Tangent Space Based Alternating Projections for Nonnegative Low Rank Matrix Approximation

Guangjing Song, Michael K. Ng, and Tai-Xiang Jiang

Abstract—In this paper, we develop a new alternating projection method to compute nonnegative low rank matrix approximation for nonnegative matrices. In the nonnegative low rank matrix approximation method, the projection onto the manifold of fixed rank matrices can be expensive as the singular value decomposition is required. We propose to use the tangent space of the point in the manifold to approximate the projection onto the manifold in order to reduce the computational cost. We show that the sequence generated by the alternating projections onto the tangent spaces of the fixed rank matrices manifold and the nonnegative matrix manifold, converge linearly to a point in the intersection of the two manifolds where the convergent point is sufficiently close to optimal solutions. This convergence result based inexact projection onto the manifold is new and is not studied in the literature. Numerical examples in data clustering, pattern recognition and hyperspectral data analysis are given to demonstrate that the performance of the proposed method is better than that of nonnegative matrix factorization methods in terms of computational time and accuracy.

Index Terms—Alternating projection method, manifolds, tangent spaces, nonnegative matrices, low rank, nonnegativity.

1 Introduction

Nonnegative data matrices appear in many data analysis applications. For instance, in image analysis, image pixel values are nonnegative and the associated nonnegative image data matrices can be formed for clustering and recognition [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]. In text mining, the frequencies of terms in documents are nonnegative and the resulted nonnegative term-to-document data matrices can be constructed for clustering [13], [14], [15], [16]. In bioinformatics, nonnegative gene expression values are studied and nonnegative gene expression data matrices are generated for diseases and genes classification [17], [18], [19], [20], [21]. Low rank matrix approximation for nonnegative matrices plays a key role in all these applications. Its main purpose is to identify a latent feature space for objects representation. The classification, clustering or recognition analysis can be done by using these latent features.

Nonnegative Matrix Factorization (NMF) has emerged in 1994 by Paatero and Tapper [22] for performing environmental data analysis. The purpose of NMF is to decompose an input m-by-n nonnegative matrix $\mathbf{A} \in \mathbb{R}_+^{m\times n}$ into m-by-r nonnegative matrix $\mathbf{B} \in \mathbb{R}_+^{m\times r}$ and r-by-n nonnegative matrix $\mathbf{C} \in \mathbb{R}_+^{r\times n}$: $\mathbf{A} \approx \mathbf{BC}$, and more precisely

$$\min_{\mathbf{B},\mathbf{C}\geq 0} \|\mathbf{A} - \mathbf{BC}\|_F^2, \tag{1}$$

where $\mathbf{B}, \mathbf{C} \geq 0$ means that each entry of $\mathbf{B}$ and $\mathbf{C}$ is nonnegative, $\|\cdot\|_F$ is the Frobenius norm of a matrix, and r (the low rank value) is smaller than m and n. Lee and Seung [8] proposed a simple yet effective algorithm with multiplicative update (MU) rules to solve model (1), i.e., minimizing the Frobenius norm between the given nonnegative matrix A and its approximation BC. Their emphasis on the potential value of the parts-based representation brought by NMF largely popularized it.

So far, numerous amounts of effort have been devoted to solve (1). Several well-known and widely used NMF algorithms have been presented, to name a few, the hierarchical alternating least squares (HALS) [23], the alternating nonnegative least squares (ANLS) [24], the accelerated versions of MU and HALS [24], the projected gradient (PG) method and its accelerated version (A-PG) [25], the Nesterov's optimal gradient method (NeNMF) [26], the active set method [27], and the version accelerated via block principal pivoting [28]. In general, the solution of those iterative algorithms may vary with different initializations. Many approaches focused on the initialization of NMF based on k-means and spherical k-means [29], rank-one approximations [30], the nonnegative singular value decomposition (NNSVD) [31]. Meanwhile, additional constraints can be imposed as regularization into (1), e.g., the sparsity [32], [33], [34], the orthogonality [4], [35], [36], the symmetry [7], [37], [38], [39], the separability [40], [41], [42], the discriminant [43], [44], [45], the local topological property [46], [47], [48], [49], etc. Moreover, the factorization paradigm of NMF is not limited in the format of (1) and new ones, such as the nonnegative matrix tri-factorization [4], [50], [51], the deep nonnegative matrix factorization [52], [53], [54], the non-negative tensor factorization [55], [56], [57], the recent pioneering disen-

- *Guangjing Song is with School of Mathematics and Information Sciences, Weifang University, Weifang 261061, P.R. China (email: sgjshu@163.com).*
M. K. Ng is with Department of Mathematics, The University of Hong Kong, Pokfulam, Hong Kong (e-mail: mng@maths.hku.hk).
T.-X. Jiang is with School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, Chengdu, Sichuan, P.R.China (e-mails: taixiangjiang@gmail.com, jiangtx@swufe.edu.cn).
G.J. Song's research is supported in part by the National Natural Science Foundation of China under Grant 12171369 and Key NSF of Shandong Province under Grant ZR2020KA008.
M. Ng's research supported in part by the HKRGC GRF 12300218, 12300519, 17201020, 17300021, C1013-21GF, C7004-21GF and Joint NSFC-RGC N-HKU76921.
T.-X. Jiang's research is supported in part by the National Natural Science Foundation of China (12001446) and Fundamental Research Funds for the Central Universities (JBK2202049, JBK2102001).
The Corresponding Author.

tangled factorization [58], are constantly emerging. Accordingly, above mentioned NMF techniques and their variants have shown promising capacity on different applications in various fields, from text data mining [13], [15], [16], [51], image classification [6], [44], and face recognition [2], [5], [11], [12], [43], to multi-view clustering [45], [48], [52], [59], blind source separation [9], [17], [60], [61], social computing [39], [50]. For a comprehensive review of the development of NMF, we refer to the recently edited books [62], [63] and review papers [64], [65].

In [66], Song and Ng proposed a new algorithm for computing nonnegative low rank matrix (NLRM) approximation for nonnegative matrices. This approach is completely different from NMF, aiming to find a nonnegative low rank matrix $\mathbf{X}$ such that the difference between $\mathbf{X}$ and the given nonnegative matrix $\mathcal{A}$ is as small as possible. The distance $\|\mathbf{A}-\mathbf{X}\|_F^2$ can be smaller than $\|\mathbf{A}-\mathbf{BC}\|_F^2$, where $\mathbf{B}$ and $\mathbf{C}$ are two nonnegative matrices determined via solving (1). This implies that directly finding $\mathbf{A}$ could obtain a better low rank matrix approximation, which would be very important in many applications [56], [67]. Mathematically, the nonnegative low rank matrix approximation can be formulated as the following optimization problem

$$\min_{\mathrm{rank}(\mathbf{X})=r,\mathbf{X}\geq 0} \|\mathbf{A}-\mathbf{X}\|_{\mathrm{F}}^2. \tag{2}$$

The convergence of the their algorithm is studied and proved. Experimental results for synthetic data and face images are presented to demonstrate the performance of NLRM is better than state-of-the-art NMF methods. In addition, the NLRM method admits a matrix singular value decomposition (SVD) automatically which provides a significant index based on singular values that can be used to identify important singular basis vectors, while this information cannot be obtained by the classical NMF methods.

1.1 The Contribution

In the algorithm proposed in [66], a projection on the fixed-rank matrices manifold and a projection onto the nonnegative matrices manifold are used alternately to compute a nonnegative low rank approximation of the given nonnegative matrix. The computational cost of the above alternating projection method is dominant by the calculation of the singular value truncations of the matrices derived at each iteration. The computation burden could be very high when the matrix size is relatively large.

In this paper, also considering the nonnegative low-rank matrix approximation, we propose to use the tangent space of the point in the manifold to approximate the projection onto the manifold that can reduce the computational cost. We show that the sequence generated by the new alternating projections converges linearly to a point in the intersection of the two manifolds. Moreover, the convergent point is proved sufficiently close to one of the optimal solutions. Numerical examples will be presented to demonstrate that the computational time of the proposed tangent space based method is less than that of the original alternating projection method proposed in [66]. Moreover, experimental results in data clustering, pattern recognition and hyperspectral data analysis, are given to demonstrate that the performance of the proposed method is better than that of other nonnegative matrix factorization methods in terms of computational time and accuracy.

The rest of this paper is organized as follows. In Section 2, we propose tangent space based alternating projection method. In Section 3, we show the convergence of the proposed method. In Section 4, numerical examples are given to show the advantages of the proposed method. Finally, some concluding remarks are given in Section 5.

2 Nonnegative Low Rank Matrix Approximation

In this paper, we are interested in the $m \times n$ fixed-rank matrices manifold

$$\mathcal{M}_r := \left\{\mathbf{X}\in\mathbb{R}^{m\times n},\ \mathrm{rank}(\mathbf{X})=r\right\}, \tag{3}$$

the $m \times n$ non-negativity matrices manifold

$$\mathcal{M}_n := \left\{\mathbf{X}\in\mathbb{R}^{m\times n}, \mathbf{X}_{ij}\geq 0,\ i=1,\cdots,m,\ j=1,\cdots,n\right\}, \tag{4}$$

and the $m \times n$ nonnegative fixed rank matrices manifold

$$\mathcal{M}_{rn} = \mathcal{M}_r \cap \mathcal{M}_n = \left\{\mathbf{X}\in\mathbb{R}^{m\times n},\ \mathrm{rank}(\mathbf{X})=r,\ \mathbf{X}_{ij}\geq 0,\ i=1,...,m,\ j=1,...,n\right\}. \tag{5}$$

The proof of $\mathcal{M}_{rn}$ is a manifold can be found in [66]. Let $\mathbf{X}\in\mathbb{R}^{m\times n}$ be an arbitrary matrix in the manifold $\mathcal{M}_r$. Assume that the SVD of $\mathbf{X}$ is denoted as: $\mathbf{X}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ where $\mathbf{U}\in\mathbb{R}^{m\times r}$, $\mathbf{\Sigma}\in\mathbb{R}^{r\times r}$, and $\mathbf{V}\in\mathbb{R}^{n\times r}$. Then by Proposition 2.1 in [68] the tangent space of $\mathcal{M}_r$ at $\mathbf{X}$ can be expressed as

$$T_{\mathcal{M}_r}(\mathbf{X}) = \{\mathbf{U}\mathbf{W}^T + \mathbf{Z}\mathbf{V}^T\}, \tag{6}$$

where $\mathbf{W}\in\mathbb{R}^{n\times r}$ and $\mathbf{Z}\in\mathbb{R}^{m\times r}$ are arbitrary. Here $\cdot^T$ denotes the transpose of a matrix. For a given m-by-n matrix $\mathbf{Y}$, the orthogonal projection of $\mathbf{Y}$ onto the subspace $T_{\mathcal{M}_r}(\mathbf{X})$ can be written as

$$P_{T_{\mathcal{M}_r}(\mathbf{X})}(\mathbf{Y}) = \mathbf{U}\mathbf{U}^T\mathbf{Y} + \mathbf{Y}\mathbf{V}\mathbf{V}^T - \mathbf{U}\mathbf{U}^T\mathbf{Y}\mathbf{V}\mathbf{V}^T. \tag{7}$$

The alternating projection method studied in [66] is based on projecting the given nonnegative matrix onto the $m \times n$ fixed-rank matrices manifold $\mathcal{M}_r$ and the non-negativity matrices manifold $\mathcal{M}_n$ iteratively. The projection onto the fixed rank matrix set $\mathcal{M}_r$ is derived by the Eckart-Young-Mirsky theorem [69] which can be expressed as follows:

$$\pi_1(\mathbf{X}) = \sum_{i=1}^{r}\sigma_i(\mathbf{X})u_i(\mathbf{X})v_i^T(\mathbf{X}), \tag{8}$$

where $\sigma_i(\mathbf{X})$ is the i-th singular value of $\mathbf{X}$, $u_i(\mathbf{X})$ and $v_i(\mathbf{X})$ are their corresponding singular vectors. The projection onto the nonnegative matrix set $\mathcal{M}_n$ is expressed as

$$\pi_2(\mathbf{X}) = \begin{cases} \mathbf{X}_{ij}, & \text{if } \mathbf{X}_{ij}\geq 0,\\ 0, & \text{if } \mathbf{X}_{ij}<0. \end{cases} \tag{9}$$

Moreover, $\pi(\mathbf{X})$ refers to a matrix on $\mathcal{M}_{rn}$ which is closest to the given nonnegative matrix $\mathbf{X}$, i.e.,

$$\pi(\mathbf{X}) = \underset{\mathbf{Y}\in\mathcal{M}_{rn}}{\mathrm{argmin}}\|\mathbf{X}-\mathbf{Y}\|_F^2, \tag{10}$$

where $\mathcal{M}_{rn}$ is the nonnegative fixed rank matrices manifold given as in (5).

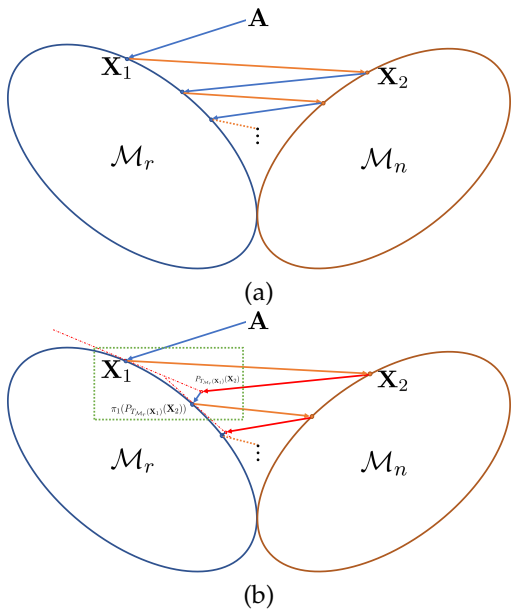


Fig. 1. The comparison between (a) the original alternating projection method and (b) the proposed TAP method.

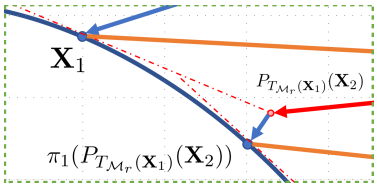


Fig. 2. The zoomed region in Figure 1(b).

Algorithm 1 Tangent spaces based Alternating Projection (TAP) Method

Input: Given a nonnegative matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$ this algorithm computes nearest rank-r nonnegative matrix.
1: Initialize $\mathbf{X}_0 = \mathbf{A}$;
2: $\mathbf{X}_1 = \pi_1(\mathbf{X}_0)$ and $\mathbf{X}_2 = \pi_2(\mathbf{X}_1)$
3: for k=1,2,...,
4: $\quad \mathbf{X}_{2k+1} = \pi_1(P_{T_{\mathcal{M}_r}(\mathbf{X}_{2k-1})}(\mathbf{X}_{2k}))$
5: $\quad \mathbf{X}_{2k+2} = \pi_2(\mathbf{X}_{2k+1})$;
6: **end**
Output: $\mathbf{X}_{2k+1}$ when the stopping criterion is satisfied.

2.1 Projections Based on Tangent Spaces

The main aim of this section is to introduce the Tangent space based Alternating Projection (TAP) method. In the original alternating projection (AP) method proposed in [66], the projection onto the fixed rank matrix manifold in computed by the singular values truncation operator given in (8). Unfortunately, it is expensive when the matrix size is big. Then in this section, we will make use of tangent spaces to design the TAP method to compute the nonnnegative low rank matrix approximation which can reduce the computational cost.

