Adaptive Method for Nonsmooth Nonnegative Matrix Factorization

Zuyuan Yang, *Member, IEEE*, Yong Xiang, *Senior Member, IEEE*, Kan Xie, and Yue Lai

***Abstract*— Nonnegative matrix factorization (NMF) is an emerging tool for meaningful low-rank matrix representation. In NMF, explicit constraints are usually required, such that NMF generates desired products (or factorizations), especially when the products have significant sparseness features. It is known that the ability of NMF in learning sparse representation can be improved by embedding a smoothness factor between the products. Motivated by this result, we propose an adaptive nonsmooth NMF (Ans-NMF) method in this paper. In our method, the embedded factor is obtained by using a data-related approach, so it matches well with the underlying products, implying a superior faithfulness of the representations. Besides, due to the usage of an adaptive selection scheme to this factor, the sparseness of the products can be separately constrained, leading to wider applicability and interpretability. Furthermore, since the adaptive selection scheme is processed through solving a series of typical linear programming problems, it can be easily implemented. Simulations using computer-generated data and real-world data show the advantages of the proposed Ans-NMF method over the state-of-the-art methods.**

***Index Terms*— Linear programming (LP), nonnegative matrix factorization (NMF), smoothness constraint, sparse representation.**

I. Introduction

NONNEGATIVE matrix factorization (NMF) is a well-known matrix decomposition technique for data analysis [1]–[5]. It aims to decompose a given nonnegative matrix into two low-rank nonnegative matrices (or products), called feature matrix and coefficient matrix, respectively [6]. Since NMF can produce compact representation for the original data [7]–[9], it has great potential to dimension reduction, feature extraction, and so on. In fact, NMF has been successfully applied to a variety of real-world data processing, such as face verification [10], document clustering [11], hyperspectral data unmixing [12], and so on. However, standard NMF does not always give a desired representation, e.g., the obtained result could be far from the expected decomposition [13] and might not match the underlying elements of the data [14], [15].

To improve the interpretability of the products, one often needs to add explicit constraints to the standard NMF. Some efficient constraints have been proposed for particular applications, such as the sparseness constraint in spectral unmixing and blind deconvolution [16], [17], the margin constraint for classification [18], the label constraint in clustering [19], and the volume constraint in blind source separation [20]–[22]. In general, different constraints lead to different kinds of results. Sparseness is a widely used constraint as both the feature and the coefficient products learned from the sparse NMF are meaningful. For example, a sparse feature reflects a parts-based representation, which is meaningful in information extraction, while a sparse coefficient means that the needed feature number is small for reconstructing the input data.

Hoyer [15] develops a sparseness constraint-based NMF (Sc-NMF) method, together with an efficient and popular sparseness measure. This method can result in decompositions with specific sparseness degree by using a special projection operator, and thus it can manage the sparseness of the products conveniently. With regard to interpretability, Sc-NMF performs better than the standard NMF in learning the parts-based representation of the face image data set used in [15]. More importantly, it can learn oriented features with biomedical meaning for representing natural images. However, it is challenging to keep the faithfulness of the results, especially when the sparseness constraint is applied to both of the products.

Recently, a nonsmoothness NMF (Ns-NMF) method is developed in [23]. It enhances the faithfulness of the decomposed results by using an interesting scheme, in which a parametric smoothness factor matrix is embedded between the products generated by the standard NMF. As shown in [23], the smoother the parametric factor, the sparser the products. Thus, similar to the aforementioned Sc-NMF method, the Ns-NMF method also has the capability of generating sparse decompositions. However, due to sharing a common and manually created smoothness factor, the sparseness levels of the products cannot be separately regulated, even though the explicit smoothness constraint is applied.

Manuscript received June 1, 2014; accepted December 26, 2015. Date of publication January 28, 2016; date of current version March 15, 2017. This work was supported in part by the Guangdong Natural Science Funds for Distinguished Young Scholar under Grant 2014A030306037, in part by the Program for NCET under Grant NCET-13-0740, in part by the National Natural Science Foundation of China under Grant 61273192, Grant 61322306, Grant 61333013, and Grant 61503083, in part by the Special Planning of Guangdong under Grant 2014TQ01X144, in part by the Zhujiang New-Star of Guangzhou, and in part by the Guangdong Funds for Excellent Doctoral Dissertations Scholar under Grant sybzzxm201214. *(Corresponding author: Yue Lai.)*

Z. Yang is with the Faculty of Automation, Guangdong University of Technology, Guangzhou 510006, China, and also with the School of Information Technology, Deakin University, Melbourne, VIC 3125, Australia (e-mail: yangzuyuan@aliyun.com).

Y. Xiang is with the School of Information Technology, Deakin University, Melbourne, VIC 3125, Australia (e-mail: yxiang@deakin.edu.au).

K. Xie is with the Institute of Intelligent Signal Processing, Guangdong University of Technology, Guangzhou 510006, China (e-mail: kanxiegdut@gmail.com).

Y. Lai is with the Faculty of Automation, Guangdong University of Technology, Guangzhou 510006, China (e-mail: hot_day@163.com).

Digital Object Identifier 10.1109/TNNLS.2016.2517096

TABLE I
NOTATIONS

$\mathbf{x}$, x_i	column vector, the ith element of $\mathbf{x}$
$\mathbf{X}$, $\mathbf{x}_j$, x_{ij}	matrix, the jth column of $\mathbf{X}$, the (i,j)th entry of $\mathbf{X}$
$\mathbf{X}^T$, $\mathbf{X}^{-1}$	transpose of $\mathbf{X}$, inverse of $\mathbf{X}$
$\|\mathbf{X}\|_F^2$	squared sum of all entries of $\mathbf{X}$
$\max(\mathbf{X})$	maximum of the entries of $\mathbf{X}$
$\det(\mathbf{X})$	determinant of $\mathbf{X}$
$\Re$	real number set
$\mathbf{I}$	identity matrix
$\mathbf{0}(\mathbf{1})$	zero (one) column vector (or matrix)
$\otimes$, $\oslash$, $\succeq$	component-wise multiplication, division, not less than.

