inside the band, and the rest blocks of the matrix BX are implicitly zero. For example, a $\{1,2\}$ block band matrix with 6×6 blocks is stored as a 4×6 block matrix as below.

$$\begin{bmatrix} X_{11} & X_{12} & X_{13} & 0 & 0 & 0 \\ X_{21} & X_{22} & X_{23} & X_{24} & 0 & 0 \\ 0 & X_{32} & X_{33} & X_{34} & X_{35} & 0 \\ 0 & 0 & X_{43} & X_{44} & X_{45} & X_{46} \\ 0 & 0 & 0 & X_{54} & X_{55} & X_{56} \\ 0 & 0 & 0 & 0 & X_{65} & X_{66} \end{bmatrix} \Longrightarrow \begin{bmatrix} 0 & 0 & X_{13} & X_{24} & X_{35} & X_{46} \\ 0 & X_{12} & X_{23} & X_{34} & X_{45} & X_{56} \\ X_{11} & X_{22} & X_{33} & X_{44} & X_{55} & X_{66} \\ X_{21} & X_{32} & X_{43} & X_{54} & X_{65} & 0 \end{bmatrix}.$$

From a computational point of view, working with block band matrices is always preferential to working with similarly dimensional dense matrices. For example, suppose $X \in R^{n\times n}$ is a $\{c_1, c_2\}$ band matrix (that is, each block of this matrix is 1×1 submatrix), c_1, c_2 are much smaller than n and $y, z \in R^n$, the arithmetic operation count for $y + Xz$ is just $2n(c_1 + c_2 + 1)$ flops, while the count is $2n^2$ flops if the matrix X is a dense matrix. Thus the work involved in performing operations such as multiplication falls significantly, often leading to huge savings in terms of computing time and complexity.

2.2. Nonnegative matrix factorization

The NMF problem can be stated as follows: for a given nonnegative matrix $Y \in R_+^{m\times n}$ and a positive integer $k \ll \min(m, n)$, we aim to find two nonnegative matrices $A \in R_+^{m\times k}$ and $X \in R_+^{k\times n}$ to minimize the squared Frobenius norm

$$D_F(Y \parallel AX) = \frac{1}{2} \parallel Y - AX \parallel_F^2. \tag{2.1}$$

Generally, A is called a basis matrix, X is called a coefficient matrix.

In NMF, it is often assumed that the factorization rank k is given. In practice, however, the given value k is suboptimal and therefore we must choose a suitable value depending on the data and setting. Two popular approaches are: trial and error (that is, test different values of k and pick the one performing best for your model), estimation using the SVD (that is, look at the decay of the singular values of the input data matrix). In this paper, k is automatically chosen according to the dimension of the subspace and level of overlap (that is, the number of the common bases between two subspaces), as will be shown in Section 3.

Lee and Sueng [6] proposed classical multiplicative update algorithm (MU) for NMF as follows:

$$a_{ij} \leftarrow a_{ij} \frac{[YX^T]_{ij}}{[AXX^T]_{ij} + \varepsilon}, \tag{2.2}$$

$$x_{jt} \leftarrow x_{jt} \frac{[A^TY]_{jt}}{[AA^TX]_{jt} + \varepsilon}. \tag{2.3}$$

The constant ε, usually taken as 10^{-9}, is added to avoid division by zero, and the following theorem ensures its convergence.

Theorem 2.1. ([6]) *The objective function* (2.1) *is nonincreasing under the update rules* (2.2) *and* (2.3).

In contrast, alternating nonnegative least squares (ANLS) is a class of methods where the subproblems $\min\limits_{A\geq 0} \parallel Y^T - X^TA^T \parallel_F^2$ and $\min\limits_{X\geq 0} \parallel Y - AX \parallel_F^2$ are solved exactly, that is, the

update for A is given by

$$A = \arg\min_{A \geq 0} \| Y^T - X^T A^T \|_F^2, \tag{2.4}$$

where X is fixed, and

$$X = \arg\min_{X \geq 0} \| Y - AX \|_F^2, \tag{2.5}$$

where A is fixed. Many methods can be used to solve the two subproblems (2.4) and (2.5), and dedicated active-set methods [22] have shown to perform very well in practice. Importantly, ANLS has the convergence property that every limit point is a stationary point [22].

2.3. Subspace clustering by low rank representation

Low rank representation [10] learns the structural representation Z over the specific dictionary Y via the low rank constraint. Thus, the final objective function is

$$\min_{Z,E} \| Z \|_* + \lambda \| E \|_{2,1}, \quad st \;\; Y = YZ + E, \tag{2.6}$$

where $\lambda > 0$ is a predefined constant and $\| Z \|_*$ indicates the nuclear norm of the matrix Z, and $\| E \|_{2,1} = \sum_i \| E(:,i) \|_2$.