The difference between the AP method and the TAP method is illustrated in Figure 1 and Figure 2. For the TAP method, the given nonnegative matrix $\mathbf{X_0} = \mathbf{A}$ was first projected onto the manifold $\mathcal{M}_r$ by $\pi_1(\cdot)$, i.e., $\mathbf{X}_1 = \pi_1(\mathbf{X_0})$, and then $\mathbf{X}_2$ is derived by projecting $\mathbf{X}_1$ onto the manifold $\mathcal{M}_n$ by $\pi_2(\cdot)$. The first two steps are same as the original AP method. The difference between the two methods starts from the third step. In the TAP method, the point $\mathbf{X}_2$ is first projected onto the tangent space of the manifold $\mathcal{M}_r$ at $\mathbf{X}_1$ by the orthogonal projection $P_{T_{\mathcal{M}_r}(\mathbf{X}_1)}(\cdot)$, and then the derived point is projected from the tangent space to the manifold $\mathcal{M}_r$, i.e., $\mathbf{X}_3 = \pi_1(P_{T_{\mathcal{M}_r}(\mathbf{X}_1)}(\mathbf{X}_2))$. Thus the sequence generated by the TAP method can be derived as follows:

$$\mathbf{X}_0 = \mathbf{A},\ \mathbf{X}_1 = \pi_1(\mathbf{X}_0),\ \mathbf{X}_2 = \pi_2(\mathbf{X}_1),$$
$$\mathbf{X}_3 = \pi_1(P_{T_{\mathcal{M}_r}(\mathbf{X}_1)}(\mathbf{X}_2)),\ \mathbf{X}_4 = \pi_2(\mathbf{X}_3),\ \cdots,$$
$$\mathbf{X}_{2k+1} = \pi_1(P_{T_{\mathcal{M}_r}(\mathbf{X}_{2k-1})}(\mathbf{X}_{2k})),\ \mathbf{X}_{2k+2} = \pi_2(\mathbf{X}_{2k+1}),\ \cdots$$

where $P_{T_{\mathcal{M}_r}(\mathbf{X}_{2k-1})}(\mathbf{X}_{2k})$ denotes the orthogonal projections of $\mathbf{X}_{2k}$ onto the tangent space of $\mathcal{M}_r$ at $\mathbf{X}_{2k-1}$. The algorithm can be summarized as the following algorithm.

Let's analyze the computational cost of each step of the TAP algorithm. Suppose the skinny SVD decompositions of $\mathbf{X}_{2k-1}$ are given as $\mathbf{X}_{2k-1} = \mathbf{U}_k\mathbf{\Sigma}_k\mathbf{V}_k^T, k = 1, \cdots$. By (6), the tangent space of $\mathcal{M}_r$ at $\mathbf{X}_{2k-1}$ can be expressed as

$$T_{\mathcal{M}_r}(\mathbf{X}_{2k-1}) = \{\mathbf{U}_k\mathbf{W}^T + \mathbf{Z}\mathbf{V}_k^T\},$$

where $\mathbf{W} \in \mathbb{R}^{n\times r}$ and $\mathbf{Z} \in \mathbb{R}^{m\times r}$ are arbitrary. By (7), $\mathbf{X}_{2k}$ can be projected onto the subspace $T_{\mathcal{M}_r(\mathbf{X}_{2k-1})}$ as follows:

$$P_{T_{\mathcal{M}_r(\mathbf{X}_{2k-1})}}(\mathbf{X}_{2k}) = \mathbf{U}_k\mathbf{U}_k^T\mathbf{X}_{2k} + \mathbf{X}_{2k}\mathbf{V}_k\mathbf{V}_k^T - \mathbf{U}_k\mathbf{U}_k^T\mathbf{X}_{2k}\mathbf{V}_k\mathbf{V}_k^T.$$

Suppose the QR decompositions of $(\mathbf{I} - \mathbf{U}_k\mathbf{U}_k^T)\mathbf{X}_{2k}\mathbf{V}_k$ and $(\mathbf{I} - \mathbf{V}_k\mathbf{V}_k^T)\mathbf{X}_{2k}\mathbf{U}_k$ are given as

$$(\mathbf{I} - \mathbf{U}_k\mathbf{U}_k^T)\mathbf{X}_{2k}\mathbf{V}_k = \mathbf{Q}_k\mathbf{R}_k$$

and

$$(\mathbf{I} - \mathbf{V}_k\mathbf{V}_k^T)\mathbf{X}_{2k}^T\mathbf{U}_k = \hat{\mathbf{Q}}_k\hat{\mathbf{R}}_k,$$

respectively. Recall that $\mathbf{U}_k^T\mathbf{Q}_k = \mathbf{V}_k^T\hat{\mathbf{Q}}_k = \mathbf{0}$, then by a direct computation, we have

$$\begin{aligned}
&P_{T_{\mathcal{M}_r(\mathbf{X}_{2k-1})}}(\mathbf{X}_{2k}) \\
&= \mathbf{U}_k\mathbf{U}_k^T\mathbf{X}_{2k}(\mathbf{I} - \mathbf{V}_k\mathbf{V}_k^T) + (\mathbf{I} - \mathbf{U}_k\mathbf{U}_k^T)\mathbf{X}_{2k}\mathbf{V}_k\mathbf{V}_k^T \\
&\quad + \mathbf{U}_k\mathbf{U}_k^T\mathbf{X}_{2k}\mathbf{V}_k\mathbf{V}_k^T \\
&= \mathbf{U}_k\hat{\mathbf{R}}_k^T\hat{\mathbf{Q}}_k^T + \mathbf{Q}_k\mathbf{R}_k\mathbf{V}_k^T + \mathbf{U}_k\mathbf{U}_k^T\mathbf{X}_{2k}\mathbf{V}_k\mathbf{V}_k^T \\
&= \begin{pmatrix} \mathbf{U}_k & \mathbf{Q}_k \end{pmatrix} \begin{pmatrix} \mathbf{U}_k^T\mathbf{X}_{2k}\mathbf{V}_k & \hat{\mathbf{R}}_k^T \\ \mathbf{R}_k & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{V}_k^T \\ \hat{\mathbf{Q}}_k^T \end{pmatrix} \\
&:= \begin{pmatrix} \mathbf{U}_k & \mathbf{Q}_k \end{pmatrix} \mathbf{M}_k \begin{pmatrix} \mathbf{V}_k^T \\ \hat{\mathbf{Q}}_k^T \end{pmatrix}.
\end{aligned}$$

Let $\mathbf{M}_k = \mathbf{\Psi}_k\mathbf{\Gamma}_k\mathbf{\Phi}_k^T$ be the skinny SVD of $\mathbf{M}_k$ which can be computed using $O(r^3)$ flops. Note that $(\mathbf{U}_k, \mathbf{Q}_k)$ and $\left(\mathbf{V}_k, \hat{\mathbf{Q}}_k\right)$ are orthogonal, then the skinny SVD of

$$P_{T_{\mathcal{M}_r(\mathbf{X}_{2k-1})}}(\mathbf{X}_{2k}) = \mathbf{\Omega}_k\mathbf{\Theta}_k\mathbf{\Upsilon}_k^T$$

can be computed by

$$\mathbf{\Omega}_k = (\mathbf{U}_k, \mathbf{Q}_k)\,\mathbf{\Psi}_k, \mathbf{\Theta}_k = \mathbf{\Gamma}_k \text{ and } \mathbf{\Upsilon}_k = \left(\mathbf{V}_k, \hat{\mathbf{Q}}_k\right)\mathbf{\Phi}_k.$$

It follows that the overall computational cost of $\pi_1(P_{T_{\mathcal{M}_r}(\mathbf{X}_{2k-1})}(\mathbf{X}_{2k}))$ can be expressed as two matrix-matrix multiplications. In addition, the calculation procedure involves the QR decomposition of two matrices of sizes $m \times r$ and $n \times r$ matrices, and the SVD of a matrix of size $2r \times 2r$. The total cost per iteration is of $4mnr + O(r^2m + r^2n + r^3)$. In contrast, the computation of the best rank-r approximation of a non-structured $m \times n$ matrix costs $O(mnr) + mn$ flops where the constant in front of mnr can be very large. In practice, the cost per iteration of the proposed TAP method is less than that of original alternating projection method. In Section 4, numerical examples will be given to demonstrate the total computational time of the proposed TAP method is less than that of the original alternating projection method.

3 The Convergence Analysis

In this section, we mainly consider the convergence of the proposed TAP method. The convergence of the original alternating projection method relate to two manifolds has been proved in [70]. Known from that, the angle of a point in the intersection of two manifolds plays a key role in the whole proof process. In our setting, for $\mathbf{B} \in \mathcal{M}_{rn}$, its angle $\alpha(\mathbf{B})$ can be defined as

$$\alpha(\mathbf{B}) = cos^{-1}(\sigma(\mathbf{B})) \tag{11}$$

where

$$\sigma(\mathbf{B}) = \lim_{\xi \to 0} \sup_{\mathbf{B}_1 \in F_1^{\xi}(\mathbf{B}), \mathbf{B}_2 \in F_2^{\xi}(\mathbf{B})} \left\{ \frac{\langle \mathbf{B}_1 - \mathbf{B}, \mathbf{B}_2 - \mathbf{B} \rangle}{\|\mathbf{B}_1 - \mathbf{B}\|_F \|\mathbf{B}_2 - \mathbf{B}\|_F} \right\},$$

with

$$F_1^{\xi}(\mathbf{B}) = \{\mathbf{B}_1 \mid \mathbf{B}_1 \in \mathcal{M}_r \backslash \mathbf{B}, \|\mathbf{B}_1 - \mathbf{B}\|_F \leq \xi, \mathbf{B}_1 - \mathbf{B} \perp T_{\mathcal{M}_r \cap \mathcal{M}_n}(\mathbf{B})\},$$

$$F_2^{\xi}(\mathbf{B}) = \{\mathbf{B}_2 \mid \mathbf{B}_2 \in \mathcal{M}_n \backslash \mathbf{B}, \|\mathbf{B}_2 - \mathbf{B}\|_F \leq \xi, \mathbf{B}_2 - \mathbf{B} \perp T_{\mathcal{M}_r \cap \mathcal{M}_n}(\mathbf{B})\},$$

and $T_{\mathcal{M}_r \cap \mathcal{M}_n}(\mathbf{B})$ is the tangent space of $\mathcal{M}_r \cap \mathcal{M}_n$ at point $\mathbf{B}$. The angle can be calculated by the two points in $\mathcal{M}_r$ and $\mathcal{M}_n$. A point $\mathbf{B}$ in $\mathcal{M}_{rn}$ is nontangential if $\alpha(\mathbf{B})$ has a positive angle, i.e., $0 \leq \sigma(A) < 1$.

In the following, the main convergence results of Algorithm 1 can be listed as follows.

Theorem 3.1. *Let $\mathcal{M}_r$, $\mathcal{M}_n$ and $\mathcal{M}_{rn}$ be given as* (3), (4) *and* (5), *the projections onto $\mathcal{M}_r$ and $\mathcal{M}_n$ be given as* (8) *and* (9), *respectively. Suppose that $\mathbf{P} \in \mathcal{M}_{rn}$ is a non-tangential intersection point, then for any given $\epsilon > 0$ and $1 > c > \sigma(\mathbf{P})$, there exist an $\xi > 0$ such that for any $\mathbf{A} \in Ball(\mathbf{P}, \xi)$ (the ball neighborhood of $\mathbf{P}$ with radius ξ contains the given nonnegative matrix $\mathbf{A}$), the sequence $\mathbf{X}_k$ generated by Algorithm 1 converges to a point $\mathbf{X}_\infty \in \mathcal{M}_{rn}$, and satisfy*
(1) $\|\mathbf{X}_\infty - \pi(\mathbf{A})\|_F \leq \epsilon \|\mathbf{A} - \pi(\mathbf{A})\|_F$,
(2) $\|\mathbf{X}_\infty - \mathbf{X}_k\|_F \leq \text{const} \cdot c^k \|\mathbf{A} - \pi(\mathbf{A})\|_F$,
where $\pi(\mathbf{A})$ is defined in (10).

When the points on the tangent spaces are used as approximation of the points in the manifold, the following results can help us to study the distances related to the proof of Theorem 3.1.

Lemma 3.2 (Proposition 4.3 and Theorem 4.1 in [70])**.** *Let $\pi_1(\cdot)$ and $\pi(\cdot)$ be defined as* (8) *and* (10), *and $\mathbf{P} \in \mathcal{M}_r$. For each $0 < \epsilon < \frac{3}{5}$, there exist an $s(\epsilon) > 0$ and an $\varepsilon(\epsilon) > 0$, such that for any given $\mathbf{Z} \in Ball(\mathbf{P}, s(\epsilon))$,*

$$\|\pi_1(\mathbf{Z}) - P_{T_{\mathcal{M}_r}(\pi(\mathbf{Z}))}(\mathbf{Z})\|_F < 4\sqrt{\epsilon}\|\mathbf{Z} - \pi(\mathbf{Z})\|_F, \tag{12}$$

and

$$\|\pi(\pi_1(\mathbf{Z})) - \pi(\mathbf{Z})\|_F < \varepsilon(\epsilon)\|\mathbf{Z} - \pi(\mathbf{Z})\|_F. \tag{13}$$

Lemma 3.3 (Proposition 2.4 in [70])**.** *Let $\mathbf{P} \in \mathcal{M}_r$ be given. For each $\epsilon > 0$, there exists $s > 0$ such that for all $\mathbf{C} \in Ball(\mathbf{P}, s) \cap \mathcal{M}_r$, we have:*
(i) $\min_{\mathbf{D}' \in T_{\mathcal{M}_r}(\mathbf{C})} \|\mathbf{D} - \mathbf{D}'\|_F \leq \epsilon \|\mathbf{D} - \mathbf{C}\|_F, \ \forall \ \mathbf{D} \in Ball(\mathbf{P}, s) \cap \mathcal{M}_r$.
(ii) $\|\mathbf{D} - \pi_1(\mathbf{D})\|_F \leq \epsilon \|\mathbf{D} - \mathbf{C}\|_F, \ \forall \ \mathbf{D} \in Ball(\mathbf{P}, s) \cap T_{\mathcal{M}_r}(\mathbf{C})$.

For a point $\mathbf{Z}$ around $\mathbf{P} \in \mathcal{M}_r$, the distance between its projected point on the manifold and the projected point on the tangent space can be estimated as follows. The proof can be found in Appendix.