Motivated by that the factor embedding scheme in [23] is helpful for both reducing decomposition error and enhancing the sparsity of the products in NMF, and an adaptive nonsmooth NMF (Ans-NMF) method is proposed in this paper, where the smoothness factor (matrix) is adaptively obtained by using a data-related approach, rather than manually created in the existing works using an *ad hoc* manner. This approach helps to find the underlying sparse products, which can match well with the original data. As a result, Ans-NMF has the benefit of balancing sparseness and faithfulness, i.e., it tends to generate a sparse representation with a high faithfulness. Furthermore, by using the adaptive approach to select the explicit smoothness factor, the Ans-NMF method can be able to constrain the sparsity of the products separately. This further enhances the applicability and flexibility of the Ans-NMF method.

In relation to implementation, the proposed Ans-NMF method works under the framework of the well-known alternating iterative update rule [13], [18], [24], [25]. The key part of our method is to find the desired smoothness factor. Regarding this, the smoothness factor is updated by separately absorbing its left neighbor matrix and right neighbor matrix, which originates from the multilayer NMF technique [26]. These two neighbor matrices are obtained by solving a series of typical linear programming (LP) problems. After the smoothness factor is attained, the algorithm in [23] can be invoked to obtain the final results.

The remainder of the paper is organized as follows. Section II provides a brief review of the NMF-style methods. Section III presents the proposed Ans-NMF method, including the model and the algorithm. The effectiveness of the new method is demonstrated by simulation results in Section IV, where both computer-generated data and real-world data are utilized. Finally, Section V concludes the paper, together with the discussion to the extension of the proposed method.

The notations used in the paper are given in Table I.

II. Review of NMF-Style Methods

In this section, we briefly review the standard NMF, constrained NMF, and Ns-NMF, which is closely related to our method.

A. NMF

Given a nonnegative matrix $\mathbf{V}$, the standard NMF aims at finding two low-rank nonnegative matrices $\mathbf{W}$ and $\mathbf{H}$, such that $\mathbf{V}$ approximates their product. Mathematically, the standard NMF can be described as follows:

$$\mathbf{V} \approx \mathbf{WH} \tag{1}$$

where $\mathbf{V} \in \Re^{m\times n}$, $\mathbf{W} \in \Re^{m\times r}$, $\mathbf{H} \in \Re^{r\times n}$, and $m, r, n (r < m, r < n)$ denote the number of the data dimension, the rank of the factorization, and the number of data samples, respectively. In order to achieve NMF, several cost functions have been proposed for particular applications [1], [5], [27], [28]. Without loss of generality, the Euclidean distance-based function is utilized throughout this paper. Then, the Euclidean distance-based optimization model to (1) is

$$\text{Min } \frac{1}{2}\|\mathbf{V} - \mathbf{WH}\|_F^2 \tag{2}$$

subject to $\mathbf{W} \succeq 0, \mathbf{H} \succeq 0$, where $\succeq$ denotes the componentwise inequality.

For solving (2), the algorithm based on the alternating iterative multiplication update rule is often employed [5], [13]. In addition, for a given $\mathbf{V}$, the corresponding iteration formulas with random initializations are as follows:

$$\mathbf{H} := \mathbf{H} \otimes (\mathbf{W}^T\mathbf{V}) \oslash (\mathbf{W}^T\mathbf{WH}) \tag{3}$$

$$\begin{cases} \mathbf{W} := \mathbf{W} \otimes (\mathbf{VH}^T) \oslash (\mathbf{WHH}^T) \\ \mathbf{W} := \mathbf{W}(\text{diag}(1 \oslash (\tilde{\mathbf{w}}))) \end{cases} \tag{4}$$

with

$$\tilde{\mathbf{w}} = \left(\sum_{i=1}^{m} \mathbf{w}^i\right)^T$$

where $\mathbf{w}^i$ is the ith row of $\mathbf{W}$, $\text{diag}(\mathbf{x})$ is a diagonal matrix whose diagonal entries correspond to the elements of the vector $\mathbf{x}$, and $\otimes$ and $\oslash$ are the componentwise multiplication and division, respectively. The second formula in (4) is used to normalize $\mathbf{W}$ to tackle the inevitable scale problem.

Clearly, there is no explicit constraint on the products $\mathbf{W}$ and $\mathbf{H}$ in (2), except for nonnegativity. Thus, it is almost impossible to ensure that the standard NMF always generates a desired result. To enhance the interpretability of the results, lots of works further analyze the explicit constraints to the traditional NMF.

B. NMF With Constraints

In general, the constrained NMF can be modeled as

$$\text{Min } \frac{1}{2}\|\mathbf{V} - \mathbf{WH}\|_F^2 + \alpha J_1(\mathbf{W}) + \beta J_2(\mathbf{H}) \tag{5}$$

subject to $\mathbf{W} \succeq 0, \mathbf{H} \succeq 0$, where $J_1(\mathbf{W})$ and $J_2(\mathbf{H})$ are penalty functions used to enforce certain constraints on the solution, and α and β are their corresponding regularization parameters. As for the exact constraints, they depend on the expectation to the results. For example, to obtain the desired $\mathbf{W}$ with orthogonal columns approximately, it is set in [29] that $\beta = 0$ and

$$J_1(\mathbf{W}) = \sum_{i=1}^{r}\sum_{j\neq i} \mathbf{w}_i\mathbf{w}_j^T. \tag{6}$$

In frontal face verification, the constraints are related to the discriminant information of the facial image vectors [10]. In data classification, they are connected with the maximum margin of interclasses [18]. Sparsity is another widely used constraint, which has aroused considerable interests.