Let Z^* be the minimizer of the problem (2.6), and the skinny SVD of Z^* is $U^*\Sigma^*(V^*)^T$, an affinity matrix W is constructed as follows:

$$[W]_{ij} = [\tilde{U}(\tilde{U})^T]_{ij}^{\,2}, \quad i,j = 1,2,\cdots,n, \tag{2.7}$$

where $\tilde{U}$ is formed by $U^*(\Sigma^*)^{\frac{1}{2}}$ with normalized rows. Finally, the spectral clustering algorithms [18] are used to segment the data samples into l clusters. Algorithm 2.1 summarizes the whole procedure of performing cluster.

Algorithm 2.1. Subspace Clustering.

1. Input data matrix Y, number l of clusters;
2. Obtain the minimizer Z^* of problem (2.6);
3. Compute the skinny SVD $Z^* = U^*\Sigma^*(V^*)^T$;
4. Construct an affinity matrix W by (2.7);
5. Use W to perform Normalized Cuts [19] and segment the data samples into l clusters.

Instead of using LRR, different subspace clustering methods are proposed based on various schemes. Fischler et al. [7] proposed a robust statistical approach, named random sample consensus (RANSAC), which fits a subspace of dimension d to randomly chosen subsets of d points until the number of inliers is large enough. In [23], Yan et al. presented a local spectral clustering-based approaches (LSA) by using local information around each point to build a similarity between pairs of points. In addition, Elhamifar et al. [11] proposed a sparse subspace clustering method (SSC) by solving a sparse optimization program whose solution is used in a spectral clustering framework to infer the clustering of the data into subspaces. About other subspace clustering algorithms, one can see paper [11,26] and the references therein.

3. Nonnegative Matrix Factorization with Band Constraint

The band constrained nonnegative matrix factorization (band NMF) studied in this paper can be stated as

Problem 3.1. *For a given nonnegative matrix $Y \in R_+^{m\times n}$, find a positive integer $k \ll \min(m,n)$, a nonnegative matrix $A \in R_+^{m\times k}$, and a nonnegative block band matrix $X \in R_+^{k\times n}$ to minimize the squared Frobenius norm*

$$D_F(Y \parallel AX) = \frac{1}{2}\parallel Y - AX \parallel_F^2.$$

To solve Problem 3.1, we mainly consider how to determine the factorization rank k and the optimal block band structure of the coefficient matrix X. To do this, we study the following two contents. First, since our band NMF model extends a single low rank subspace model to a mixture of several overlapping low rank subspaces, we need to segment all data points (i.e., all of the columns of the data matrix Y) into their respective subspaces–overlapping subspace clustering. Second, we need to capture the level of overlap between neighboring subspaces. Based on these two contents, two algorithms are presented for band NMF.

(1) Overlapping subspace clustering

Completing overlapping subspace clustering for all columns of Y, we first use the soft thresholding approach [10] to estimate the number l of subspaces.

In the clustering process, one of the drawbacks of the Algorithm 2.1 is that the clustering performance is not accurate enough when some subspaces share a few base vectors. To get more accurate results, we propose the Algorithm 3.1 based on the idea of reassignment. That is, after the initial l clusters $S_1, S_2, \cdots, S_l$ have been completed using Algorithm 2.1 (or other clustering algorithms like SSC [26]), we repeat the following two steps until convergence: 1) compute the measure matrix Q by (3.1) to capture closeness between data points and clusters. 2) reassign data points into the closest cluster.

Here, the measure matrix Q is defined as follows:

$$[Q]_{ij} = \min_x \parallel y_i - A_j x \parallel_2, \tag{3.1}$$

where y_i indicates the i th columns of the matrix Y, and A_j is a $m \times d_j$ matrix whose columns are randomly selected from the j th cluster S_j, $i = 1, 2, \cdots, n, j = 1, 2, \cdots, l$. Note that the value of d_j is obtained according to the priori information. For example, all of $d_j (j = 1, 2, \cdots, l)$ are set as 9 when processing face images because face images under different illuminations can be well-approximated by a 9 dimensional linear subspace.

Algorithm 3.1. Overlapping Subspace Clustering.