Lemma 3.4. *Let $P_{T_{\mathcal{M}_r}}(\cdot)$ and $\pi_1(\cdot)$ be given as* (7) *and* (8), *and $\mathbf{P} \in \mathcal{M}_r$. For each $0 < \epsilon < \frac{3}{5}$, there exist an $s(\epsilon) > 0$ and a point $\mathbf{Q} \in Ball(\mathbf{P}, s(\epsilon)) \cap \mathcal{M}_r$ such that for any given $\mathbf{Z} \in Ball(\mathbf{P}, s(\epsilon))$, we have*

$$\|\pi_1(\mathbf{Z}) - P_{T_{\mathcal{M}_r}(\mathbf{Q})}(\mathbf{Z})\|_F < 4\sqrt{\epsilon}\|\mathbf{Z} - \mathbf{Q}\|_F. \tag{14}$$

Lemma 3.5 (Theorem 4.5 in [70])**.** *Suppose $\mathbf{P}$ is a nontangential point with $\sigma(\mathbf{P}) < c$. Then there exists an $s > 0$ such that for all $\mathbf{Z} \in \mathcal{M}_n \cap Ball(\mathbf{P}, s)$, we have*

$$\|\pi_1(\mathbf{Z}) - \pi(\mathbf{Z})\|_F < c\|\mathbf{Z} - \pi(\mathbf{Z})\|_F. \tag{15}$$

Suppose that $\mathbf{Q} \in \mathcal{M}_r \cap Ball(\mathbf{P}, s_1(\epsilon))$ is defined as Lemma 3.4, then the distance between $\pi(\pi_1(P_{T_{\mathcal{M}_r}(\mathbf{Q})}(\mathbf{Z})))$ and $\pi(\mathbf{Z})$ can be estimated as follows. And the proof can be found in Appendix.

Lemma 3.6. *Let $\mathbf{P} \in \mathcal{M}_{rn}$ be given. For each $0 < \epsilon < \frac{3}{5}$, there exist $\varepsilon_1(\epsilon) > 0$, $\varepsilon_2(\epsilon) > 0$ and $s_1(\epsilon) > 0$ such that for all $\mathbf{Z} \in Ball(\mathbf{P}, s_1(\epsilon))$,*

$$\|\pi(\pi_1(P_{T_{\mathcal{M}_r}(\mathbf{Q})}(\mathbf{Z}))) - \pi(\mathbf{Z})\|_F \leq \varepsilon_1(\epsilon)\|\mathbf{Z} - \pi(\mathbf{Z})\|_F + \varepsilon_2(\epsilon)\|\mathbf{Q} - \pi(\mathbf{Z})\|_F.$$

In order to prove the convergence of Algorithm 1, we also need to estimate the distance between $\pi_1(P_{T_{\mathcal{M}_r}(\mathbf{Q})}(\mathbf{Z}))$ and $\pi(\mathbf{Z})$. The proof can be found in Appendix.

Lemma 3.7. *Suppose $\mathbf{P}$ is a nontangential point in $\mathcal{M}_{rn}$ with $\sigma(\mathbf{P}) < c$, and $\mathbf{Q} \in \mathcal{M}_r$. Then there exists an $s > 0$ such that when $\mathbf{Z} = \pi_2(\mathbf{Q}) \in \mathcal{M}_n \cap Ball(\mathbf{P}, s)$ and $P_{T_{\mathcal{M}_r}(\mathbf{Q})}(\mathbf{Z}) \in T_{\mathcal{M}_r}(\mathbf{Q}) \cap Ball(\mathbf{P}, s)$, we have*

$$\|\pi_1(P_{T_{\mathcal{M}_r}(\mathbf{Q})}(\mathbf{Z})) - \pi(\mathbf{Z})\|_F < c\|\mathbf{Z} - \pi(\mathbf{Z})\|_F. \tag{16}$$

With the above tools in hand, we can list the proof of Theorem 3.1 as follows.

Proof of Theorem 3.1 Note that $\mathcal{M}_{rn}$ is a smooth manifold [66] and $\mathbf{P} \in \mathcal{M}_{rn}$, then there exists an s' such that π is

continuous on $Ball(\mathbf{P}, s')$. In other words, we can find a constant $\alpha > 0$ such that

$$\|\pi(\mathbf{X}) - \pi(\mathbf{X}')\|_F \leq \alpha\|\mathbf{X} - \mathbf{X}'\|_F, \forall\ \mathbf{X}, \mathbf{X}' \in Ball(\mathbf{P}, s'). \tag{17}$$

Suppose that $\epsilon < 1$, and set $\sigma(\mathbf{P}) < c < 1$ and

$$\varepsilon = \frac{1-c}{2(3-c)}\epsilon,\ \ \varepsilon_2(\epsilon) = \frac{1-c}{2+2\alpha}\epsilon,$$

where α is a constant given as in (17). It follows Lemma 3.5-3.7 that there exist some possibly distinct radii which can guarantee (15)-(16) are satisfied. Let s denote the minimum of these possibly radii and pick $r < \frac{s(1-\epsilon)}{4(2+\epsilon)}$, so that $\pi(Ball(\mathbf{P}, r)) \subseteq Ball(\mathbf{P}, \frac{s}{4})$. Then $\|\pi(\mathbf{A}) - \mathbf{P}\|_F < \frac{s}{4}$ follows from the latter condition. Denote $l = \|\mathbf{A} - \pi(\mathbf{A})\|_F$ and note that

$$l = \|\mathbf{A}-\mathbf{P}+\mathbf{P}-\pi(\mathbf{A})\|_F \leq \|\mathbf{A}-\mathbf{P}\|_F+\|\mathbf{P}-\pi(\mathbf{A})\|_F \leq r+\frac{s}{4}.$$

As $\pi(\mathbf{A}) \in \mathcal{M}_{rn}$ and note that $\mathbf{X}_1 = \pi_1(\mathbf{A})$, we have

$$\|\mathbf{X}_1 - \mathbf{A}\|_F = \|\pi_1(\mathbf{A}) - \mathbf{A}\|_F \leq \|\pi(\mathbf{A}) - \mathbf{A}\|_F = l$$

and

$$\begin{aligned}\|\mathbf{X}_1 - \pi(\mathbf{X}_1)\|_F &\leq \|\mathbf{X}_1 - \pi(\mathbf{A})\|_F \\ &\leq \|\mathbf{X}_1 - \mathbf{A}\|_F + \|\mathbf{A} - \pi(\mathbf{A})\|_F \leq 2l.\end{aligned}$$

In order to prove $\{\mathbf{X}_k\}$ derived by Algorithm 1 is convergent, we need to prove $\{\mathbf{X}_k\}$ is a Cauchy sequence. By Lemma 3.7, there exist an c_1 such that

$$\begin{aligned}\|\mathbf{X}_{2k+1} - \pi(\mathbf{X}_{2k+1})\|_F &\leq \|\mathbf{X}_{2k+1} - \pi(\mathbf{X}_{2k})\|_F \\ &\leq c_1\|\mathbf{X}_{2k} - \pi(\mathbf{X}_{2k})\|_F. \end{aligned}\tag{18}$$

In addition, by Lemma 3.5, there exist an c_2 such that

$$\begin{aligned}\|\mathbf{X}_{2k} - \pi(\mathbf{X}_{2k})\|_F &\leq \|\mathbf{X}_{2k} - \pi(\mathbf{X}_{2k-1})\|_F \\ &\leq c_2\|\mathbf{X}_{2k-1} - \pi(\mathbf{X}_{2k-1})\|_F. \end{aligned}\tag{19}$$

Set $c = \max\{c_1, c_2\}$, combine (18) and (19) together gives

$$\|\mathbf{X}_k - \pi(\mathbf{X}_k)\|_F \leq c\|\mathbf{X}_{k-1} - \pi(\mathbf{X}_{k-1})\|_F. \tag{20}$$

Then $\{\mathbf{X}_k\}$ is a Cauchy sequence if and only if

$$\{\mathbf{X}_k\}_{k=1}^{\infty} \subseteq Ball(\mathbf{P}, s) \tag{21}$$

is satisfied. The remaining task is to show (21) is satisfied by induction. For $k = 1$,

$$\begin{aligned}\|\mathbf{X}_1 - \mathbf{P}\|_F &\leq \|\mathbf{X}_1 - \mathbf{A}\|_F + \|\mathbf{A} - \mathbf{P}\|_F \leq l + \frac{r}{2} \\ &\leq 2r + \frac{s}{4} \leq \frac{s(1-\epsilon)}{2(2+\epsilon)} + \frac{s}{4} < s.\end{aligned}$$

Assume that (21) is satisfied when $n = k$, then it follows from (20) that

$$\|\mathbf{X}_k - \pi(\mathbf{X}_k)\|_F \leq c^k\|\mathbf{X}_1 - \pi(\mathbf{X}_1)\|_F \leq 2lc^k. \tag{22}$$

For an arbitrary k and $i = 1$ or 2, we have

$$\begin{aligned}&\|\mathbf{X}_{k-2} - \pi(\mathbf{X}_{k-1})\|_F \\ &= \|\mathbf{X}_{k-2} - \pi(\pi_i(\mathbf{X}_{k-2}))\|_F \\ &= \|\mathbf{X}_{k-2} - \pi(\mathbf{X}_{k-2}) + \pi(\mathbf{X}_{k-2}) - \pi(\pi_i(\mathbf{X}_{k-2}))\|_F \\ &\leq \|\mathbf{X}_{k-2} - \pi(\mathbf{X}_{k-2})\|_F + \|\pi(\mathbf{X}_{k-2}) - \pi(\pi_i(\mathbf{X}_{k-2}))\|_F \\ &\leq \|\mathbf{X}_{k-2} - \pi(\mathbf{X}_{k-2})\|_F + \alpha\|\mathbf{X}_{k-2} - \pi_i(X_{k-2})\|_F \\ &\leq (1+\alpha)\|\mathbf{X}_{k-2} - \pi(\mathbf{X}_{k-2})\|_F.\end{aligned}$$

The second part of the second inequality is derived by the continuous of π, the third inequality is derived by

$$\|\mathbf{X}_{k-2} - \pi_i(\mathbf{X}_{k-2})\|_F \leq \|\mathbf{X}_{k-2} - \pi(\mathbf{X}_{k-2})\|_F,\ \ i = 1, 2.$$

In addition, when k is even, by lemma 3.2, we have

$$\|\pi(\mathbf{X}_k) - \pi(\mathbf{X}_{k-1})\|_F < \varepsilon(\epsilon)\|\mathbf{X}_{k-1} - \pi(\mathbf{X}_{k-1})\|_F. \tag{23}$$

When k is odd, applying Lemma 3.6 gives

$$\begin{aligned}&\|\pi(\mathbf{X}_k) - \pi(\mathbf{X}_{k-1})\|_F \\ &< \varepsilon_1(\epsilon)\|\mathbf{X}_{k-1} - \pi(\mathbf{X}_{k-1})\|_F + \varepsilon_2(\epsilon)\|\mathbf{X}_{k-2} - \pi(\mathbf{X}_{k-1})\|_F \\ &< \varepsilon_1(\epsilon)\|\mathbf{X}_{k-1} - \pi(\mathbf{X}_{k-1})\|_F \\ &\quad + \varepsilon_2(\epsilon)(1+\alpha)\|\mathbf{X}_{k-2} - \pi(\mathbf{X}_{k-2})\|_F \\ &\leq 2\varepsilon_1(\epsilon)c^{k-1}l + 2\varepsilon_2(\epsilon)(1+\alpha)c^{k-2}l \\ &= (\varepsilon_1(\epsilon)c + \varepsilon_2(\epsilon)(1+\alpha))2c^{k-2}l.\end{aligned}$$

Set $\varepsilon = \max\{\varepsilon(\epsilon), \varepsilon_1(\epsilon)\}$, then for an arbitrary k, we have

$$\|\pi(\mathbf{X}_k) - \pi(\mathbf{X}_{k-1})\|_F \leq (\varepsilon c + \varepsilon_2(\epsilon)(1+\alpha))2c^{k-2}l. \tag{24}$$

By (24) and Lemma 3.2, we have

$$\begin{aligned}&\|\pi(\mathbf{X}_k) - \pi(\mathbf{A})\|_F \\ &\leq \|\pi(\mathbf{A}) - \pi(\mathbf{X}_1)\|_F + \|\pi(\mathbf{X}_2) - \pi(\mathbf{X}_1)\|_F \\ &\quad + \|\sum_{j=3}^{k}\pi(\mathbf{X}_j) - \pi(\mathbf{X}_{j-1})\|_F \\ &\leq \varepsilon l + 2\varepsilon l + \sum_{j=3}^{k}(\varepsilon_1(\epsilon)c + \varepsilon_2(\epsilon)(1+\alpha))2c^{j-2}l \\ &\leq 3\varepsilon l + \frac{2(\varepsilon_1(\epsilon)c + \varepsilon_2(\epsilon)(1+\alpha))}{1-c}l \\ &= \frac{3\varepsilon(1-c) + 2\varepsilon c + (1+\alpha)\varepsilon_2(\epsilon)}{1-c}l \leq \epsilon l.\end{aligned}\tag{25}$$

Thus,

$$\begin{aligned}\|\mathbf{P} - \mathbf{X}_k\|_F &\leq \|\mathbf{P} - \pi(\mathbf{A})\|_F + \|\pi(\mathbf{A}) - \pi(\mathbf{X}_k)\|_F \\ &\quad + \|\pi(\mathbf{X}_k) - \mathbf{X}_k\|_F \leq s/4 + \epsilon l + 2l < s,\end{aligned}$$

which shows that (21) is satisfied.

It follows from (24) that the sequence $(\pi(\mathbf{X}_k))_{k=1}^{\infty}$ is a Cauchy sequence which converges to a point $\mathbf{Z}_\infty$. Note that (22) is satisfied, the sequence $(\mathbf{X}_k)_{k=1}^{\infty}$ also converges. In addition, $\mathbf{Z}_\infty = \pi(\mathbf{Z}_\infty)$ can be derived by noting that the projection is local continuous. Moreover, by taking the limitation of (25) we can get (i). For (ii). Note that

$$\begin{aligned}\|\pi(\mathbf{X}_k) - \mathbf{X}_\infty\|_F &\leq \sum_{j=k+1}^{\infty}\|\pi(\mathbf{X}_j) - \pi(\mathbf{X}_{j-1})\|_F \\ &\leq \frac{2l\varepsilon c^k}{1-c} + \frac{2(1+\alpha)l\varepsilon_2(\epsilon)c^{k-1}}{1-c},\end{aligned}$$

and combine with (22), we can get

$$\begin{aligned}\|\mathbf{X}_k - \mathbf{X}_\infty\|_F &\leq \|\mathbf{X}_k - \pi(\mathbf{X}_k)\|_F + \|\pi(\mathbf{X}_k) - \mathbf{X}_\infty\|_F \\ &\leq \left(2l + \frac{2l\varepsilon}{1-c} + \frac{2(1+\alpha)l\varepsilon_2(\epsilon)}{1-c}\right)c^k \\ &= \beta c^k l,\end{aligned}$$

with a constant β as desired.

4 Experimental Results

The main aim of this section is to demonstrate that (i) the computational time requried by the proposed TAP method is faster than that by the original alternating projection (AP) method with about the same approximation ability; (ii) the performance of the proposed TAP method is better than that of nonnegative matrix factorization methods in terms of computational time and accuracy for examples in data clustering, pattern recognition and hyperspectral data analysis.

The experiments in Subsection 4.1 are performed under Windows 10 and MATLAB R2020a running on a desktop (Intel Core i7, @ 5.1GHz, 32.00G RAM) and experiments in Subsections 4.2-4.6 are performed under Windows 10 and MATLAB R2020a running on a desktop (AMD Ryzen 9 3950, @ 3.49GHz, 64.00G RAM).