Regarding the sparsity constraint, people often utilize the L2/L1-norm-based methods to impose sparsity [15], [30], and the corresponding sparseness measure for a vector $\mathbf{x} \in \Re^m$ is

$$S_{\mathbf{x}} = \frac{\sqrt{m} - \left(\sum_{i=1}^{m} x_i\right) \Big/ \left(\sqrt{\sum_{i=1}^{m} x_i^2}\right)}{\sqrt{m} - 1}. \tag{7}$$

These methods can be easily implemented, but seldom explore the spatial structure of data. Thus, it is difficult for them to generate nonoverlapping results, discouraging their applications in extracting independent features. The sparsity constraint is also introduced in [31], where the additional prior information of the basis features is incorporated. To impose sparsity to both $\mathbf{W}$ and $\mathbf{H}$, the attractive Ns-NMF method is developed in [23], and the details of Ns-NMF are shown in Section II-C.

C. Ns-NMF

Compared with the standard NMF, the following smoothness factor matrix $\mathbf{S}$ is embedded between $\mathbf{W}$ and $\mathbf{H}$ in Ns-NMF [23]:

$$\mathbf{S} = (1-\theta)\mathbf{I} + \frac{\theta}{r}\mathbf{1}\mathbf{1}^T \tag{8}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{1}$ is a vector of ones, r is the column number of $\mathbf{W}$, and the parameter θ belongs to the interval [0, 1]. Under this embedded factor, the corresponding Euclidean distance-based optimization model for Ns-NMF is given by

$$\text{Min } \frac{1}{2}\|\mathbf{V} - \mathbf{WSH}\|_F^2 \tag{9}$$

subject to $\mathbf{W} \succeq 0, \mathbf{H} \succeq 0$. Similar to the method solving (2), the multiplication update algorithm is also utilized to solve (9). For the given $\mathbf{V}$ and $\mathbf{S}$, the corresponding iteration formulas are as follows [23]:

$$\mathbf{H} := \mathbf{H} \otimes ((\mathbf{WS})^T\mathbf{V}) \oslash ((\mathbf{WS})^T\mathbf{WSH}) \tag{10}$$

$$\begin{cases} \mathbf{W} := \mathbf{W} \otimes (\mathbf{V}(\mathbf{SH})^T) \oslash (\mathbf{WSH}(\mathbf{SH})^T) \\ \mathbf{W} := \mathbf{W}(\text{diag}(1 \oslash (\tilde{\mathbf{w}}))). \end{cases} \tag{11}$$

The embedded smoothness factor $\mathbf{S}$ has great potential to balancing the decomposition error and the sparsity of $\mathbf{W}$ and $\mathbf{H}$, and thus improving the learning ability of NMF explicitly. As explained and shown in [23], it can force $\mathbf{W}$ and $\mathbf{H}$ to be sparse during the iteration. However, since $\mathbf{S}$ is manually created and shared by $\mathbf{W}$ and $\mathbf{H}$, as shown in (9), enhancing the sparseness of $\mathbf{W}$ (respectively $\mathbf{H}$) by using the explicit smoothness constraint will inevitably augment that of $\mathbf{H}$ (respectively $\mathbf{W}$). That is, the sparsity constraints to $\mathbf{W}$ and $\mathbf{H}$ are bound together in Ns-NMF. Consequently, $\mathbf{W}$ and $\mathbf{H}$ cannot be sparsified separately. This restricts its application to some practical problems, in which the sparseness degrees of the two factors are not consistent. For example, in the feature extraction of natural images, the corresponding coefficient can be much sparser than the feature based on the test in [15]. In this case, it is difficult for Ns-NMF to learn an oriented feature, which has special meaning. The aforementioned restriction is also verified by the simulations in Section IV-D, in which Ns-NMF does not extract the desired oriented features.

III. Adaptive Nonsmooth NMF

In this section, we propose the new Ans-NMF method, which can regulate the sparsity of $\mathbf{W}$ and $\mathbf{H}$ separately. This section includes two parts: 1) the proposition of the Ans-NMF model and 2) the derivation of the corresponding factorization algorithm.

A. Model

The involved NMF problem is composed of two aspects: 1) the decomposition error and 2) the optional sparsity constraints on $\mathbf{W}$ or $\mathbf{H}$ or both. Regarding the NMF error function, we use without loss of generality, the Euclidean distance-based function in our method. Then, the proposed model is described as follows.

Given a nonnegative matrix $\mathbf{V}$, find the nonnegative matrices $\mathbf{W}$, $\mathbf{S}$, and $\mathbf{H}$ such that

$$D = \frac{1}{2}\|\mathbf{V} - \mathbf{WSH}\|_F^2 \tag{12}$$

is minimized, under the optional sparsity constraints on $\mathbf{W}$ or $\mathbf{H}$, where $\sum_{i=1}^{m} w_{ij} = 1, \sum_{j=1}^{r} s_{jk} = 1$, and the usage of constraints depends on practical requirements.

In the above model, $\mathbf{W}$ and $\mathbf{S}$ are forced to be column-sum-to-one to deal with the scaling problem. Motivated by [23], we replace the optional sparsity constraints on $\mathbf{W}$ and $\mathbf{H}$ with the smoothness constraint on $\mathbf{S}$ for a better balance between decomposition error and sparsity. The respective sparseness requirements on $\mathbf{W}$ and $\mathbf{H}$ are achieved by adaptively selecting the smoothness of $\mathbf{S}$. To give a more clear explanation to (12), the mentioned smoothness constraint is further analyzed here.