1. Input data matrix Y, the number l of clusters and the subspace dimensions $d_j, j = 1, 2, \cdots, l$;
2. Class initialization using Algorithm 2.1 (or SSC) to obtain S_j;
3. For each $\{y_i\}_{i=1}^n$, compute the measure matrix Q by (3.1);
4. For each $\{y_i\}_{i=1}^n$, reassign y_i to the j th cluster S_j if the value $[Q]_{ij}$ is the smallest;
5. Repeat 3,4 until convergence.

The following definitions are presented to build neighboring relations between subspaces and capture their levels of overlap, which determine the factorization rank k and the block band structure of the coefficient matrix X.

Definition 3.1. *Suppose U_j and U_h are two nonnegative matrices, the nonnegative principal angles $\theta_{j,h}^{(1)}, \cdots, \theta_{j,h}^{(\min(d_j,d_h))}$ between two subspaces* $\mathrm{span}(U_j)$ *and* $\mathrm{span}(U_h)$ *of dimensions d_j and d_h are recursively defined by*

$$\cos(\theta_{j,h}^{(i)}) = \max_{y\in\mathrm{span}(U_j),y\geq 0} \max_{z\in\mathrm{span}(U_h),z\geq 0} \frac{y^T z}{\parallel y \parallel_2 \parallel z \parallel_2} := \frac{y_i^T z_i}{\parallel y_i \parallel_2 \parallel z_i \parallel_2},$$

with the constraints that $y, y_1, \cdots, y_{i-1}$ and $z, z_1, \cdots, z_{i-1}$ are linearly independent respectively.

For the sake of simplicity, if the columns of U_j and U_h are nonnegative normalized bases, then the cosine of the nonnegative principal angles between two subspaces $\mathrm{span}(U_j)$ and $\mathrm{span}(U_h)$ can be approximated as the first d_j (if $d_j \leq d_h$) maximums from different rows and columns of $U_j^T U_h$.

Definition 3.2. *The similarity between two subspaces is defined by*

$$\mathrm{sim}(\mathrm{span}(U_j), \mathrm{span}(U_h)) = \sqrt{\cos^2\theta_{j,h}^{(1)} + \cdots + \cos^2\theta_{j,h}^{(\min(d_j,d_h))}}.$$

Based on overlapping subspace clustering and the similarity between subspaces, we can partition the data matrix Y into $[Y_1\ Y_2\ \cdots\ Y_l]$ where $\sum_{j=1}^{l-1} sim(Y_j, Y_{j+1})$ is the maximum for all permutations. Note that $Y_1, Y_2, \cdots, Y_l$ are approximated as l linear subspaces. For this permutation, we may reasonably conclude that Y_j and Y_{j+1} have a few common nonnegative basis vectors. To capture the level of overlap (that is, the number of common nonnegative bases) between neighboring subspaces, we define a measure vector $M_c = [m_c^{1,2}\ m_c^{2,3}\ \cdots\ m_c^{l-1,l}]$ as follows:

$$m_c^{j,j+1} = \#\{i \mid \cos(\theta_{j,j+1}^{(i)}) > \sigma,\ i = 1, 2, \cdots, \min(d_j, d_{j+1})\}, \tag{3.2}$$

where σ is a threshold parameter and taken as 0.9 in this paper, $\#\{\cdot\}$ is the function that counts the number of elements of a finite set, and $j = 1, 2, \cdots, l-1$.

After obtaining the measure vector M_c, we suggest the factorization rank k is taken as

$$k = \sum_{j=1}^{l} d_j - \sum_{j=1}^{l-1} m_c^{j,j+1}. \tag{3.3}$$

This practice has two main advantages. One lies in the usage of prior information (i.e., the dimension d_j); the other one is to avoid using overlapping basis vectors to represent the same group data.

Accordingly, the block band structure of the coefficient matrix $X = [X_1\ X_2\ \cdots\ X_l]$ can be obtained based on levels of overlap. It is worth noting that the band property of the matrix X does not require the data samples to have been grouped together according to their subspace memberships. There is no loss of generality to assume that the indices of the samples have been rearranged to satisfy the true subspace memberships, because the Frobenius norm keeps the orthogonal invariance.

(3) Algorithms for band NMF

In this subsection, two algorithms for band NMF are presented.

The first one is based on the multiplicative update formulas mentioned in Section 2.2. To do this, the initial values of every block X_j ($j = 1, 2, \cdots, l$) are taken as

$$X_j(p,:) = \begin{cases} 1 & \text{if } \ 1 \le j \le l,\ q_j \le p \le q_j + d_j - 1, \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

Here, $X_j(p,:)$ denotes the p th row of the matrix X_j, and q_j is defined as follows:

$$q_j = \begin{cases} 1 & \text{if } \ j = 1, \\ \max(j, 1 + \sum_{i=1}^{j-1}(d_i - m_c^{i,i+1})) & \text{if } \ j = 2, 3, \cdots, l. \end{cases} \tag{3.5}$$

Algorithm 3.2 summarizes the whole procedure of band constrained multiplicative update formulas (band MU), where $\otimes(\oslash)$ indicates the element-wise product (division), and ε is taken as 10^{-9} in this paper.