4.1 The First Experiment

The synthetic matrices are of the sizes 200-by-200, 400-by-400 and 800-by-800 and for each size we run nonnegative matrix factorization algorithms (A-MU [24], A-HALS [24], A-PG1 [25], NeNMF [26], and NNSVDLRC [31]) 10 times. In the experiment, we randomly generated n-by-n nonnegative matrices $\mathbf{A}$ where their matrix entries follow a uniform distribution in between 0 and 1. We employed the proposed TAP method and the original alternating projection (AP) method [66] to test the relative approximation error $\|\mathbf{A}-\mathbf{X}_c\|_F/\|\mathbf{A}\|_F$, where $\mathbf{X}_c$ are the computed rank r solutions by different methods. The stopping criteria of each method is that the successive relative approximation error is less than 10^{-5} or the maximum number (MaxIter) of iterations (10^4 or 10^2) is attained. In Tables 1 and 2, the same randomly initial guess is employed in A-MUm A-HALS, A-PG1, NeNMF. In Tables 3 and 4, different randomly initial guesses are employed in A-MUm A-HALS, A-PG1, NeNMF for each trial. However, for NNSVDLRC, which works on generating initial factor matrices, the initial guesses are get from NNSVDLRT and then input into A-HALS [24].

Tables 1-4 shows the relative approximation error of the computed solutions from the proposed TAP method and the other testing methods for synthetic data sets of different sizes. We have the following results.

- For the TAP and AP methods, the non-negative constraint are only added to the low-rank matrix itself, while non-negative constraints are simultaneously added to the two low rank factor matrices. Thus, the relative approximation errors of TAP and AP are always lower than those of NMF methods. These results are confirmed in the tables. Because of tangent space method, the computational time required by the proposed TAP method is less than that required by AP method.
- We find in the tables that the relative approximation errors computed by the TAP method is the same as those by the AP method. It implies that the proposed TAP method can achieve the same accuracy of classical alternating projection.
- NMF algorithms can be sensitive to initial guesses, see Tables 1-4. We illustrate this phenomena by displaying the mean relative approximation error and the range containing both the minimum and the maximum relative approximation errors by ten initial guesses randomly generated. According to the tables, this phenomena is still valid when different (or same) randomly randomly initializations are used in NMF methods in each trial or the maximum number (MaxIter) of iterations is set to be 10^4 or 10^2. However, the computational time required by the TAP method is smaller than those required by NMF methods.

4.2 The Second Experiment

4.2.1 Face Data

In this subsection, we consider two frequently-used face data sets, i.e., the ORL face date set[1] and the extended Yale B face data set [2]. The ORL face data set contains images from 40 individuals, each providing 10 different images with the size 112×92 . In the extended Yale B face data set, we take a subset which consists of 38 people and 64 facial images with different illuminations for each individual. Each testing image is reshaped to a vector, and all the image vectors are combined together to form a nonnegative matrix. Here we perform NMF algorithms and TAP algorithm to obtain low rank approximations with a predefined rank r. There are several NMF algorithms to be compared, namely multiplicative updates (MU) [8], [71], accelerated MU (A-MU) [24], hierarchical alternating least squares (HALS) algorithm [23], accelerated HALS (A-HALS) [24], projected gradient (PG) method [25], accelerated PG (A-PG) [25], NeNMF [26], and NNSVDLRC [31].

Approximation: Firstly, we compare the low rank approximation results by different methods with respect to different predefined ranks r. We report the relative approximation errors in Table 5. For ORL data set, we set r to be 10 and 40 because face data contains 40 individuals and each individual has 10 different images. Similarly, r is set to be 38 and 64 for the extended Yale B data set. In the numerical results, we compare the relative approximation error: $\|\mathbf{X}_c-\mathbf{A}\|_F/\|\mathbf{A}\|_F$. For the TAP and AP methods the nonnegative low rank approximation is directly computed, while for the NMF methods, we multiply the factor matrices. We can see from the table that the relative approximation errors by TAP and AP methods are lower than those by NMF methods.

The relative approximation errors on these two face data sets with respect to different ranks r are plotted in Figure 3. We can see that as r increases, the gap of relative approximation errors between TAP (or AP) method and NMF methods becomes larger. The total computational time required by the proposed TAP method (2.84 seconds) is less than that (17.44 seconds) required by the AP method. The proposed TAP method is more efficient than the AP method.

Recognition: Next, we test the face recognition performance with respect to TAP approximations and NMF approximations. We use the k-fold cross-validation strategy. For each data set, the data is split into k ($k = 10$ for the ORL data set and $k = 64$ for the Yale B data set) groups and each group contains one facial image of each

1. http://www.uk.research.att.com/facedatabase.html
2. http://vision.ucsd.edu/∼leekc/ExtYaleDatabase/ExtYaleB.html

TABLE 1
The relative approximation error and computation time on the synthetic data sets with MaxIter = 10^4. The **best** values are respectively highlighted by bolder fonts. Here the same randomly initialization is used for NMF methods in each trial.

	200-by-200 matrix					
	Relative approximation error			Computation time (s)		
Method	$r=10$	$r=20$	$r=40$	$r=10$	$r=20$	$r=40$
TAP	**0.4574**	**0.4158**	**0.3426**	**0.02**	**0.01**	**0.02**
AP	**0.4574**	**0.4158**	**0.3426**	0.03	0.02	0.03
A-MU: mean	0.4588	0.4244	0.3717	15.18	8.82	8.97
A-MU: range	[0.4588,0.4589]	[0.4242,0.4246]	[0.3713,0.3720]	[14.55,15.44]	[8.72,9.00]	[8.78,9.14]
HALS: mean	0.4588	0.4243	0.3710	16.29	16.35	16.26
HALS: range	[0.4588,0.4589]	[0.4242,0.4245]	[0.3707,0.3712]	[16.05,16.72]	[16.16,16.48]	[16.05,16.50]
A-PG1: mean	0.4588	0.4243	0.3711	15.40	9.46	9.64
A-PG1: range	[0.4588,0.4589]	[0.4242,0.4244]	[0.3708,0.3714]	[15.19,15.54]	[9.16,10.08]	[9.55,9.72]
NeNMF: mean	0.4588	0.4245	0.3723	0.51	0.59	0.72
NeNMF: range	[0.4588,0.4589]	[0.4243,0.4247]	[0.3716,0.3728]	[0.45,0.92]	[0.45,0.77]	[0.51,0.91]
NNSVDLRC: mean	0.4588	0.4243	0.3712	21.19	20.33	19.07
NNSVDLRC: range	[0.4588,0.4588]	[0.4243,0.4243]	[0.3711,0.3712]	[20.77,21.81]	[19.91,21.30]	[17.98,19.58]
	400-by-400 matrix					
	Relative approximation error			Computation time (s)		
Method	$r=20$	$r=40$	$r=80$	$r=20$	$r=40$	$r=80$
TAP	**0.4560**	**0.4153**	**0.3419**	**0.03**	**0.03**	**0.06**
AP	**0.4560**	**0.4153**	**0.3419**	0.04	0.05	0.13
A-MU:mean	0.4593	0.4288	0.3840	8.77	9.02	9.42
A-MU:range	[0.4592,0.4593]	[0.4287,0.4290]	[0.3838,0.3844]	[8.62,8.86]	[8.91,9.14]	[9.33,9.52]
HALS:mean	0.4592	0.4283	0.3823	16.11	15.69	15.89
HALS:range	[0.4591,0.4592]	[0.4282,0.4284]	[0.3822,0.3825]	[15.96,16.34]	[15.39,16.12]	[15.73,16.09]
A-PG1:mean	0.4592	0.4286	0.3836	9.05	9.17	10.00
A-PG1:range	[0.4592,0.4593]	[0.4285,0.4287]	[0.3834,0.3838]	[8.85,9.15]	[9.06,9.31]	[9.74,10.20]
NeNMF:mean	0.4593	0.4291	0.3856	0.74	0.89	0.92
NeNMF:range	[0.4593,0.4594]	[0.4289,0.4293]	[0.3852,0.3859]	[0.56,0.99]	[0.71,1.20]	[0.84,1.12]
NNSVDLRC:mean	0.4592	0.4283	0.3822	16.62	15.83	15.83
NNSVDLRC:range	[0.4591,0.4592]	[0.4282,0.4284]	[0.3820,0.3824]	[15.97,18.55]	[15.31,17.03]	[15.76,15.99]
	800-by-800 matrix					
	Relative approximation error			Computation time (s)		
Method	$r=40$	$r=80$	$r=160$	$r=40$	$r=80$	$r=160$
TAP	**0.4551**	**0.4145**	**0.3411**	**0.14**	**0.16**	**0.33**
AP	**0.4551**	**0.4145**	**0.3411**	0.20	0.28	0.68
A-MU:mean	0.4607	0.4346	0.3977	9.12	9.44	11.32
A-MU:range	[0.4606,0.4607]	[0.4345,0.4347]	[0.3976,0.3978]	[9.05,9.19]	[9.28,9.76]	[11.08,11.51]
HALS:mean	0.4603	0.4334	0.3945	14.58	15.03	15.20
HALS:range	[0.4603,0.4604]	[0.4334,0.4335]	[0.3944,0.3946]	[14.29,14.90]	[14.87,15.17]	[15.00,15.39]
A-PG1:mean	0.4605	0.4339	0.3980	9.58	10.41	11.16
A-PG1:range	[0.4605,0.4606]	[0.4339,0.4340]	[0.3977,0.3985]	[9.44,9.78]	[10.25,10.54]	[10.99,11.38]
NeNMF:mean	0.4609	0.4356	0.3996	1.08	1.30	2.51
NeNMF:range	[0.4609,0.4610]	[0.4354,0.4357]	[0.3993,0.3997]	[0.93,1.18]	[1.11,1.40]	[2.41,2.64]
NNSVDLRC:mean	0.4603	0.4334	0.3946	14.83	15.09	15.30
NNSVDLRC:range	[0.4603,0.4604]	[0.4334,0.4335]	[0.3945,0.3947]	[14.43,15.06]	[14.94,15.25]	[15.09,15.50]

individual. For instance, the ORL data set is split into $k = 10$ groups and each group contains 40 facial images. Then, we circularly take one group as a test data set and the remaining groups as a training data set until all the groups have been selected as the test data. Given the original training data $\mathbf{A}_{\text{train}}$ with the size $m \times n$, where n indicates the pixels of each face image and m is the amount of training samples, we first perform NMF and TAP (or AP) algorithms to obtain non-negative low rank approximations $\mathbf{A}_{\text{train}} \approx \mathbf{B}_{\text{NMFtran}}\mathbf{C}_{\text{NMFtrain}}$ and $\mathbf{A}_{\text{train}} \approx \mathbf{U}_{\text{TAPtrain}}\mathbf{\Sigma}_{\text{TAPtrain}}\mathbf{V}_{\text{TAPtrain}}$ respectively with rank r. The new representations of $\mathbf{A}_{\text{train}}$ are given by $\mathbf{U}^T_{\text{NMFtrain}}\mathbf{A}_{\text{train}}$ and $\mathbf{U}^T_{\text{TAPtrain}}\mathbf{A}_{\text{train}}$ respectively by the NMF methods and the TAP (or AP) method. The nearest neighbor (NN) classifier is adopted by recognized the testing data based on the distance between their representations and the projected training data.

The face recognition results are exhibited in Table 6. From this table, we can see that the accuracies based on TAP approximations are higher than those based on NMF approximations. To further investigate how the rank r affects the recognition results, we plot the recognition accuracy on the extended Yale B data set with respect to r in Figure 4. It can be found that the recognition accuracy based on TAP and AP approximations is always better than those based on NMF approximations. Meanwhile, to see the features learned by different methods, we exhibit

TABLE 2
The relative approximation error and computation time on the synthetic data sets with MaxIter = 10^2. The **best** values are respectively highlighted by bolder fonts. Here the same randomly initialization is used for NMF methods in each trial.

	200-by-200 matrix					
	Relative approximation error			Computation time (s)		
Method	$r = 10$	$r = 20$	$r = 40$	$r = 10$	$r = 20$	$r = 40$
TAP	**0.4574**	**0.4158**	**0.3426**	**0.01**	**0.01**	**0.01**
AP	**0.4574**	**0.4158**	**0.3426**	**0.01**	**0.01**	0.03
A-MU:mean	0.4593	0.4262	0.3766	0.08	0.11	0.11
A-MU:range	[0.4591,0.4594]	[0.4259,0.4264]	[0.3760,0.3771]	[0.07,0.09]	[0.10,0.12]	[0.11,0.12]
HALS:mean	0.4596	0.4258	0.3742	0.02	0.03	0.05
HALS:range	[0.4593,0.4599]	[0.4253,0.4260]	[0.3738,0.3746]	[0.01,0.04]	[0.02,0.03]	[0.04,0.05]
A-PG1:mean	0.4590	0.4252	0.3746	0.11	0.16	0.20
A-PG1:range	[0.4589,0.4592]	[0.4249,0.4254]	[0.3742,0.3751]	[0.09,0.19]	[0.15,0.18]	[0.20,0.22]
NeNMF:mean	0.4591	0.4251	0.3735	0.07	0.14	0.27
NeNMF:range	[0.4590,0.4592]	[0.4249,0.4254]	[0.3732,0.3738]	[0.05,0.12]	[0.11,0.18]	[0.24,0.30]
NNSVDLRC:mean	0.4592	0.4255	0.3734	0.05	0.04	0.06
NNSVDLRC:range	[0.4592,0.4592]	[0.4254,0.4255]	[0.3734,0.3734]	[0.03,0.14]	[0.03,0.04]	[0.06,0.07]
	400-by-400 matrix					
	Relative approximation error			Computation time (s)		
Method	$r = 20$	$r = 40$	$r = 80$	$r = 20$	$r = 40$	$r = 80$
TAP	**0.4560**	**0.4153**	**0.3419**	**0.02**	**0.03**	**0.06**
AP	**0.4560**	**0.4153**	**0.3419**	0.03	0.05	0.12
A-MU:mean	0.4600	0.4309	0.3887	0.31	0.30	0.33
A-MU:range	[0.4598,0.4601]	[0.4307,0.4311]	[0.3883,0.3890]	[0.30,0.34]	[0.29,0.31]	[0.31,0.35]
HALS:mean	0.4601	0.4300	0.3853	0.05	0.10	0.23
HALS:range	[0.4600,0.4602]	[0.4298,0.4302]	[0.3852,0.3856]	[0.05,0.06]	[0.09,0.10]	[0.23,0.25]
A-PG1:mean	0.4598	0.4306	0.3893	0.50	0.52	0.69
A-PG1:range	[0.4597,0.4599]	[0.4303,0.4307]	[0.3890,0.3897]	[0.49,0.56]	[0.51,0.53]	[0.66,0.72]
NeNMF:mean	0.4596	0.4296	0.3856	0.22	0.42	0.72
NeNMF:range	[0.4595,0.4597]	[0.4295,0.4298]	[0.3852,0.3859]	[0.19,0.30]	[0.38,0.44]	[0.65,0.85]
NNSVDLRC:mean	0.4599	0.4298	0.3852	0.07	0.13	0.29
NNSVDLRC:range	[0.4599,0.4599]	[0.4298,0.4298]	[0.3851,0.3853]	[0.07,0.08]	[0.12,0.13]	[0.28,0.33]
	800-by-800 matrix					
	Relative approximation error			Computation time (s)		
Method	$r = 40$	$r = 80$	$r = 160$	$r = 40$	$r = 80$	$r = 160$
TAP	**0.4551**	**0.4145**	**0.3411**	**0.11**	**0.13**	**0.26**
AP	**0.4551**	**0.4145**	**0.3411**	0.15	0.22	0.53
A-MU:mean	0.4614	0.4364	0.4010	0.83	0.95	1.50
A-MU:range	[0.4614,0.4615]	[0.4363,0.4364]	[0.4008,0.4012]	[0.80,0.89]	[0.94,0.99]	[1.48,1.55]
HALS:mean	0.4614	0.4350	0.3970	0.23	0.45	0.98
HALS:range	[0.4613,0.4615]	[0.4349,0.4352]	[0.3969,0.3971]	[0.22,0.23]	[0.44,0.46]	[0.97,1.00]
A-PG1:mean	0.4610	0.4352	0.4047	1.60	1.68	2.47
A-PG1:range	[0.4610,0.4611]	[0.4351,0.4352]	[0.4044,0.4049]	[1.57,1.65]	[1.66,1.71]	[2.45,2.50]
NeNMF:mean	0.4610	0.4356	0.3996	0.77	1.03	2.02
NeNMF:range	[0.4609,0.4610]	[0.4354,0.4357]	[0.3993,0.3997]	[0.70,0.81]	[0.91,1.09]	[1.92,2.13]
NNSVDLRC:mean	0.4610	0.4347	0.3968	0.35	0.64	1.31
NNSVDLRC:range	[0.4610,0.4611]	[0.4347,0.4348]	[0.3968,0.3969]	[0.31,0.42]	[0.58,0.69]	[1.24,1.38]

the column vectors of $\mathbf{B}_{\text{NMFtrain}}$ and singular vectors of $\mathbf{U}_{\text{TAPtrain}}$ in Figure 5. These vectors are reshaped to the same size as facial images and their values are normalized to [0,255] for the display purpose. We see that the nonnegative low rank matrix approximation methods do not give the part-based representations, but provides different important facial representations in the recognition.