We first introduce a smoothness measure for the nonnegative square matrix $\mathbf{S}$, which is of full rank and column-sum-to-one. Based on the analysis about smoothness and sparseness [32], [33], the smoothness (or sparseness) of a full column rank matrix with column-sum-to-one can be measured by the determinant of its Gram matrix, and a smaller determinant implies a higher smoothness level. Thus, one can measure the smoothness of $\mathbf{S}$ by $\det(\mathbf{S}^T\mathbf{S})$. Since $\mathbf{S}$ is square, it holds that $\det(\mathbf{S}) = \det(\mathbf{S}^T)$. Then, $\det(\mathbf{S}^T\mathbf{S}) = (\det(\mathbf{S}))^2$. Therefore, the smoothness measure can be simplified by $|\det(\mathbf{S})|$, and a smaller $|\det(\mathbf{S})|$ corresponds to a smoother $\mathbf{S}$.

Next, we present the scheme to get a smoother $\mathbf{S}$. Taking into account of the multilayer NMF decomposition technique [26], it is known that any nonnegative matrix $\mathbf{V} \approx \mathbf{WSH}$ can be further decomposed as

$$\mathbf{V} \approx \hat{\mathbf{W}}\mathbf{\Psi}\mathbf{S}\mathbf{\Phi}\hat{\mathbf{H}} \tag{13}$$

where all factors are nonnegative, and $\mathbf{\Psi}$ and $\mathbf{\Phi}$ are square and satisfy column-sum-to-one. Moreover, based on the lemma

in [34], it holds that $|\det(\mathbf{\Psi})| \leq 1$ and $|\det(\mathbf{\Phi})| \leq 1$. Then, it yields

$$\begin{aligned} |\det(\mathbf{S})| &\geq |\det(\mathbf{\Psi})||\det(\mathbf{S})||\det(\mathbf{\Phi})| \\ &= |\det(\mathbf{\Psi}\mathbf{S}\mathbf{\Phi})|. \end{aligned} \tag{14}$$

The above relation (14) implies that for a given $\mathbf{V}$, one can obtain a smoother $\mathbf{S}$ by absorbing some smoothness from its left neighbor $\mathbf{W}$ or right neighbor $\mathbf{H}$ or both of them. Furthermore, if $\mathbf{\Psi}$ (respectively $\mathbf{\Phi}$) is absorbed by $\mathbf{S}$, then $\mathbf{W}$ (respectively $\mathbf{H}$) becomes sparser $\hat{\mathbf{W}}$ (respectively $\hat{\mathbf{H}}$). As a result, this absorbing scheme enables us to generate sparse $\mathbf{W}$ and $\mathbf{H}$ separately.

As for how to obtain $\mathbf{\Phi}$ and $\mathbf{\Psi}$, it is analyzed as follows.

Case 1 (Obtaining $\mathbf{\Phi}$): Based on (13), $\mathbf{\Phi}$ is involved in

$$\mathbf{H} \approx \mathbf{\Phi}\hat{\mathbf{H}} \tag{15}$$

where $\mathbf{\Phi}$ satisfies column-sum-to-one. In general, there exist a lot of factors $\mathbf{\Phi}$ satisfying (15). Based on the smoothness measure, one can get a desired $\mathbf{\Phi}$ via solving the following model:

$$\text{Min } |\det(\mathbf{\Phi})| \tag{16}$$

subject to $\mathbf{\Phi}^{-1}\mathbf{H} \geq -\delta_{\mathbf{H}}$, $\Phi \geq 0$, $\sum_{i=1}^{r} \phi_{ij} = 1, \forall j$, where $\delta_{\mathbf{H}} \geq 0$ is a boundary parameter.

Case 2 (Obtaining $\mathbf{\Psi}$): Similar to (15), $\mathbf{\Psi}$ is involved in

$$\mathbf{W} \approx \hat{\mathbf{W}}\mathbf{\Psi} \tag{17}$$

where $\mathbf{\Psi}$ satisfies column-sum-to-one. A proper Ψ can be obtained via solving the following model:

$$\text{Min } |\det(\mathbf{\Psi})| \tag{18}$$

subject to $\mathbf{W}\mathbf{\Psi}^{-1} \geq -\delta_{\mathbf{W}}$, $\mathbf{\Psi} \geq 0$, $\sum_{i=1}^{r} \psi_{ij} = 1$, $\forall j$, where $\delta_{\mathbf{W}} \geq 0$ is another boundary parameter.

Finally, the smoothness constraint on $\mathbf{S}$ can be implemented based on the discussions in the above two cases, implying the accomplishment of the sparsity constraints on $\mathbf{W}$ and $\mathbf{H}$ in (12). In Section III-B, an effective algorithm will be derived to optimize (12), which will yield the desired products. The derivation of the algorithm relies on the following lemma about matrix expansion.

Lemma 1: Given any full rank matrix $\mathbf{A} \in \Re^{r\times r}$, for all $j \in \{1, 2, \ldots, r\}$, if only the elements of the jth column $a_{1j}, a_{2j}, \ldots, a_{rj}$ are unknown, then the following holds.

1) $\det(\mathbf{A})$ is a linear function with respect to $a_{1j}, a_{2j}, \ldots, a_{rj}$.
2) All entries of $\mathbf{A}^{-1}$ are linear functions with respect to $a_{1j}, a_{2j}, \ldots, a_{rj}$ if the common nonlinear factor $(1/\det(\mathbf{A}))$ is removed.

Proof: We first prove conclusion 1). Assume that $\mathbf{M}^{ij}$ is the submatrix of $\mathbf{A}$ with the ith row and the jth column removed. Considering the cofactor expansion of $\det(\mathbf{A})$ with respect to the jth column [35], we have

$$\det(\mathbf{A}) = \sum_{i=1}^{r} (-1)^{i+j} a_{ij} \det(\mathbf{M}^{ij}). \tag{19}$$

Since only $a_{1j}, a_{2j}, \ldots, a_{rj}$ are unknown, $\det(\mathbf{M}^{ij}), \forall i$ is a constant. Thus, $\det(\mathbf{A})$ is a linear function with respect to $a_{1j}, a_{2j}, \ldots, a_{rj}$. This completes the proof of conclusion 1).