Algorithm 3.2. Band Constrained Multiplicative Update Formulas (band MU).

1. Input data matrix Y, and the dimensions $d_j, j = 1, 2, \cdots, l$;
2. Obtain the measure vector M_c by (3.2), compute the value k by (3.3);
3. Initialize X by (3.4), and let $A = YX^T$;
4. Update A and X:
$$A = A \otimes (YX^T) \oslash (AXX^T + \varepsilon);$$
$$X = X \otimes (A^TY) \oslash (A^TAX + \varepsilon);$$
5. Repeat 4 until convergence.

Theorem 3.1. *Each iteration of the matrix X is a block band matrix, and the objective function $D_F(Y \parallel AX)$ of* Problem 3.1 *is nonincreasing under the band constrained multiplicative update formulas.*

Proof. As a result of the multiplicative update rule to keep the zero element invariant, and the start matrix X is a block band matrix, so each iteration of matrix X is block band matrix. From Theorem 2.1, we can obtain that the objective function $D_F(Y \parallel AX)$ is nonincreasing.

Table 3.1: The computing complexity of band MU.

MU update for $A^{(r+1)}$	MU update for $X^{(r+1)}$
$B1 = YX^{(r)^T} \longrightarrow \sum_{i=1}^{l} 2md_il_i$ flops	$B2 = A^{(r+1)^T}Y \longrightarrow 2mnk$ flops
$C1 = X^{(r)}X^{(r)^T} \longrightarrow \sum_{i=1}^{l} 2d_i^2l_i$ flops	$C2 = A^{(r+1)^T}A^{(r+1)} \longrightarrow 2mk^2$ flops
$D1 = A^{(r)}C1 \longrightarrow 2mk^2$ flops	$D2 = C2X^{(r)} \longrightarrow \sum_{i=1}^{l} 2kd_il_i$ flops
$A^{(r+1)} = A^{(r)} \otimes (B1 \oslash D1) \longrightarrow 2mk$ flops	$X^{(r+1)} = X^{(r)} \otimes (B2 \oslash D2) \longrightarrow \sum_{i=1}^{l} d_il_i$ flops

For each iteration, the computing complexity of band MU is presented in Table 3.1. Note that here l_i denotes the number of data points in the set S_i, and $n = \sum_{i=1}^{l} l_i$. Apparently, the total arithmetic operation for band MU is no more than $2km(n+2k+1)+2d_m n(m+d_m+k+1)$ flops, where $d_m = \max_i d_i$. However, the computing complexity for general NMF is $2k(2mn + 2mk + 2nk + m + n)$ flops, so the band NMF leads to a saving of computing complexity.

Since each iteration of ANLS computes an optimal solution of the nonnegativity constrained least squares subproblems, ANLS decreases the error the most among NMF algorithms in each iteration and every limit point of the sequence $\{A^{(r)}, X^{(r)}\}$ is a stationary point. In order to take advantage of the ANLS algorithm and the block band structure, a band alternating nonnegative least squares (band ANLS) method for band NMF, is described in Algorithm 3.3.

Algorithm 3.3. Band Alternating Nonnegative Least Squares (band ANLS).

1. Input data matrix Y, and the dimensions $d_j, j = 1, 2, \cdots, l$;
2. Obtain the measure vector M_c by (3.2), compute the value k by (3.3), and the q_j by (3.5);
3. Initialize nonnegative matrix $A \in R_+^{m\times k}$, $X = 0^{k\times n}$;
4. Update A and X:

$$\begin{aligned}
&\textbf{for } j = 1 : l\\
&s = q_j + d_j - 1,\ A_j = A(:, q_j : s),\\
&X_j(q_j : s, :) = \arg\min_{X'\geq 0} \| Y_j - A_j X' \|_F^2,\\
&\textbf{end}\\
&A = \arg\min_{A\geq 0} \| Y - AX \|_F^2;
\end{aligned}$$

5. Repeat 4 until convergence.

Theorem 3.2. *Any limit point of the sequence $\{A^{(r)}, X^{(r)}\}$ generated by Algorithm* 3.3 *is a stationary point of Problem* 3.1.