4.2.2 Document Data

In this subsection, we use the NIST Topic Detection and Tracking (TDT2) corpus as the document data. The TDT2 corpus consists of data collected during the first half of 1998 and taken from 6 sources, including 2 newswires (APW, NYT), 2 radio programs (VOA, PRI) and 2 television programs (CNN, ABC). It consists of 11201 on-topic documents which are classified into 96 semantic categories. In this experiment, the documents appearing in two or more categories were removed, and only the largest 30 categories were kept, thus leaving us with 9394 documents in total. Then, each document is represented by the weighted term-frequency vector [16], and all the documents are gathered as a matrix $\mathbf{A}_{\text{doc}}$ of size 9394×36771. By using the procedure given in [16], we compute the projected results $\mathbf{U}_{\text{TAP}}^T\mathbf{A}_{\text{TAP}} = \mathbf{\Sigma}_{\text{TAP}}\mathbf{V}_{\text{TAP}}^T$, and then use k-means clustering method and Kuhn-Munkres algorithm to find the best mapping which maps each cluster label to the equivalent

TABLE 3
The relative approximation error and computation time on the synthetic data sets with MaxIter = 10^4. The **best** values are respectively highlighted by bolder fonts. Here different randomly initializations are used in NMF methods in each trial.

	200-by-200 matrix					
	Relative approximation error			Computation time (s)		
Method	$r=10$	$r=20$	$r=40$	$r=10$	$r=20$	$r=40$
TAP	**0.4574**	**0.4158**	**0.3426**	**0.01**	**0.01**	**0.01**
AP	**0.4574**	**0.4158**	**0.3426**	**0.01**	**0.01**	0.02
A-MU:mean	0.4588	0.4245	0.3721	3.85	4.14	4.20
A-MU:range	[0.4588,0.4589]	[0.4243,0.4247]	[0.3718,0.3723]	[3.71,3.93]	[4.08,4.22]	[4.14,4.26]
HALS:mean	0.4588	0.4243	0.3711	0.79	1.20	2.25
HALS:range	[0.4588,0.4589]	[0.4241,0.4244]	[0.3706,0.3713]	[0.76,0.82]	[1.18,1.23]	[2.17,2.32]
A-PG1:mean	0.4588	0.4243	0.3713	4.69	4.53	4.83
A-PG1:range	[0.4588,0.4589]	[0.4242,0.4245]	[0.3710,0.3717]	[4.61,4.77]	[4.40,4.68]	[4.65,4.96]
NeNMF:mean	0.4588	0.4246	0.3723	0.48	0.51	0.56
NeNMF:range	[0.4588,0.4590]	[0.4244,0.4249]	[0.3718,0.3728]	[0.45,0.69]	[0.45,0.71]	[0.45,0.72]
NNSVDLRC:mean	0.4588	0.4243	0.3710	0.80	1.22	2.26
NNSVDLRC:range	[0.4588,0.4589]	[0.4243,0.4245]	[0.3709,0.3713]	[0.79,0.81]	[1.18,1.25]	[2.18,2.35]
	400-by-400 matrix					
	Relative approximation error			Computation time (s)		
Method	$r=20$	$r=40$	$r=80$	$r=20$	$r=40$	$r=80$
TAP	**0.4560**	**0.4153**	**0.3419**	**0.02**	**0.03**	**0.06**
AP	**0.4560**	**0.4153**	**0.3419**	0.03	0.05	0.12
A-MU:mean	0.4593	0.4291	0.3846	4.15	4.24	4.43
A-MU:range	[0.4593,0.4594]	[0.4290,0.4292]	[0.3845,0.3847]	[4.09,4.24]	[4.17,4.30]	[4.37,4.49]
HALS:mean	0.4592	0.4284	0.3823	2.65	4.95	7.21
HALS:range	[0.4591,0.4592]	[0.4283,0.4285]	[0.3821,0.3825]	[2.61,2.67]	[4.89,5.09]	[6.93,7.42]
A-PG1:mean	0.4593	0.4288	0.3845	4.54	4.31	4.51
A-PG1:range	[0.4592,0.4593]	[0.4287,0.4291]	[0.3843,0.3847]	[4.44,4.81]	[4.19,4.43]	[4.41,4.58]
NeNMF:mean	0.4593	0.4291	0.3855	0.64	0.71	0.78
NeNMF:range	[0.4593,0.4594]	[0.4289,0.4294]	[0.3851,0.3857]	[0.54,0.79]	[0.56,0.85]	[0.70,0.89]
NNSVDLRC:mean	0.4592	0.4283	0.3825	2.74	5.18	7.30
NNSVDLRC:range	[0.4592,0.4592]	[0.4283,0.4284]	[0.3823,0.3826]	[2.70,2.81]	[5.11,5.25]	[7.07,7.42]
	800-by-800 matrix					
	Relative approximation error			Computation time (s)		
Method	$r=40$	$r=80$	$r=160$	$r=40$	$r=80$	$r=160$
TAP	**0.4551**	**0.4145**	**0.3411**	**0.10**	**0.13**	**0.27**
AP	**0.4551**	**0.4145**	**0.3411**	0.15	0.22	0.54
A-MU:mean	0.4608	0.4350	0.3984	4.29	4.47	5.60
A-MU:range	[0.4608,0.4609]	[0.4348,0.4351]	[0.3983,0.3986]	[4.20,4.36]	[4.36,4.68]	[5.39,6.28]
HALS:mean	0.4603	0.4335	0.3948	7.17	7.45	7.62
HALS:range	[0.4603,0.4604]	[0.4334,0.4335]	[0.3947,0.3948]	[7.03,7.38]	[7.24,7.57]	[7.50,7.76]
A-PG1:mean	0.4606	0.4342	0.4001	4.43	4.86	5.63
A-PG1:range	[0.4606,0.4607]	[0.4342,0.4343]	[0.3999,0.4004]	[4.29,4.56]	[4.76,4.99]	[5.43,5.82]
NeNMF:mean	0.4609	0.4354	0.3995	0.85	1.09	2.05
NeNMF:range	[0.4608,0.4610]	[0.4352,0.4356]	[0.3991,0.3997]	[0.74,0.96]	[1.00,1.18]	[1.99,2.13]
NNSVDLRC:mean	0.4604	0.4335	0.3949	7.23	7.53	7.70
NNSVDLRC:range	[0.4604,0.4604]	[0.4335,0.4336]	[0.3947,0.3950]	[7.11,7.40]	[7.41,7.63]	[7.62,7.85]

label from the document corpus. For NMF methods, we scale each column of $\mathbf{B}_{\text{NMF}}$ such that their ℓ_2 norms are equal to 1, and the corresponding scaled $\mathbf{C}_{\text{NMF}}$ is used for clustering and label assignment. To quantitatively evaluate the clustering performance of each method, we selected two metrics, i.e., the accuracy and the normalized mutual information (NMI) (we refer to [46] for detailed discussion). According to Table (7), it is clear that nonnegative low rank matrix approximation can provide more effective latent features ($\mathbf{U}_{\text{TAP}}^T\mathbf{A}_{\text{TAP}} = \mathbf{\Sigma}_{\text{TAP}}\mathbf{V}_{\text{TAP}}^T$) for document clustering task. Note that the computational time required by the proposed TAP method (309.22 seconds) is less than that (3417.33 seconds) required by the AP method. Again the results demonstrate that the proposed TAP method is more efficient than the AP method.

4.3 Separable Nonnegative Matrices

In this subsection, we compare the performance of the nonnegative low rank matrix approximation method and separable NMF algorithms. Here we generate two kinds of synthetic separable nonnegative matrices.

- (Separable) The first case $\mathbf{A} = \mathbf{BC} + \mathbf{N}$ is generated the same as [40], in which $\mathbf{B} \in \mathbb{R}^{200\times 20}$ is *uniform distributed* and $\mathbf{C} = [\mathbf{I}_{20}, \mathbf{H}'] \in \mathbb{R}^{20\times 210}$ with $\mathbf{H}'$

TABLE 4
The relative approximation error and computation time on the synthetic data sets with MaxIter = 10^2. The **best** values are respectively highlighted by bolder fonts. Here different randomly initializations are used in NMF methods in each trial.

	200-by-200 matrix					
	Relative approximation error			Computation time (s)		
Method	$r=10$	$r=20$	$r=40$	$r=10$	$r=20$	$r=40$
TAP	**0.4574**	**0.4158**	**0.3426**	**0.01**	**0.01**	**0.01**
AP	**0.4574**	**0.4158**	**0.3426**	**0.01**	**0.01**	0.02
A-MU:mean	0.4593	0.4261	0.3765	0.08	0.11	0.12
A-MU:range	[0.4591,0.4596]	[0.4258,0.4265]	[0.3761,0.3768]	[0.07,0.09]	[0.11,0.12]	[0.12,0.12]
HALS:mean	0.4596	0.4258	0.3743	0.02	0.03	0.05
HALS:range	[0.4594,0.4600]	[0.4255,0.4260]	[0.3739,0.3748]	[0.01,0.02]	[0.02,0.03]	[0.04,0.05]
A-PG1:mean	0.4591	0.4252	0.3745	0.10	0.16	0.21
A-PG1:range	[0.4589,0.4593]	[0.4248,0.4255]	[0.3739,0.3755]	[0.09,0.11]	[0.16,0.17]	[0.21,0.23]
NeNMF:mean	0.4591	0.4251	0.3734	0.05	0.15	0.28
NeNMF:range	[0.4590,0.4593]	[0.4250,0.4254]	[0.3729,0.3740]	[0.05,0.06]	[0.12,0.17]	[0.24,0.30]
NNSVDLRC:mean	0.4592	0.4255	0.3734	0.03	0.04	0.06
NNSVDLRC:range	[0.4592,0.4592]	[0.4255,0.4255]	[0.3734,0.3735]	[0.02,0.03]	[0.03,0.04]	[0.06,0.06]
	400-by-400 matrix					
	Relative approximation error			Computation time (s)		
Method	$r=20$	$r=40$	$r=80$	$r=20$	$r=40$	$r=80$
TAP	**0.4560**	**0.4153**	**0.3419**	**0.02**	**0.03**	**0.06**
AP	**0.4560**	**0.4153**	**0.3419**	0.04	0.05	0.12
A-MU:mean	0.4600	0.4309	0.3887	0.32	0.32	0.35
A-MU:range	[0.4599,0.4603]	[0.4306,0.4310]	[0.3883,0.3892]	[0.31,0.32]	[0.31,0.33]	[0.34,0.40]
HALS:mean	0.4602	0.4301	0.3854	0.06	0.10	0.24
HALS:range	[0.4601,0.4603]	[0.4298,0.4303]	[0.3852,0.3856]	[0.05,0.06]	[0.10,0.11]	[0.23,0.26]
A-PG1:mean	0.4598	0.4307	0.3892	0.51	0.55	0.72
A-PG1:range	[0.4597,0.4598]	[0.4304,0.4309]	[0.3889,0.3894]	[0.50,0.54]	[0.54,0.56]	[0.70,0.74]
NeNMF:mean	0.4596	0.4295	0.3855	0.24	0.46	0.78
NeNMF:range	[0.4595,0.4597]	[0.4293,0.4297]	[0.3851,0.3857]	[0.21,0.28]	[0.44,0.47]	[0.71,0.90]
NNSVDLRC:mean	0.4599	0.4298	0.3852	0.08	0.14	0.29
NNSVDLRC:range	[0.4599,0.4599]	[0.4297,0.4298]	[0.3851,0.3852]	[0.07,0.08]	[0.13,0.14]	[0.26,0.31]
	800-by-800 matrix					
	Relative approximation error			Computation time (s)		
Method	$r=40$	$r=80$	$r=160$	$r=40$	$r=80$	$r=160$
TAP	**0.4551**	**0.4145**	**0.3411**	**0.10**	**0.13**	**0.27**
AP	**0.4551**	**0.4145**	**0.3411**	0.15	0.22	0.53
A-MU:mean	0.4615	0.4363	0.4009	0.83	0.96	1.50
A-MU:range	[0.4614,0.4615]	[0.4362,0.4365]	[0.4008,0.4011]	[0.82,0.84]	[0.95,0.97]	[1.49,1.52]
HALS:mean	0.4614	0.4350	0.3970	0.23	0.45	0.98
HALS:range	[0.4613,0.4615]	[0.4349,0.4351]	[0.3968,0.3973]	[0.22,0.23]	[0.44,0.47]	[0.96,1.00]
A-PG1:mean	0.4610	0.4351	0.4047	1.61	1.71	2.50
A-PG1:range	[0.4610,0.4611]	[0.4350,0.4352]	[0.4045,0.4050]	[1.59,1.64]	[1.69,1.75]	[2.48,2.53]
NeNMF:mean	0.4609	0.4354	0.3995	0.79	1.09	2.04
NeNMF:range	[0.4609,0.4610]	[0.4352,0.4356]	[0.3991,0.3997]	[0.76,0.82]	[1.00,1.18]	[1.99,2.13]
NNSVDLRC:mean	0.4610	0.4347	0.3968	0.36	0.67	1.36
NNSVDLRC:range	[0.4610,0.4611]	[0.4347,0.4348]	[0.3967,0.3969]	[0.30,0.38]	[0.63,0.78]	[1.25,1.47]

TABLE 5
The relative approximation error on the Yale-B data set and the ORL data set. The **best** values and the second best values are respectively highlighted by bolder fonts and underlines.