Now, we prove conclusion 2). Let $\mathbf{A}^*$ be the adjoint matrix of $\mathbf{A}$. Then, we have

$$\mathbf{A}^{-1} = \frac{1}{\det(\mathbf{A})}\mathbf{A}^* \tag{20}$$

and the corresponding componentwise representation of $\mathbf{A}^{-1}$ is

$$[\mathbf{A}^{-1}]_{tk} = \frac{(-1)^{k+t}}{\det(\mathbf{A})} \det(\mathbf{M}^{kt}) \tag{21}$$

where $\mathbf{M}^{kt}$ denotes the submatrix of $\mathbf{A}$ with the kth row and the tth column removed. To further show the relationship between the entries of $\mathbf{A}^{-1}$ and the unknown entries $a_{1j}, a_{2j}, \ldots, a_{rj}$ of $\mathbf{A}$, we analyze $\det(\mathbf{M}^{kt}), \forall k, t$ by considering the following two scenarios.

1) If $t = j$, it is clear that for all k, $\det(\mathbf{M}^{kt})$ is independent of $a_{1j}, a_{2j}, \ldots, a_{rj}$.
2) If $t \neq j$, let

$$j_1 = \begin{cases} j-1, & \text{if } t < j \\ j, & \text{if } t > j. \end{cases} \tag{22}$$

Then, for all k, $\{[\mathbf{M}^{kt}]_{1j_1}, [\mathbf{M}^{kt}]_{2j_1}, \ldots, [\mathbf{M}^{kt}]_{(r-1)j_1}\}$ is an $r-1$ subset of $\{a_{1j}, a_{2j}, \ldots, a_{rj}\}$, and $\det(\mathbf{M}^{kt})$ can be factorized by cofactor expansion with respect to the j_1th column of $\mathbf{M}^{kt}$ as follows:

$$\det(\mathbf{M}^{kt}) = \sum_{i_1=1}^{r-1} [\mathbf{M}^{kt}]_{i_1 j_1} d^{kt}_{i_1 j_1} \quad \text{for } t \neq j \tag{23}$$

where $d^{kt}_{i_1 j_1}$ stands for the algebraic complement of $[\mathbf{M}^{kt}]_{i_1 j_1}$ and is independent of $a_{1j}, a_{2j}, \ldots, a_{rj}$.

Notably, for all $k, t \neq j$, $\{[\mathbf{M}^{kt}]_{1j_1}, [\mathbf{M}^{kt}]_{2j_1}, \ldots, [\mathbf{M}^{kt}]_{(r-1)j_1}\}$ in (23) is only related to $r-1$ elements out of the r elements $a_{1j}, a_{2j}, \ldots, a_{rj}$. To reflect all of the r elements $a_{1j}, a_{2j}, \ldots, a_{rj}$, we extend $\{[\mathbf{M}^{kt}]_{1j_1}, [\mathbf{M}^{kt}]_{2j_1}, \ldots, [\mathbf{M}^{kt}]_{(r-1)j_1}\}$ to $\{[\tilde{\mathbf{M}}^{kt}]_{1j_1}, [\tilde{\mathbf{M}}^{kt}]_{2j_1}, \ldots, [\tilde{\mathbf{M}}^{kt}]_{rj_1}\}$, that is

$$[\tilde{\mathbf{M}}^{kt}]_{i_1 j_1} = \begin{cases} [\mathbf{M}^{kt}]_{i_1 j_1} & \forall i_1 \in \{1, \ldots, k-1\} \\ a_{i_1 j}, & i_1 = k \\ [\mathbf{M}^{kt}]_{(i_1-1) j_1} & \forall i_1 \in \{k+1, \ldots, r\}. \end{cases} \tag{24}$$

Then, it holds that

$$[[\tilde{\mathbf{M}}^{kt}]_{1j_1}, [\tilde{\mathbf{M}}^{kt}]_{2j_1}, \ldots, [\tilde{\mathbf{M}}^{kt}]_{rj_1}]^T = [a_{1j}, a_{1j}, \ldots, a_{rj}]^T. \tag{25}$$

For consistency, we also extend $\{d^{kt}_{1j_1}, d^{kt}_{2j_1}, \ldots, d^{kt}_{(r-1)j_1}\}$ in (23) to $\{\tilde{d}^{kt}_{1j_1}, \tilde{d}^{kt}_{2j_1}, \ldots, \tilde{d}^{kt}_{rj_1}\}$, that is

$$\tilde{d}^{kt}_{i_1 j_1} = \begin{cases} d^{kt}_{i_1 j_1} & \forall i_1 \in \{1, \ldots, k-1\} \\ 0, & i_1 = k \\ d^{kt}_{(i_1-1) j_1} & \forall i_1 \in \{k+1, \ldots, r\}. \end{cases} \tag{26}$$

Based on (24)–(26), one can rewrite (23) as

$$\det(\mathbf{M}^{kt}) = \sum_{i_1=1}^{r} [\tilde{\mathbf{M}}^{kt}]_{i_1 j_1} \tilde{d}^{kt}_{i_1 j_1} = \sum_{i_1=1}^{r} a_{i_1 j} \tilde{d}^{kt}_{i_1 j_1} \quad \text{for } t \neq j. \tag{27}$$