Proof. Suppose X^* is the limit point of the sequence $X^{(r)}$, since all variables of the coefficient matrix $X^{(r)}$ are in the band, and the bandwidth of the matrix $X^{(r)}$ in each iteration is invariant, then X^* is a block band matrix whose bandwidth is no more than the bandwidth of $X^{(r)}$. Directly from Corollary 2 of [21], we have that any limit point of the sequence $\{A^{(r)}, X^{(r)}\}$ is a stationary point.

4. Numerical Experiments

In this section, we investigate the performance of our proposed overlapping subspace clustering method (named OSC) by comparing to four popular methods, that is, RANSAC [7], LSA [23], LRR [10] and SSC [11]. For band NMF, we first compare the band MU with the band ANLS on different datasets, and then compare the approximate error and sparseness of band NMF with standard NMF [6], sparse NMFs [13] and NLRGS [15]. In addition, we apply

the band NMF to image compression to test its compressibility. The datasets we employed are detailed below.

Synthetic dataset: We generate 10 groups of data and each group contains 100 data points drawn from five overlapping subspaces in the following procedure: the bases U_i of each subspace are 30×8 nonnegative matrices with full column rank, and U_i and U_{i+1} have some common base vectors. The data points from each subspace are sampled by $Y_i = U_iR_i$, $1 \leq i \leq 5$, with R_i being a 8×20 matrix with uniform distribution.

YaleB dataset[1)] **:** This dataset has 38 subjects and around 64 near frontal images under different illuminations per subject. Each image is manually cropped and normalized to the size of 32×32 pixels. 20 images of each subject are randomly selected to form test data.

It should be pointed out that, in follow-up experiments, the evaluations are conducted by running 10 times and the average performance is recorded as the final result to remove the influence of randomness in the process of testing.

4.1. Experiments for overlapping subspace clustering

We do experiments on synthetic and Yale B data to test the performance of OSC. First, to observe how the performances vary between the feature dimensions (i.e., the rows of basis matrix), we carry out 6 trails on simple synthetic data in which the data points are drawn from $R^{10}, \cdots, R^{60}$ respectively with their levels of overlap being three. The clustering errors of LRR (Algorithm 2.1, where $\lambda = 0.1$) and OSC (where $d_j = 8$) are shown in Table 4.1. Next, to observe how the performances vary between the levels of overlap, we do experiments on the synthetic data points drawn from the first two subspaces whose levels of overlap range from 1 to 7, then report the clustering errors in Table 4.2. Finally, Table 4.3 presents the performance of OSC against other four popular subspace clustering algorithms when applied to the employed datasets.

Table 4.1: Clustering errors of LRR and OSC versus the feature dimension.

dimension	10	20	30	40	50	60
LRR	0.186	0.132	0.082	0.063	0.020	0.016
OSC	0.033	0.014	0.002	0.000	0	0

Table 4.2: Clustering errors of LRR and OSC versus the level of overlap.

level of overlap	1	2	3	4	5	6	7
LRR	0.000	0.006	0.013	0.065	0.125	0.162	0.280
OSC	0	0	0	0	0	0.000	0.150

Experimental results confirm that OSC makes an effective improvement of LRR in term of various feature dimensions and levels of overlap, and it obtains the best performance among all the competing algorithms on overlapping subspace clustering. Besides, it is noted that the value of d_j can be taken as a value higher than the true value if there is no prior information about d_j, because which can effectively avoid a confusion of neighboring subspaces especially when they have quite high levels of overlap.

[1)] http://vision.ucsd.edu/ leekc/ExtYaleDatabase/ExtYaleB.html:

Table 4.3: Clustering error of different algorithms on the synthetic and Yale B datasets.

results	LSA	RANSAC	LRR	SSC	OSC
Synthetic	0.088	0.218	0.034	0.076	0.000
YaleB(2 subjects)	0.225	0.2755	0.1	0.025	0.025
YaleB(3 subjects)	0.3667	0.3415	0.1167	0.1333	0.0833
YaleB(5 subjects)	0.51	0.4923	0.18	0.13	0.1
YaleB(10 subjects)	0.5643	0.6714	0.3650	0.2929	0.2571

4.2. Experiments for band NMF

To observe the performances of the band MU and band ANLS, Fig. 4.1 displays the evolution of the Frobenius errors ($\frac{1}{2}\| Y - AX \|_F^2$) of the band NMF: on the left, the synthetic dataset, and, on the right, the YaleB dataset with $m = 1024$ and $n = 200$. We observe that: 1) the band MU converges rather slowly, but costs less computation time. 2) the band ANLS converges very fast and obtains a smaller approximate error, but each iteration is time consuming.