Dataset	r	MU	A-MU	HALS	A-HALS	PG	A-PG	Ne-NMF	NNSV-DLRC	AP	TAP
Extended Yale B	38	0.186	0.182	0.181	0.182	0.187	0.184	0.182	0.181	**0.164**	**0.164**
	64	0.160	0.157	0.152	0.152	0.159	0.159	0.159	0.151	**0.131**	**0.131**
ORL	10	0.206	0.206	0.205	0.205	0.206	0.206	0.205	0.205	**0.204**	**0.204**
	40	0.159	0.156	0.155	0.155	0.160	0.158	0.154	0.154	**0.147**	**0.147**

containing all possible combinations of two non-zero entries equal to 0.5 at different positions. The columns of $\mathbf{BH}'$ are all the middle points of the columns of $\mathbf{B}$. Meanwhile, the i-th column of $\mathbf{N}$, denoted as n_i, obeys $n_i = \sigma(m_i - \bar{w})$ for $21 \leq i \leq 210$, where $\sigma > 0$ is the noise level, m_i is the i-th column of $\mathbf{B}$, and $\bar{w}$ denotes the average of columns of $\mathbf{B}$. This means that we move the columns of $\mathbf{A}$ toward the outside of the convex hull of the columns of $\mathbf{B}$.

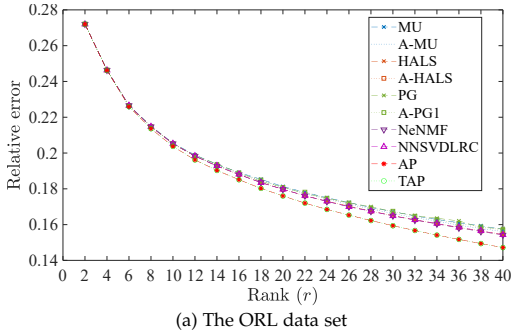
- (Generalized separable) The second case is generated almost the same as the first case but simultaneously considering the separability of rows, known as generalized separable NMF [42]. For this case, the size of

TABLE 6
The recognition accuracy on the Yale-B data set and the ORL data set. The **best** values and the second best values are respectively highlighted by bolder fonts and underlines.

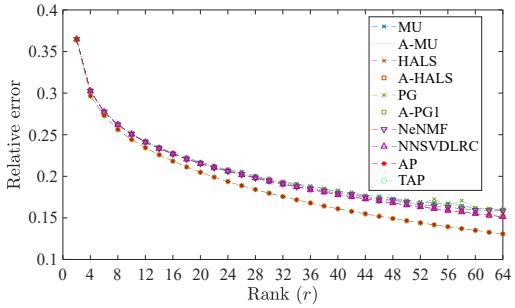
Dataset	Parameter	MU	A-MU	HALS	A-HALS	PG	A-PG	NeNMF	NNSVDLRC	AP	TAP
Yale B	$r = 38$	61.061%	61.143%	61.637%	62.253%	58.306%	60.074%	61.102%	62.130%	**66.776%**	67.681%
	$r = 64$	69.942%	70.477%	72.821%	72.821%	65.502%	68.586%	69.572%	72.656%	76.563%	**76.809%**
ORL	$r = 10$	95.750%	96.250%	96.250%	96.250%	96.500%	96.500%	95.750%	96.000%	**96.750%**	**96.750%**
	$r = 40$	98.250%	98.000%	98.250%	**98.500%**	79.250%	98.250%	97.750%	**98.500%**	**98.500%**	**98.500%**

TABLE 7
The accuracy and NMI values of the document clustering results on the TDT2 data set.

Metric	MU	A-MU	HALS	A-HALS	PG	A-PG	NeNMF	NNSVDLRC	AP	TAP
Accuracy	52.800%	50.724%	54.322%	53.108%	54.205%	51.661%	54.68%	47.23%	61.294%	**61.326%**
NMI	0.674	0.651	0.663	0.643	0.681	0.661	0.693	0.667	**0.728**	**0.728**



(a) The ORL data set



(b) The extended Yale B data set

Fig. 3. Relative approximation errors on the ORL data set (a) and the extended Yale B data set (b), with respect to the different ranks r.

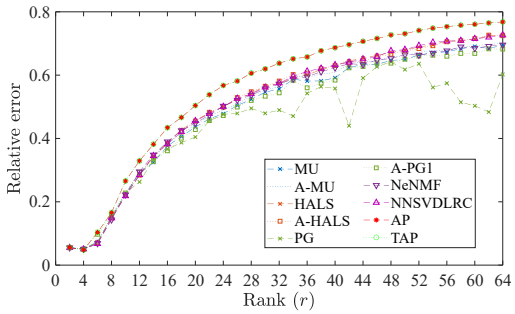


Fig. 4. The recognition accuracy (%) on the extended Yale-B data set with respect to rank r.

$\mathbf{A}$ is set as 78×55 with column-rank 10 and row-rank 12, being the same as [42].

Firstly, we test the approximation ability of TAP and AP methods, NMF methods, and the successive projection

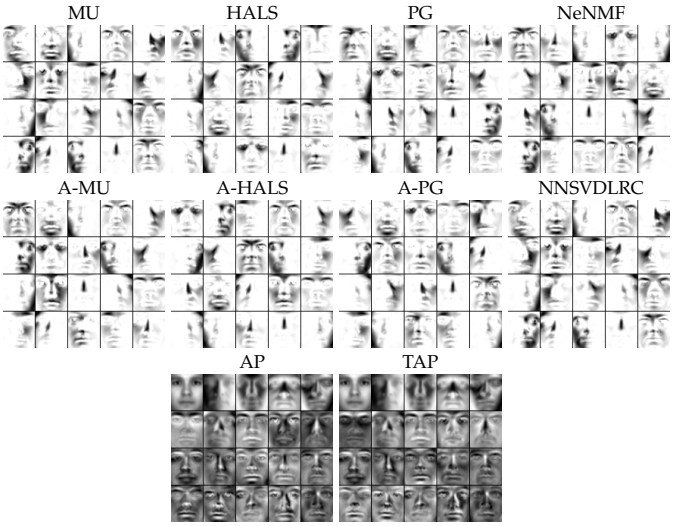


Fig. 5. The first 20 singular vectors of the results by the TAP (or AP) method and the columns of left factor matrices resulted by NMF methods when the rank $r = 20$. These vectors are reshaped to the size of facial images and their values are adaptively normalized.

algorithm (SPA) [40], [41] for separable NMF for synthetic separable data. For the generalized separable case, we compare the TAP (or AP) method with SPA, the generalized SPA (GSPA) [42], and the generalized separable NMF with a fast gradient method (GS-FGM) [42]. Note that when we apply SPA on the generalized separable matrix, we run it firstly to identify the important columns and with the transpose of the input to identify the important rows. This variant is referred to SPA*. The noise level σ is logarithmic spaced in the interval $[10^{-3}, 1]$. For each noise level, we independently generate 25 matrices for both separable and generalized separable cases, respectively. We report the averaged approximation error in Figures 6 and 7. It can be found that TAP and AP methods can achieve the lowest average errors in the testing examples.

The approximation errors of TAP and AP methods are much lower than separable and generalized separable NMF methods when the noise level is high. Note that the average computational time required by the proposed TAP method (0.0064 seconds) is less than that (0.0165 seconds) required by the AP method. It is interesting whether a better non-

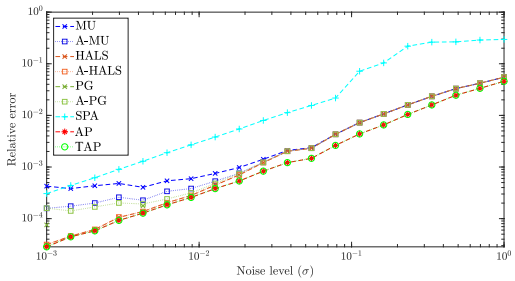


Fig. 6. Average relative approximation error on separable matrices (Case 1), with respect to the different values of σ.

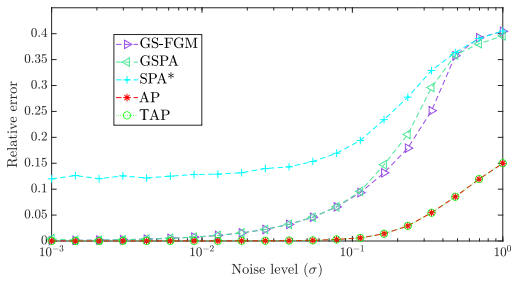


Fig. 7. Average relative approximation error on generalized separable matrices (Case 2), with respect to the different values of σ.

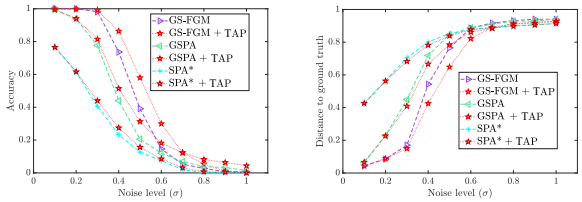


Fig. 8. Average accuracy (left) and distance to ground truth (right) for the different algorithms on generalized separable matrices (Case 2), with respect to the different σs.

negative low rank matrix approximation could contribute to a better separable (or generalized separable) NMF result. To further investigate whether nonnegative low rank matrix approximation could help separable and generalized separable NMF methods, we conduct the experiments with inputting the nonnegative low rank approximation to separable and generalized separable NMF methods. We adopt the accuracy and the distance to ground truth defined in Eqs. (16) and (17) of [42] as the quantitative metrics. The accuracy reports the proportion of correctly identified row and column indices while the distance to ground truth reports the relative errors between the identified important rows (columns) to the ground truth important rows (columns). We present the computational results in Figure 8. When the noise level is between 0.1 and 1, the nonnegative low rank matrix approximation by our TAP method obviously enhances the accuracy and decrease the distance between the identified rows (columns) to the ground truth.

4.4 Symmetric Nonnegative Matrices for Graph Clustering

In this subsection, we test TAP and AP methods on the symmetric matrices. It can readily be found that the output of TAP and AP algorithms would be symmetric if the input matrix is symmetric since that the projection onto the nonnegative matrix manifold or the low rank matrix manifold would never affect the symmetry. Here symmetric NMF methods are the coordinate descent algorithm (denoted as "CD-symNMF ") [38], the Newton-like algorithm (denoted as "Newton-symNMF") [37], and the alternating least squares algorithm (denoted as "ALS-symNMF") [37].

We perform experiments by using symmetric NMF methods, TAP and AP methods on the synthetic graph data, which is reproduced from [72] with six different cases. The data points in 2-dimensional space are displayed in the first row of Figure 9. Each case contains clear cluster structures. By following the procedures in [37], [72], a similarity matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, where n represents the number of data points, is constructed to characterize the similarity between each pair of data points. Each data point is assumed to be only connected to its nearest nine neighbors. Given a specific pair of the i-th and j-th data points x_i and x_j, we firstly construct the distance matrix $\mathbf{D} \in \mathbb{R}^{n\times n}$ with $D_{ij} = D_{ji} = \|x_i - x_j\|_2^2$. Then, the similarity matrix is given as

$$A_{ij} = \begin{cases} 0, \text{ if } i = j, \\ e^{(\frac{-D_{ij}}{\sigma_i \sigma_j})}, \text{ if } i \neq j, \end{cases} \tag{26}$$

where σ_i is the Euclidean distance between the i-th data point x_i and its 9-th neighbor. Then, we perform NMF, TAP and AP methods for $\mathbf{A}$.

The clustering results of the symmetric NMF methods and nonnegative low rank matrix approximation are obtained by using the k-means method on $\mathbf{B}$ and $\mathbf{U}$ respectively. The clustering results are shown in Figure 9. CD-symNMF method fails in most examples except the example in the second column. Both Newton-symNMF and ALS-symNMF methods fail in the example in the fifth column. However, TAP and AP methods perform well for all the examples. The average computational time required by the proposed TAP method (0.0321 seconds) is less than that (0.1035 seconds) required by the AP method. The proposed TAP is faster than the AP method.

4.5 Orthogonal Decomposable Non-negative Matrices

In this subsection, we test TAP and AP methods and orthogonal NMF (ONMF) methods [4], [35] on the approximation of the synthetic data and the unmixing of hyperspectral images. The orthogonal NMF method is a multiplicative updating algorithm proposed by Ding et al. [4]. We refer to Ding-Ti-Peng-Park (DTPP)-ONMF. A multiplicative updating algorithm utilizing the true gradient in Stiefel manifold is proposed in [35]. We refer to SM-ONMF.

We construct an orthogonal nonnegative matrix $\mathbf{B} \in \mathbb{R}^{100\times 10}$, whose transpose is shown in Figure 10. Then a matrix $\mathbf{C} \in \mathbb{R}^{10\times 30}$ is generated with entries uniformly distributed in $[0, 1]$. Then, we obtain an orthogonal decomposable matrix $\mathbf{A} = \mathbf{BC} \in \mathbb{R}^{100\times 30}$. Next, a noise matrix based on MATLAB command $\sigma \times \texttt{rand}(100, 30)$ is added to $\mathbf{A}$. We set $\sigma = 0, 0.02, 0.04, \cdots, 0.1$. The relative approximation errors of the results by different methods are shown in Table 8. We can see that the approximation errors of TAP and AP methods are the lowest among the testing examples.

As a real-world application of ONMF, hyperspectral image unmixing aims at factoring the observed hyperspectral image in matrix format into an endmember matrix and

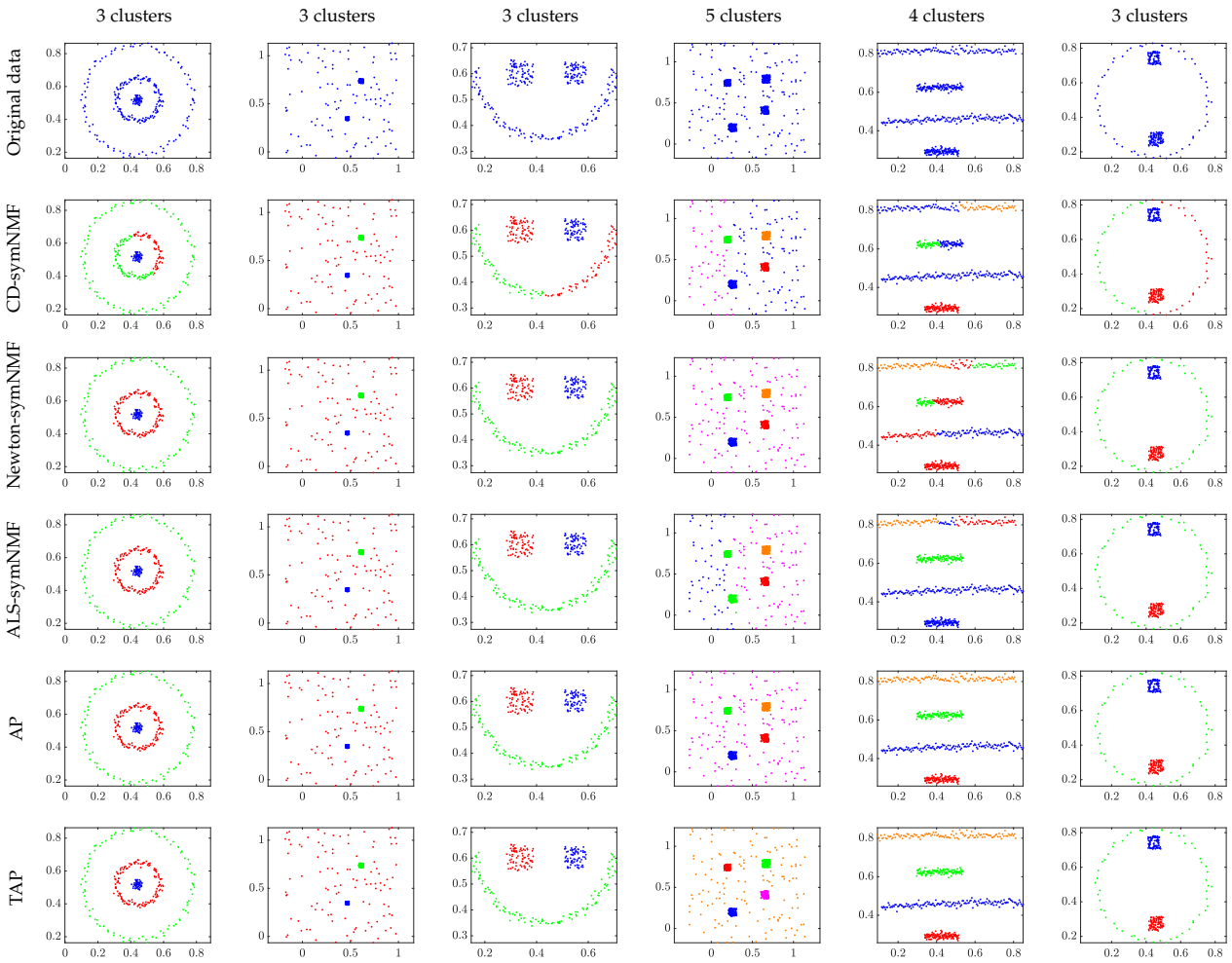


Fig. 9. The graph clustering results by the TAP (or AP) method and symmetric NMF methods on 6 cases of synthetic graph data. Different color represents different clusters.