From (21), (27), and the discussion in scenario 1), it follows that, for all k:

$$[\mathbf{A}^{-1}]_{tk} = \begin{cases} \dfrac{(-1)^{k+t}}{\det(\mathbf{A})} \displaystyle\sum_{i=1}^{r} a_{ij} \tilde{d}^{kt}_{i(j-1)} & \forall t < j \\ \dfrac{(-1)^{k+t}}{\det(\mathbf{A})} \det(\mathbf{M}^{kt}), & t = j \\ \dfrac{(-1)^{k+t}}{\det(\mathbf{A})} \displaystyle\sum_{i=1}^{r} a_{ij} \tilde{d}^{kt}_{ij} & \forall t > j \end{cases} \tag{28}$$

where $\det(\mathbf{M}^{kt})$ and $\tilde{d}^{kt}_{ij}$, $\forall i, j, k, t$ are independent of $a_{1j}, a_{2j}, \ldots, a_{rj}$. Therefore, for all k, t, $[\mathbf{A}^{-1}]_{tk}$ is a linear function with respect to $a_{1j}, a_{2j}, \ldots, a_{rj}$ if the common factor $[1/\det(\mathbf{A})]$ is removed. This completes the proof of conclusion 2). ■

B. Algorithm

Similar to the mainstream NMF-style methods, the well-known alternating iterative optimization scheme is utilized to solve (12). In particular, one can invoke (10) and (11) to update $\mathbf{H}$ and $\mathbf{W}$ directly if $\mathbf{S}$ is given. Notably, the explicit sparsity constraints on $\mathbf{H}$ and $\mathbf{W}$ will be implemented through the explicit smoothness constraint on $\mathbf{S}$. Hereafter, we mainly focus on the optimization of $\mathbf{S}$ by considering, to apply the sparsity constraint on $\mathbf{H}$ and $\mathbf{W}$, respectively.

1) Applying Sparsity Constraint on $\boldsymbol{H}$: Based on the analysis in Section III-A, the update of $\mathbf{S}$ in this case is implemented through absorbing an optimal $\mathbf{\Phi}$, which can be obtained by solving (16). We now present an optimization scheme for (16). Regarding the cost function, the determinant is highly nonlinear with respect to the entries of $\mathbf{\Phi}$, implying a big obstacle in the optimization. To deal with this issue, we utilize a column-by-column scheme, i.e., optimizing each column of $\mathbf{\Phi}$ alternatively while all other columns are fixed. Under this scheme, the determinant of $\mathbf{\Phi}$ in the cost function degenerates into an easier linear function with respect to each column based on Lemma 1. At the same time, the entries of $\mathbf{\Phi}^{-1}$ in the constraints become some kind of linear functions. As a result, (16) will fall into a much more solvable LP problem, if the involved common nonlinear factor in the constraints and the absolute in the cost function are removed. We shall further analyze how to remove them for the convenience of obtaining $\mathbf{\Phi}$.

With regard to the nonlinearity in the constraints of (16), clearly, it hides in the first inequality constraint $\mathbf{\Phi}^{-1}\mathbf{H} \succeq -\delta_{\mathbf{H}}$. Since $\delta_{\mathbf{H}}$ may not equal zero, removing the nonlinearity from the left-hand side of this inequality is difficult. Instead, we replace (16) with the following equivalent model[1]:

$$\text{Max } |\det(\mathbf{F})| \tag{29}$$

subject to $\mathbf{FH} \succeq -\delta_{\mathbf{H}}, \mathbf{F}^{-1} \succeq 0$, $\sum_{i=1}^{r} f_{ij} = 1$, where $\mathbf{F} = \mathbf{\Phi}^{-1}$.

Regarding (29), since $\mathbf{F} = \mathbf{I}$ is a possible solution, the solution space is not null. As for the optimization to $\mathbf{F}$, a column-by-column scheme will be utilized, then the mentioned nonlinearity exists in the entries of $\mathbf{F}^{-1}$ based on Lemma 1. Furthermore, the corresponding common nonlinear factor is $[1/\det(\mathbf{F})]$, as one can see from (28) after replacing $\mathbf{A}$ with $\mathbf{F}$. Since the right-hand side of the inequality constraint $\mathbf{F}^{-1} \succeq 0$ is zero, we can completely remove the nonlinearity from the constraints. So, the remaining problem is to identify the sign of the nonlinear factor $[1/\det(\mathbf{F})]$. As we know, during the optimization of each column of $\mathbf{F}$, the term on the right-hand side of the second equation in (28) is composed of a series of constants independent of that column, if $[1/\det(\mathbf{F})]$ is removed (where $\mathbf{A}$ is replaced by $\mathbf{F}$). Thus, to satisfy the nonnegativity constraint on this term, the sign of $[1/\det(\mathbf{F})]$ should keep the same, i.e., it should be always positive or negative, depending on the remaining constants. Considering that these constants are related to other entries of $\mathbf{F}$, we set the mentioned sign to be positive for the convenience of constructing the initial value of $\mathbf{F}$ (see the end of part 1) in Section III.B). Then, the nonlinear factor $[1/\det(\mathbf{F})]$ can be removed directly, without changing the corresponding inequalities.

It is worth noting that the absolute in the cost function in (29) can also be directly eliminated in the case that the sign of $[1/\det(\mathbf{F})]$ is positive. As a result, by replacing (16) with (29), both the nonlinearity in the constraints and the absolute in the cost function can be removed. Consequently, the complex optimization of (16) can be implemented by solving much easier LP problems related to the columns of $\mathbf{F}$ in (29). Without loss of generality, we discuss the LP problem with respect to the jth column of $\mathbf{F}$.[2] We start from the analysis of the constraints.