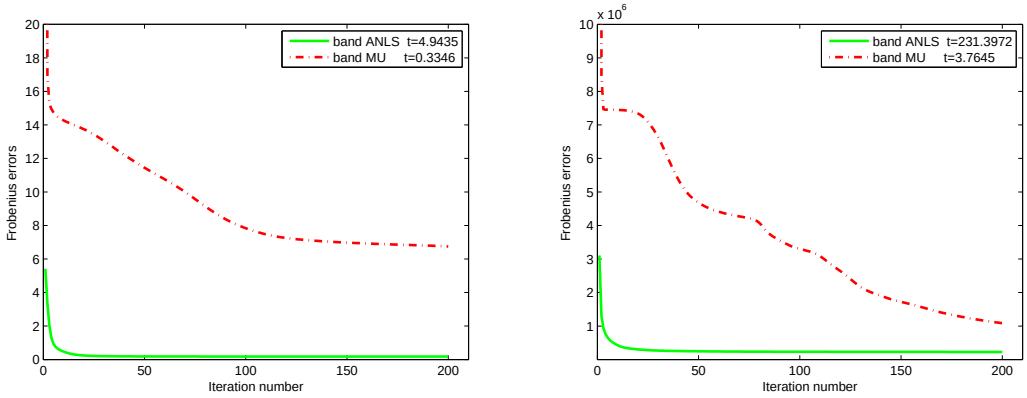


Fig. 4.1. Comparison of band MU and band ANLS. Here, t (second) indicates the time required.

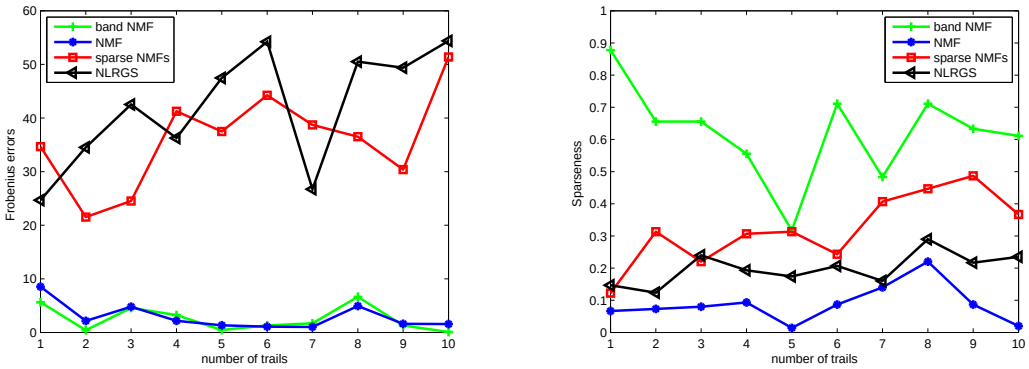
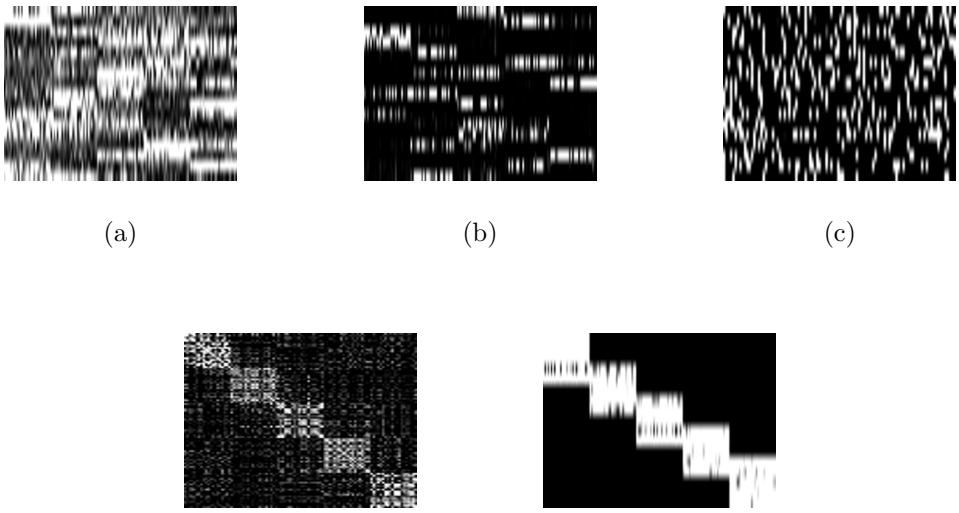


Fig. 4.2. Frobenius errors (left), sparseness (right) of NMF, sparse NMFs, NLRGS and band NMF on 10 trails



(a) (b) (c)

(d) (e)

Fig. 4.3. The structure of coefficient matrix X by NMF (a), sparse NMFs (b), NLRGS (c), LRR (d) and band NMF (e) when samples are from a union of several overlapping subspaces.