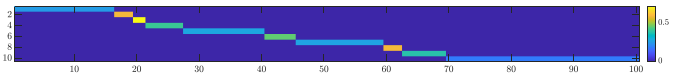


Fig. 10. An illustration of the generated $\mathbf{B}^T$.

TABLE 8
The relative approximation errors ($\times 100$) on the orthogonal symmetric matrix data. The **best** values and the second best values are respectively highlighted by bolder fonts and underlines.

σ	0	0.02	0.04	0.06	0.08	0.1
DTPP-ONMF	0.022	2.730	5.231	7.567	9.465	11.232
SM-ONMF	0.016	2.741	5.169	7.533	9.424	14.180
AP	**0.000**	**2.364**	**4.471**	**6.529**	**8.215**	**9.700**
TAP	**0.000**	**2.364**	**4.471**	**6.529**	**8.215**	**9.700**

an abundance matrix. The abundance matrix is indeed the classification of the pixels to different clusters, with each corresponding to a material (endmember). In this part, we use a sub-image of the Samson data set [73], consisting of $95 \times 95 = 9025$ spatial pixels and 156 spectral bands. We form a matrix A of size 9025×156 to represent this sub-image. Three different materials, i.e., "Tree", "Rock", and "Water", are in this sub-image, and we set the rank r as 3. The factor matrices $\mathbf{B} \in \mathbb{R}^{9025\times 3}$ and $\mathbf{C} \in \mathbb{R}^{3\times 156}$ can be obtained by the orthogonal NMF methods. We use k-means and do hard clustering on $\mathbf{B} \in \mathbb{R}^{9025\times 3}$ to obtain the abundance matrix, and we can obtain the i-th feature image by reshaping its i-th column to a 95×95 matrix. Each row of

TABLE 9
The quantitative metrics of the unmixing results on the hyperspetral image Samson. The **best** values and the second best values are respectively highlighted by bolder fonts and underlines.

Metric	DTPP-ONMF	SM-ONMF	AP	TAP
SAD	0.3490	0.4389	**0.0765**	**0.0765**
Similartity	0.5887	0.5640	**0.9383**	**0.9383**

$\mathbf{C}$ represents the spectral reflectance of on material ("Tree", "Rock", or "Water"). As for TAP and AP methods, we apply singular value decomposition on approximated non-negative low rank matrices to obtain the left singular value matrices which contain the first 3 left singular vectors. Then, we use k-means and do hard clustering on the left singular matrices to cluster three materials and obtain abundance matrices and endmember matrices.

To quantitatively evaluate the umixing results, we employ two metrics. The first one is the spectral angle distance (SAD) as follows:

$$\text{SAD} = \frac{1}{r}\sum_{i=1}^{r} \arccos\left(\frac{s_i^T \hat{s}_i}{\|s_i\|_2 \|\hat{s}_i\|_2}\right),$$

where $\{s_i\}_{i=1}^r$ are the estimated spectral reflectance (rows of the endmember matrix) and $\{\hat{s}_i\}_{i=1}^r$ are the groundtruth

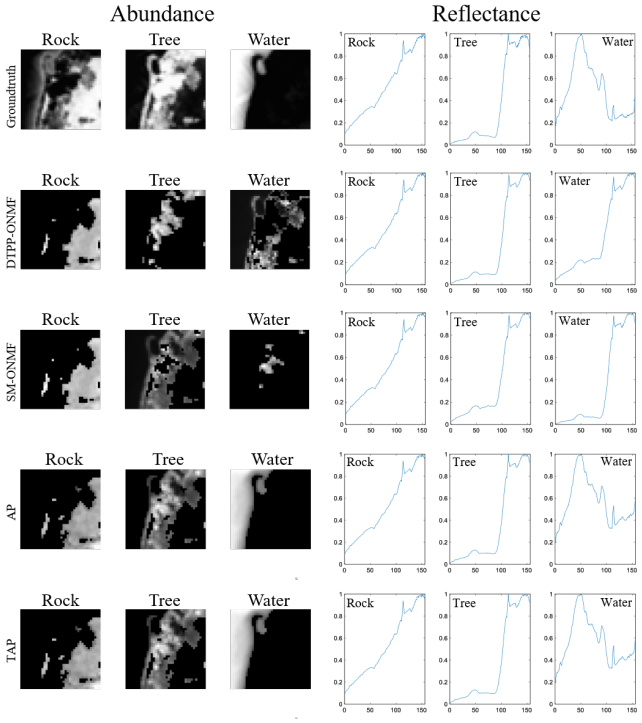


Fig. 11. Left: Abundance maps of Rock, Tree, and Water; Right: Reflectance of Rock, Tree, and Water. From top to bottom: groundtruth, DTPP-ONMF, SM-ONMF, AP, TAP.

TABLE 10
Data sets for community detection.

Data set	Karate [75]	Dolphins [76]	Friendship6 [77]	Friendship7 [77]	Football [78]	Polbooks [79]
# samples	34	62	68	68	115	105
# clusters	2	2	6	7	12	3

spectral reflectance. The second one is the similarity of the abundance feature image [74] as follows:

$$\text{Similarity} = \frac{1}{r}\sum_{i=1}^{r}\frac{a_i^T\hat{a}_i}{\|a_i\|_2\|\hat{a}_i\|_2},$$

where $\{a_i\}_{i=1}^r$ are the estimated abundance feature (columns of the abundance matrix) and $\{\hat{a}_i\}_{i=1}^r$ are the groundtruth ones. We note that a larger Similarity and a smaller SAD indicate a better unmixing result. We exhibit the quantitative metrics in Table 9. We can evidently see that the proposed TAP and AP methods obtain the best metrics. Meanwhile, we illustrate the estimated spectral reflectance and abundance feature images in Figure 9. It can be found from the second row that DTPP-ONMF and SM-ONMF perform well for the materials "Rock" and "Tree" but poor on "Water". TAP and AP methods unmix these three materials well, but the proposed TAP method (the computational time = 0.1492 seconds) is faster than the AP method (the computational time = 0.3738 seconds).

4.6 Other Applications

As we discussed in the introduction part, the NMF has been utilized in a wide range of applications. In this part, we select two representative examples, i.e., multi-view clustering and community detection, and show how our TAP could be applied for these tasks.

4.6.1 Community Detection

The community detection aims at figuring out groups of nodes with dense internal connections and sparse external connections, for real-world complex interaction systems characterized by complex networks. When the network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}|$ nodes and $m = |\mathcal{E}|$ edges is described by an adjacency matrix $\mathbf{A}$, which is symmetric, the community detection is a direct application of graph clustering on symmetric nonnegative matrices in Sec. 4.4. In this part, we select 6 widely-used real networks, listed in Table 10, for evaluation. As the superiority of our TAP over traditional symmetric NMF methods has been illustrated in Sec. 4.4, we consider two more recent methods, i.e., the deep nonlinear reconstruction method [80] (denoted as "DNR") and the deep autoencoder-like nonnegative matrix factorization method [53] (denoted as "DANMF"). We perform all the methods with feeding them the adjacency matrix $\mathbf{A}$. The clustering results of our TAP is obtained by using the k-means method on $\mathbf{U}$ while clustering results of DNR and DANMF are obtained by using the k-means method on their factors. We run DNR and DANMF 20 times with different random initializations and the k-means method is also conducted 20 times on $\mathbf{U}$ for our TAP. Two quantitative metrics, i.e., the accuracy and the NMI, are reported in Table 11. We can see that our method achieves comparable performance compared with DANMF and obtains the best results for many cases.

4.6.2 Multi-View Clustering

Compared to traditional data that describes objects from single perspective, multi-view data, collected from different sources in diverse domains (or obtained from various feature collectors), is semantically richer, more useful, however more complex. The goal of multi-view clustering is to explore the underlying structure of data by leveraging heterogeneous information of different views. In this part, we conduct experiments on the following multi-view datasets, which are commonly used in the literature.

• 3 source data set[3] (3sourse): This data set consists of 169 news reported by three news organizations, i.e., BBC, Reuters, and The Guardian. Each news was manually annotated with one of six topical labels.

• BBC data set[4] (BBC): It is collected from the BBC news website. BBC data set consists of 685 documents. Each document was split into four segments and was manually annotated with one of five topical labels.

• Handwritten digit 2 source data set[5] (HW2sources): This is a handwritten numerals (0−9) data set containing 2000 samples and 10 digits from two sources, i.e., MNIST Handwritten Digits and USPS Handwritten Digits.

• Yale-B 3 views [6] (Yale-B3): This data set is constructed via extracting three kinds of features , i.e., intensity, LBP [81], and Gabor [82], from 165 facial images (15 individuals) of the Yale B facial image data set.

The statistics of above data sets are summaries in Table 12.

3. http://mlg.ucd.ie/datasets/3sources.html
4. http://mlg.ucd.ie/datasets/segment.html
5. https://cs.nyu.edu/roweis/data.html
6. https://github.com/hdzhao/DMF_MVC/blob/master/data/yale_mtv.mat

TABLE 11
The quantitative metrics (mean values and standard deviations) of community detection results. The **best** values are highlighted by bolder fonts.

Method	Metrics	Karate [75]	Dolphins [76]	Friendship6 [77]	Friendship7 [77]	Football [78]	Polbooks [79]
DNR	Accuracy	88.53% (0.112)	93.15% (0.039)	72.03% (0.068)	75.51% (0.047)	87.09% (0.041)	74.38% (0.017)
	NMI	0.607 (0.287)	0.667 (0.109)	0.714 (0.046)	0.736 (0.038)	0.893 (0.018)	0.467 (0.019)
DANMF	Accuracy	**100.00% (0.000)**	**98.39% (0.000)**	80.22% (**0.018**)	**92.10% (0.014)**	86.22% (0.022)	82.33% (0.013)
	NMI	**1.000 (0.000)**	**0.889 (0.000)**	**0.814 (0.031)**	**0.877 (0.021)**	0.877 (0.013)	0.535 (0.016)
TAP	Accuracy	**100.00% (0.000)**	**98.39% (0.000)**	**81.52%** (0.050)	80.29% (0.050)	**90.70% (0.014)**	**82.86% (0.000)**
	NMI	**1.000 (0.000)**	**0.889 (0.000)**	0.764 (0.044)	0.763 (0.037)	**0.918 (0.011)**	**0.571 (0.000)**

TABLE 12
Multi-view data sets.

Data set	# samples	# views	# clusters
3sources	169	3	6
BBC	685	4	5
HW2sources	2000	2	10
Yale-B2	165	3	15

Our TAP is designed for the approximation of single-view matrices and it is interesting to extend our method for multi-view clustering. It would be our future research direction. As we can see in Sec. 4.3, here our method could be helpful when it serves as a preprocessing step. That is, we apply our nonnegative low rank matrix approximation method firstly on data matrices with different views, the performance of the subsequent multi-view clustering method could be improved. In order to validate this preprocessing procedure, we compare multi-view clustering methods with and without the preprocessing by our method on above data sets. Selected multi-view clustering methods are the multiview concept clustering (denoted as "MVCC") [48] method , which is based on the matrix concept factorization with the local manifold regularization, the graph-based multi-view clustering (denoted as "GMC") method [83], and a deep matrix factorization (DMF) [52] method. In Table 13, we report quantitative metrics, i.e., the accuracy and the NMI, of all results on four data sets. As the results of DMF and MVCC would vary with different initializations, we run DMF and MVCC 10 trails and report the mean value and the standard deviation. We can see that, compared with GMC, MVCC is more suitable for the data sets 3sources and BBC. With the help of our TAP, all the methods obtain better results. Some improvements brought in by our method are significant, e.g., the MVCC on HW2sources, GMC on 3sources and BBC, and DMF on Yale-B2. Meanwhile, when the data matrices are preprocessed by our method, the standard deviations also become smaller in many cases.

5 Conclusion

In this paper, we have proposed a new alternating projection method to compute nonnegative low rank matrix approximation for nonnegative matrices. Our main idea is to use the tangent space of the point in the fixed-rank matrix manifold to approximate the projection onto the manifold in order to reduce the computational cost. Numerical examples in data clustering, pattern recognition and hyperspectral data analysis have shown that the proposed alternating projection method is better than that of nonnegative matrix factorization methods in terms of accuracy, and the computational time required by the proposed alternating projection method is less than that required by the original alternating projection method.

Moreover, we have shown that the sequence generated by the alternating projections onto the tangent spaces of the fixed rank matrices manifold and the nonnegative matrix manifold, converge linearly to a point in the intersection of the two manifolds where the convergent point is sufficiently close to optimal solutions. Our theoretical convergence results are new and are not studied in the literatures. We remark that Andersson and Carlsson [70] assumed that the exact projection onto each manifold and then obtained the convergence result of the alternating projection method. Because of our proposed inexact projection onto each manifold, our proof can be extended to show the sequence generated by alternating projections on one or two nontangential manifolds based on tangent spaces, converges linearly to a point in the intersection of the two manifolds.

As a future research work, it is interesting to study (i) the convergence results when inexact projections on several manifolds are employed, and (ii) applications where the other norms (such as l_1 norm) in data fitting instead of the Frobenius norm. It is necessary to develop the related algorithms for such manifold optimization problems. Meanwhile, it will also be interesting to extend our method for the case where multiple data matrices need to be processed.

References

[1] K. Chen, *Matrix preconditioning techniques and applications*. Cambridge University Press, 2005, vol. 19.

[2] M. Chen, W.-S. Chen, B. Chen, and B. Pan, "Non-negative sparse representation based on block nmf for face recognition," in *Chinese Conference on Biometric Recognition*. Springer, 2013, pp. 26–33.