Regarding the first inequality constraint $\mathbf{FH} \succeq -\delta_{\mathbf{H}}$ of (29), its left-hand side can be written as

$$\mathbf{FH} = \sum_{i=1}^{j-1} \mathbf{f}_i \mathbf{h}^i + \mathbf{f}_j \mathbf{h}^j + \sum_{i=j+1}^{r} \mathbf{f}_i \mathbf{h}^i \tag{30}$$

where $\mathbf{f}_j$ and $\mathbf{h}^i$ denote the jth column and the ith row of $\mathbf{F}$ and $\mathbf{H}$, respectively. Hence, this inequality constraint is equivalent to

$$\mathbf{f}_j \mathbf{h}^j \succeq -\delta_{\mathbf{H}} - \sum_{i=1}^{j-1} \mathbf{f}_i \mathbf{h}^i - \sum_{i=j+1}^{r} \mathbf{f}_i \mathbf{h}^i \tag{31}$$

that is

$$f_{ij} \geq \max_{t \in \{t | h_{jt} \neq 0\}} \left\{ \frac{\left[-\delta_{\mathbf{H}} - \sum_{i=1}^{j-1} \mathbf{f}_i \mathbf{h}^i - \sum_{i=j+1}^{r} \mathbf{f}_i \mathbf{h}^i \right]_{it}}{h_{jt}} \right\}. \tag{32}$$

[1]Since $\mathbf{F\Phi} = \mathbf{I}$ and $\mathbf{\Phi F} = \mathbf{I}$, the equation constraints $\sum_{i=1}^{r} \phi_{ij} = 1, \forall j$ in (16) are equivalent to $\sum_{i=1}^{r} f_{ij} = 1, \forall j$ in this model.

[2]The problem structures corresponding to other columns are similar.

As for the second inequality constraint $\mathbf{F}^{-1} \geq 0$ of (29), we invoke some formulas from the proof of Lemma 1 directly. Denoting $\mathbf{M}^{kt}$ to be the submatrix of $\mathbf{F}$ with the kth row and the tth column removed, and d_{ij}^{kt} to be the algebraic component of $[\mathbf{M}^{kt}]_{ij}$, we construct $\tilde{d}_{ij}^{kt}$ by

$$\tilde{d}_{ij}^{kt} = \begin{cases} d_{ij}^{kt} & \forall i \in \{1, \ldots, k-1\} \\ 0, & i = k \\ d_{(i-1)j}^{kt} & \forall i \in \{k+1, \ldots, r\}. \end{cases} \tag{33}$$

Let $\tilde{\mathbf{d}}_j^{kt} = [\tilde{d}_{1j}^{kt}, \ldots, \tilde{d}_{rj}^{kt}]^T$. Then, based on (28) and the positivity of $[1/\det(\mathbf{F})]$, the constraint $\mathbf{F}^{-1} \geq 0$ can be simplified as

$$\begin{cases} (-1)^{k+t} \mathbf{f}_j^T \tilde{\mathbf{d}}_{j-1}^{kt} \geq 0 & \forall t < j \quad \forall k \\ (-1)^{k+t} \mathbf{f}_j^T \tilde{\mathbf{d}}_j^{kt} \geq 0 & \forall t > j \quad \forall k. \end{cases} \tag{34}$$

Finally, based on the above-derived constraints and the cofactor expansion of matrix determinant with respect to each column [35], the mentioned LP problem with respect to the jth column of $\mathbf{F}$ is as follows:

$$\text{Max} \ \det(\mathbf{F}) = \sum_{i=1}^{r} (-1)^{i+j} f_{ij} \det(\mathbf{M}^{ij}) \tag{35}$$

subject to

$$\begin{cases} (-1)^{k+t} \mathbf{f}_j^T \tilde{\mathbf{d}}_{j-1}^{kt} \geq 0 & \forall t < j \quad \forall k \\ (-1)^{k+t} \mathbf{f}_j^T \tilde{\mathbf{d}}_j^{kt} \geq 0 & \forall t > j \quad \forall k \\ \mathbf{f}_j^T \mathbf{1} = 1 \\ \mathbf{f}_j \mathbf{h}^j \geq -\delta_{\mathbf{H}} - \sum_{i=1}^{j-1} \mathbf{f}_i \mathbf{h}^i - \sum_{i=j+1}^{r} \mathbf{f}_i \mathbf{h}^i. \end{cases}$$

For solving (35), one can directly use the existing interior point methods [36], [37]. Here, the MATLAB toolbox about the LP problem is invoked, where the initial value of $\mathbf{F}$ is set as the identity matrix to ensure the positivity of $[1/\det(\mathbf{F})]$. After $\mathbf{F}$ is obtained, $\mathbf{\Phi}$ in (16) can be calculated by

$$\mathbf{\Phi} = \mathbf{F}^{-1}. \tag{36}$$

2) Applying Sparsity Constraint on $\mathbf{W}$: In this case, the update of $\mathbf{S}$ is implemented through absorbing an optimal $\mathbf{\Psi}$, which can be obtained from solving (18). Similar to the optimization scheme presented in Section III-B1, we replace (18) with

$$\text{Max} \ |\det(\mathbf{G})| \tag{37}$$

subject to $\mathbf{WG} \geq -\delta_{\mathbf{W}}$, $\mathbf{G}^{-1} \geq 0$, $\sum_{i=1}^{r} g_{ij} = 1$, where $\mathbf{G} = \mathbf{\Psi}^{-1}$. Furthermore, convert it to the following LP problem with respect to the jth column of $\mathbf{G}$:

$$\text{Max} \ \det(\mathbf{G}) = \sum_{i=1}^{r} (-1)^{i+j} g_{ij} \det(\bar{\mathbf{M}}^{ij}) \tag{38}$$

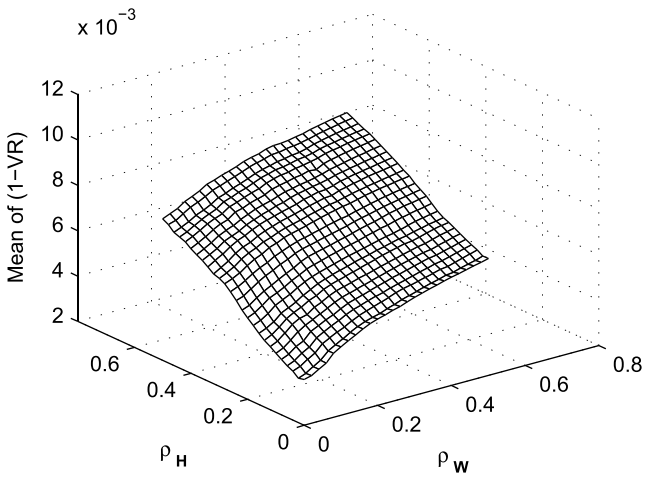


Fig. 1. Mean of VR index under different values of $\rho_{\mathbf{W}}$ and $\rho_{\mathbf{H}}$, where VR is replaced with $1 - \text{VR}$ for better visualization.