Moreover, we use synthetic dataset to compare the Frobenius errors and sparseness (that is, the number of the zero elements in X over kn) of NMF, sparse NMFs (parameter $\beta = 10$), NLRGS (λ =0.1, μ = 0.008 and ρ = 1.05) and band NMF (using band MU algorithm) on 10 trails, in which all of the used NMF methods have the same factorization rank k. Fig. 4.2 shows that band NMF has the highest sparseness, leading to the final errors which is comparable with the values reached by NMF. In other words, band NMF provides a optimal sparse representation among these models. Besides, Fig. 4.3 shows the structure of coefficient matrix X of all factorization models. From the results, we can see that band NMF captures the real data structure nicely but other models do not.

4.3. Experiments for image compression

In this section, we build a collection of images, which is presented in Fig. 4.4. The first five images are web images with gradually-changed color and the last five are from BSDS 300 databases. SSIM [27] is used to evaluate their performance, and all of NMF models have the same value of k. As can be seen from Table 4.4, band NMF achieves the best performance among all the competing NMF methods. In fact, band NMF stores the least data to get these compressed images. To better understand the behavior of band NMF, Fig. 4.5 shows the sample compressed images using band NMF, NMF and sparse NMFs respectively. The experimental results also show that band NMF approximates the ground truth images more accurately than the other two methods.

Fig. 4.4. Ten employed images

Table 4.4: Compression performances of three competing NMF methods on the ten employed images.

SSIM	image 1	image 2	image 3	image 4	image 5
NMF	0.7813	0.8184	0.8093	0.8499	0.7862
sparse NMFs	0.9774	0.8980	0.8957	0.8672	0.8268
band NMF	0.9866	0.9479	0.9768	0.9136	0.8291
SSIM	image 6	image 7	image 8	image 9	image 10
NMF	0.8190	0.7209	0.6055	0.6604	0.7503
sparse NMFs	0.8003	0.7586	0.8334	0.9058	0.8673
band NMF	0.9450	0.8567	0.8680	0.9340	0.9184

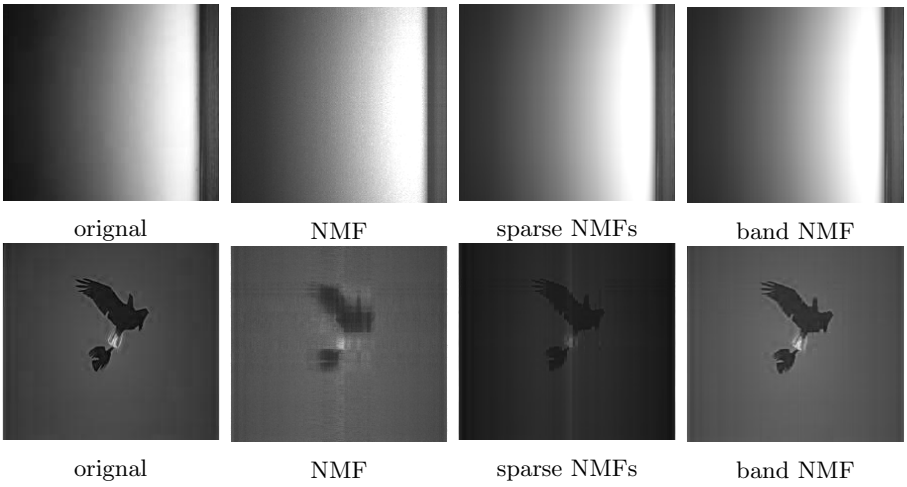


Fig. 4.5. Sample compressed images by NMF, sparse NMFs and band NMF.

5. Conclusion

In this paper we proposed a band NMF model to extend a single low rank subspace model to a mixture of several overlapping low rank subspaces. To efficiently solve band NMF, we

developed two algorithms in the frame of the MU and the ANLS by capturing the level of overlap between two neighbouring subspaces. We conducted different numerical experiments to verify that band NMF enhances the performance of NMF in data representation and processing. In the future, we will explore a better way to reduce the impact from the clustering method to band NMF, and explore the applications of this idea on other methods.

Acknowledgments. The authors thank the anonymous reviewers for their valuable comments and suggestions that helped improve the quality of this paper.

References

[1] 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, Chichester, UK: John Wiley & Sons, Ltd, 2009.