[3] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proceedings of the 2005 SIAM International Conference on Data Mining*. SIAM, 2005, pp. 606–610.

[4] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2006, pp. 126–135.

[5] D. Guillamet and J. Vitria, "Non-negative matrix factorization for face recognition," in *Catalonian Conference on Artificial Intelligence*. Springer, 2002, pp. 336–344.

[6] D. Guillamet, J. Vitria, and B. Schiele, "Introducing a weighted non-negative matrix factorization for image classification," *Pattern Recognition Letters*, vol. 24, no. 14, pp. 2447–2454, 2003.

[7] L. Jing, J. Yu, T. Zeng, and Y. Zhu, "Semi-supervised clustering via constrained symmetric non-negative matrix factorization," in *International Conference on Brain Informatics*. Springer, 2012, pp. 309–319.

[8] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[9] J. Liu, Z. Wu, Z. Wei, L. Xiao, and L. Sun, "A novel sparsity constrained nonnegative matrix factorization for hyperspectral unmixing," in *2012 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2012, pp. 1389–1392.

TABLE 13
The quantitative metrics (mean values and standard deviations) of multi-view clustering results. The **best** values are highlighted by bolder fonts. For results by MVCC and DMF, we report the mean values and the standard deviations.

Method	Metrics	Data sets			Method	Metrics	Data sets			Method	Metrics	Data set
		3sources	BBC	HW2sources			3sources	BBC	HW2sources			Yale-B2
MVCC	Accuracy	74.55%(0.018)	74.16% (0.051)	58.72% (0.055)	GMC	Accuracy	69.23%	69.34%	99.40%	DMF	Accuracy	71.715% (0.007)
	NMI	0.707 (0.015)	0.606 % (0.038)	0.618 (**0.035**)		NMI	0.6216	0.562	0.985		NMI	0.709 (**0.002**)
MVCC +TAP	Accuracy	**75.50% (0.026)**	77.32% (**0.038**)	68.83% (**0.045**)	GMC +TAP	Accuracy	75.14%	**87.88%**	**99.60%**	DMF +TAP	Accuracy	**78.769%** (0.005)
	NMI	**0.688 (0.008)**	0.595 (**0.035**)	0.699 (0.042)		NMI	0.6495	**0.740**	**0.989**		NMI	**0.757** (0.003)

[10] Y. Liu, X.-Z. Pan, R.-J. Shi, Y.-L. Li, C.-K. Wang, and Z.-T. Li, "Predicting soil salt content over partially vegetated surfaces using non-negative matrix factorization," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 11, pp. 5305–5316, 2015.

[11] Y. Wang, Y. Jia, C. Hu, and M. Turk, "Non-negative matrix factorization framework for face recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 19, no. 04, pp. 495–511, 2005.

[12] D. Zhang, S. Chen, and Z.-H. Zhou, "Two-dimensional non-negative matrix factorization for face representation and recognition," in *International Workshop on Analysis and Modeling of Faces and Gestures*. Springer, 2005, pp. 350–363.

[13] M. W. Berry and J. Kogan, *Text mining: applications and theory*. John Wiley & Sons, 2010.

[14] T. Li and C. Ding, "The relationships among various nonnegative matrix factorization methods for clustering," in *Sixth International Conference on Data Mining (ICDM'06)*. IEEE, 2006, pp. 362–371.

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

[16] 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 Informaion Retrieval*, 2003, pp. 267–273.

[17] 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 & Sons, 2009.

[18] P. M. Kim and B. Tidor, "Subsystem identification through dimensionality reduction of large-scale gene expression data," *Genome Research*, vol. 13, no. 7, pp. 1706–1718, 2003.

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

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

[21] G. Wang, A. V. Kossenkov, and M. F. Ochs, "LS-NMF: a modified non-negative matrix factorization algorithm utilizing uncertainty estimates," *BMC Bioinformatics*, vol. 7, no. 1, p. 175, 2006.

[22] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[23] A. Cichocki, R. Zdunek, and S.-i. Amari, "Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization," in *International Conference on Independent Component Analysis and Signal Separation*. Springer, 2007, pp. 169–176.

[24] N. Gillis and F. Glineur, "Accelerated multiplicative updates and hierarchical ALS algorithms for nonnegative matrix factorization," *Neural Computation*, vol. 24, no. 4, pp. 1085–1105, 2012.

[25] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Computation*, vol. 19, no. 10, pp. 2756–2779, 2007.

[26] N. Guan, D. Tao, Z. Luo, and B. Yuan, "NeNMF: An optimal gradient method for nonnegative matrix factorization," *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2882–2898, 2012.

[27] H. Kim and H. Park, "Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method," *SIAM journal on matrix analysis and applications*, vol. 30, no. 2, pp. 713–730, 2008.

[28] J. Kim and H. Park, "Toward faster nonnegative matrix factorization: A new algorithm and comparisons," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 353–362.

[29] S. Wild, J. Curry, and A. Dougherty, "Improving non-negative matrix factorizations through structured initialization," *Pattern recognition*, vol. 37, no. 11, pp. 2217–2232, 2004.

[30] Z. Liu and V. Y. Tan, "Rank-one nmf-based initialization for nmf and relative error bounds under a geometric assumption," *IEEE Transactions on Signal Processing*, pp. 4717–4731, 2017.

[31] S. M. Atif, S. Qazi, and N. Gillis, "Improved svd-based initialization for nonnegative matrix factorization using low-rank correction," *Pattern Recognition Letters*, vol. 122, pp. 53–59, 2019.

[32] Y. Gao and G. Church, "Improving molecular cancer class discovery through sparse non-negative matrix factorization," *Bioinformatics*, vol. 21, no. 21, pp. 3970–3975, 2005.

[33] N. Gillis and F. Glineur, "Using underapproximations for sparse nonnegative matrix factorization," *Pattern recognition*, vol. 43, no. 4, pp. 1676–1687, 2010.

[34] B. Du, S. Wang, N. Wang, L. Zhang, D. Tao, and L. Zhang, "Hyperspectral signal unmixing based on constrained non-negative matrix factorization approach," *Neurocomputing*, vol. 204, pp. 153–161, 2016.

[35] S. Choi, "Algorithms for orthogonal nonnegative matrix factorization," in *2008 IEEE International Joint Vonference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, 2008, pp. 1828–1832.

[36] D. Tolić, N. Antulov-Fantulin, and I. Kopriva, "A nonlinear orthogonal non-negative matrix factorization approach to subspace clustering," *Pattern Recognition*, vol. 82, pp. 40–55, 2018.

[37] D. Kuang, C. Ding, and H. Park, "Symmetric nonnegative matrix factorization for graph clustering," in *Proceedings of the 2012 SIAM International Conference on Data Mining*. SIAM, 2012, pp. 106–117.

[38] A. Vandaele, N. Gillis, Q. Lei, K. Zhong, and I. Dhillon, "Coordinate descent methods for symmetric nonnegative matrix factorization," *arXiv preprint arXiv:1509.01404*, 2015.

[39] X. Luo, Z. Liu, M. Shang, J. Lou, and M. Zhou, "Highly-accurate community detection via pointwise mutual information-incorporated symmetric non-negative matrix factorization," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, pp. 463–476, 2020.

[40] N. Gillis and S. A. Vavasis, "Fast and robust recursive algorithms-for separable nonnegative matrix factorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 4, pp. 698–714, 2013.

[41] M. C. U. Araújo, T. C. B. Saldanha, R. K. H. Galvao, T. Yoneyama, H. C. Chame, and V. Visani, "The successive projections algorithm for variable selection in spectroscopic multicomponent analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 57, no. 2, pp. 65–73, 2001.

[42] J. Pan and N. Gillis, "Generalized separable nonnegative matrix factorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[43] S. Zafeiriou, A. Tefas, I. Buciu, and I. Pitas, "Exploiting discriminant information in nonnegative matrix factorization with application to frontal face verification," *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 683–695, 2006.

[44] M. Das Gupta and J. Xiao, "Non-negative matrix factorization as a feature selection tool for maximum margin classifiers," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 2841–2848.

[45] J. Ma, Y. Zhang, and L. Zhang, "Discriminative subspace matrix factorization for multiview data clustering," *Pattern Recognition*, vol. 111, p. 107676, 2021.

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

[47] L. Zong, X. Zhang, L. Zhao, H. Yu, and Q. Zhao, "Multi-view clustering via multi-manifold regularized non-negative matrix factorization," *Neural Networks*, vol. 88, pp. 74–89, 2017.

[48] H. Wang, Y. Yang, and T. Li, "Multi-view clustering via concept factorization with local manifold regularization," in *IEEE 16th International Conference on Data Mining (ICDM)*, Bacerlona, Spain, December 2016, pp. 1245–1250.

[49] T.-X. Jiang, L. Zhuang, T.-Z. Huang, X.-L. Zhao, and J. M. Bioucas-Dias, "Adaptive hyperspectral mixed noise removal," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[50] Y. Zhang and D.-Y. Yeung, "Overlapping community detection via bounded nonnegative matrix tri-factorization," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 606–614.

[51] Y. Chen, Z. Lei, Y. Rao, H. Xie, F. L. Wang, J. Yin, and Q. Li, "Parallel non-negative matrix tri-factorization for text data co-clustering," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[52] H. Zhao, Z. Ding, and Y. Fu, "Multi-view clustering via deep matrix factorization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[53] F. Ye, C. Chen, and Z. Zheng, "Deep autoencoder-like nonnegative matrix factorization for community detection," in *Proceedings of the 27th ACM international conference on information and knowledge management*, 2018, pp. 1393–1402.

[54] Y. Zhao, H. Wang, and J. Pei, "Deep non-negative matrix factorization architecture based on underlying basis images learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 6, pp. 1897–1913, 2019.

[55] A. Man Shun Ang, J. E. Cohen, N. Gillis, and L. Thi Khanh Hien, "Accelerating block coordinate descent for nonnegative tensor factorization," *Numerical Linear Algebra with Applications*, vol. 28, no. 5, p. e2373, 2021.

[56] L. Zhang, L. Song, B. Du, and Y. Zhang, "Nonlocal low-rank tensor completion for visual data," *IEEE transactions on cybernetics*, vol. 51, no. 2, pp. 673–685, 2019.

[57] T.-X. Jiang, X.-L. Zhao, H. Zhang, and M. K. Ng, "Dictionary learning with low-rank coding coefficients for tensor completion," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[58] Y.-C. Miao, X.-L. Zhao, X. Fu, J.-L. Wang, and Y.-B. Zheng, "Hyperspectral denoising using unsupervised disentangled spatiospectral deep priors," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–16, 2021.

[59] T. He, Y. Liu, T. H. Ko, K. C. Chan, and Y.-S. Ong, "Contextual correlation preserving multiview featured graph clustering," *IEEE Transactions on Cybernetics*, vol. 50, no. 10, pp. 4318–4331, 2019.

[60] C.-H. Lin and J. M. Bioucas-Dias, "Nonnegative blind source separation for ill-conditioned mixtures via john ellipsoid," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 5, pp. 2209–2223, 2020.

[61] C. Kervazo, N. Gillis, and N. Dobigeon, "Provably robust blind source separation of linear-quadratic near-separable mixtures," *SIAM Journal on Imaging Sciences*, vol. 14, no. 4, pp. 1848–1889, 2021.

[62] G. R. Naik, *Non-negative matrix factorization techniques*. Springer, 2016.

[63] N. Gillis, *Nonnegative matrix factorization*. SIAM, 2020.

[64] Y.-X. Wang and Y.-J. Zhang, "Nonnegative matrix factorization: A comprehensive review," *IEEE Transactions on knowledge and data engineering*, vol. 25, no. 6, pp. 1336–1353, 2012.

[65] X. Fu, K. Huang, N. D. Sidiropoulos, and W.-K. Ma, "Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications," *IEEE Signal Processing Magazine*, vol. 36, no. 2, pp. 59–80, 2019.

[66] G. Song and M. K. Ng, "Nonnegative low rank matrix approximation for nonnegative matrices," *Applied Mathematics Letters*, vol. 105, p. 106300, 2020.

[67] J. Ye, "Generalized low rank approximations of matrices," *Machine Learning*, vol. 61, no. 1, pp. 167–191, 2005.

[68] B. Vandereycken, "Low-rank matrix completion by riemannian optimization," *SIAM Journal on Optimization*, vol. 23, no. 2, pp. 1214–1236, 2013.

[69] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2012, vol. 3.

[70] F. Andersson and M. Carlsson, "Alternating projections on non-tangential manifolds," *Constructive Approximation*, vol. 38, no. 3, pp. 489–525, 2013.

[71] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, 2001, pp. 556–562.

[72] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Advances in neural information processing systems*, 2005, pp. 1601–1608.

[73] F. Zhu, Y. Wang, B. Fan, S. Xiang, G. Meng, and C. Pan, "Spectral unmixing via data-guided sparsity," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5412–5427, 2014.

[74] J. Pan, M. K. Ng, Y. Liu, X. Zhang, and H. Yan, "Orthogonal non-negative tucker decomposition," *arXiv preprint arXiv:1912.06836*, 2019.

[75] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of anthropological research*, vol. 33, no. 4, pp. 452–473, 1977.

[76] D. Lusseau and M. E. Newman, "Identifying the role that animals play in their social networks," *Proceedings of the Royal Society of London. Series B: Biological Sciences*, vol. 271, no. suppl_6, pp. S477–S481, 2004.

[77] J. Xie, S. Kelley, and B. K. Szymanski, "Overlapping community detection in networks: The state-of-the-art and comparative study," *Acm computing surveys (csur)*, vol. 45, no. 4, pp. 1–35, 2013.

[78] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[79] M. E. Newman, "Fast algorithm for detecting community structure in networks," *Physical review E*, vol. 69, no. 6, p. 066133, 2004.

[80] L. Yang, X. Cao, D. He, C. Wang, X. Wang, and W. Zhang, "Modularity based community detection with deep learning." in *IJCAI*, vol. 16, 2016, pp. 2252–2258.

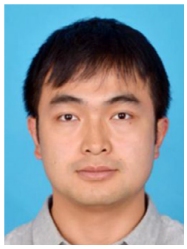
[81] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 2037–2041, 2006.

[82] H. G. Feichtinger and T. Strohmer, *Gabor analysis and algorithms: Theory and applications*. Springer Science & Business Media, 2012.

[83] H. Wang, Y. Yang, and B. Liu, "GMC: Graph-based multi-view clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 6, pp. 1116–1129, 2019.

Guang-Jing Song received the Ph.D. degree in mathematics from Shanghai University, Shanghai, China, in 2010. He is currently a Professor of School of Mathematics and Information Sciences, Weifang University. His research interests include numerical linear algebra, sparse and low-rank modeling, tensor decomposition and multi-dimensional image processing.

Michael K. Ng is the Director of Research Division for Mathematical and Statistical Science, and Chair Professor of Department of Mathematics, the University of Hong Kong, and Chairman of HKU-TCL Joint Research Center for AI. His research areas are data science, scientific computing, and numerical linear algebra.



Tai-Xiang Jiang received the B.S., Ph.D. degrees in mathematics and applied mathematics from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2013. He is currently a Professor with the School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics. His research interests include sparse and low-rank modeling, tensor decomposition and multi-dimensional image processing. https://taixiangjiang.github.io/