[2] G. Zhou, A. Cichocki, Q. Zhao, and S. Xie, Nonnegative matrix and tensor factorizations: An algorithmic perspective, *IEEE Signal Processing Magazine*, **31** (2014), 54-65.

[3] L.I. Rudin, S. Osher, and E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D: Nonlinear Phenomena*, **60** (1992), 259-268.

[4] R. Basri and D.W. Jacobs, Lambertian reflectance and linear subspaces, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25** (2003), 218-233.

[5] Y.W. Lu, Z.H. Lai, Y. Xu, J.N. You, X.L. Li and C. Yuan, Projective robust nonnegative factorization, *Information Sciences*, **s 364-365** (2016), 16-32.

[6] D. Lee, S. Sueng, Algorithms for nonnegative matrix factorization, *Adv. Neural Inf. Process. Syst.*, **13** (2001), 556-562.

[7] M.A. Fischler and R.C. Bolles, RANSAC Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, *Comm. ACM*, **26** (1981), 381-395.

[8] H. Laurberg, M.N. Schmidt, M.G. Christensen, and S. H. Jensen, Structured non-negative matrix factorization with sparsity patterns, *Proceedings Asilomar Conference on Signals, Systems, and Computers*, 2008: 1693-1697.

[9] L. Du, X. Li, and Y. Shen, Robust nonnegative matrix factorization via half-quadratic minimization, *IEEE ICDM*, **5** (2012), 201-210.

[10] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, Robust recovery of subspace structures by low-rank representation, *IEEE Trans. Pattern Anal. Mach. Intell.*, **35** (2013), 171-184.

[11] E. Elhamifar, R. Vidal, Sparse subspace clustering: algorithm, theory, and applications, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **35** (2012), 2765-2781.

[12] D. Cai, X. He, J. Han and T.S. Huang, Graph regularized nonnegative matrix factorization for data representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **33** (2011), 1548-1560.

[13] H. Kim and H. Park, Sparse non-negative matrix factorizations via alternating non-negativity constrained least squares for microarray data analysis, *Bioinformatics*, **23** (2007), 1495-1502.

[14] X.B. Pei, Y.T. Wu, Neighbors isometric embedding nonnegative matrix factorization for image representation, *Multidimensional Systems and Signal Processing*, 2015: 1-19.

[15] S. Wu, X. Zhang, N. Guan, D. Tao, X. Huang, and Z. Luo, Non-negative Low-Rank and Group-Sparse Matrix Factorization, Springer International Publishing, 2015.

[16] X. Zhang, F. Sun, G. Liu and Y. Ma, Fast Low-Rank Subspace Segmentation, *IEEE Transactions on Knowledge & Data Engineering*, **26** (2014), 1293-1297.

[17] K. Tang, R. Liu, Z. Su, and J. Zhang, Structure-constrained low-rank representation, *IEEE TNNL-S*, **25** (2014), 2167-2179.

[18] A. Ng, M. Jordan and Y. Weiss, On spectral clustering: Analysis and an algorithm, *Advances in Neural Information Processing Systems*, **24** (2002), 849-856.

[19] J. Shi and J. Malik, Normalized Cuts and Image Segmentation, *IEEE Trans. Pattern Analysis and Machine Intelligence*, **22** (2000), 888-905.

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

[21] L. Grippo and M. Sciandrone, On the convergence of the block nonlinear Gauss- Seidel method under convex constraints, *Operations Research Letters*, **26** (2000), 127-136.

[22] H. Kim and H. Park, Non-negative Matrix Factorization Based on Alternating Non-negativity Constrained Least Squares and Active Set Method, *SIAM J. on Matrix Analysis and Applications*, **30** (2008), 713-730.

[23] J. Yan and M. Pollefeys, A General Framework for Motion Segmentation: Independent, Articulated, Rigid, Non-Rigid, Degenerate and Non-Degenerate, Proc. European Conf. Computer Vision, 2006.

[24] D. Langr, I. Šimeček, P. Tvrdík, T. Dytrych, and J.P. Draayer, Adaptive blocking hierarchical storage format for sparse matrices, *IEEE Xplore Digital Library*, **11** (2012), 545-551.

[25] I. Šimeček, D. Langr and P. Tvrdík, Space efficient formats for structure of sparse matrices based on tree structures, *IEEE Computer Society*, 2013: 344-351.

[26] T. Zhang, A. Szlam, Y. Wang, and G. Lerman, Hybrid Linear Modeling via Local Best-Fit Flats, *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, **100** (2010), 1927-1934.

[27] Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Transactions on Image Processing*, **13** (2004), 600-612.