subject to

$$\begin{cases} (-1)^{k+t} \mathbf{g}_j^T \bar{\mathbf{d}}_{j-1}^{kt} \geq 0 & \forall t < j \quad \forall k \\ (-1)^{k+t} \mathbf{g}_j^T \bar{\mathbf{d}}_j^{kt} \geq 0 & \forall t > j \quad \forall k \\ \mathbf{g}_j^T \mathbf{1} = 1 \\ \mathbf{W} \mathbf{g}_j \geq -\delta_{\mathbf{W}} \end{cases}$$

where $\bar{\mathbf{M}}^{ij}$ is the submatrix of $\mathbf{G}$ with the ith row and the jth column removed, and $\mathbf{g}_j$ and $\bar{\mathbf{d}}_j^{kt}$ are similar to $\mathbf{f}_j$ and $\tilde{\mathbf{d}}_j^{kt}$ in (34), respectively.

Clearly, the optimization scheme used for (35) can also be employed to solve (38) here, and one only needs to change the parameters of the corresponding LP function in MATLAB. After $\mathbf{G}$ is obtained, $\mathbf{\Psi}$ in (18) can be calculated by

$$\mathbf{\Psi} = \mathbf{G}^{-1}. \tag{39}$$

3) Summary of Algorithm: Based on the analysis in Sections III-B1 and III-B2, the proposed Ans-NMF algorithm for optimizing (12) is summarized as follows.

1) *Initialization:* Randomly generate the two positive matrices $\mathbf{W}$ and $\mathbf{H}$, and set $\mathbf{S} = \mathbf{I}$.
2) *Iteration:*
   a) Update $\mathbf{H}$ using (10). If a sparsity constraint on $\mathbf{H}$ applies, set a value for $\delta_{\mathbf{H}}$, which is related to the degree of the sparsity constraint. Then, calculate $\mathbf{\Phi}$ using (35) and (36), and update $\mathbf{H}$ and $\mathbf{S}$ by
   $$\begin{cases} \mathbf{H} := \max(\mathbf{\Phi}^{-1}\mathbf{H}, \mathbf{0}) \\ \mathbf{S} := \mathbf{S}\mathbf{\Phi}. \end{cases} \tag{40}$$
   b) Update $\mathbf{W}$ using (11). If sparsity constraint on $\mathbf{W}$ applies, such as a), set a value for $\delta_{\mathbf{W}}$, calculating $\mathbf{\Psi}$ using (38) and (39), and update $\mathbf{W}$ and $\mathbf{S}$ by
   $$\begin{cases} \mathbf{W} := \max(\mathbf{W}\mathbf{\Psi}^{-1}, \mathbf{0}) \\ \mathbf{W} := \mathbf{W}(\text{diag}(1 \oslash \tilde{\mathbf{w}})) \\ \mathbf{S} := \mathbf{\Psi}\mathbf{S}. \end{cases} \tag{41}$$

In the algorithm above, the function $\max(\mathbf{A}, \mathbf{B})$ returns a matrix $\mathbf{C}$ with the same size of $\mathbf{A}$ and $\mathbf{B}$, where each

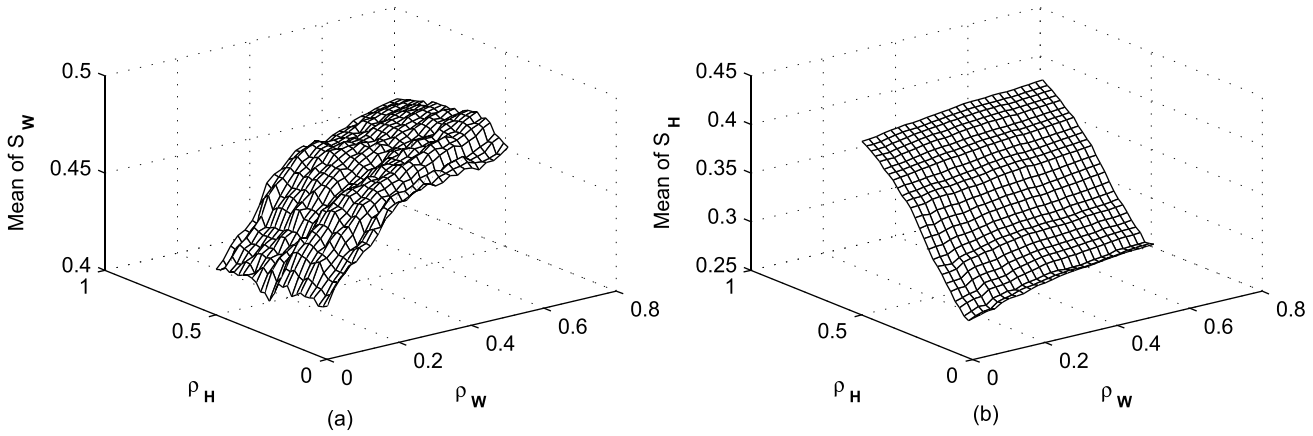


Fig. 2. Means of $S_{\mathbf{W}}$ and $S_{\mathbf{H}}$ indices under different values of $\rho_{\mathbf{W}}$ and $\rho_{\mathbf{H}}$. (a) Mean of $S_{\mathbf{W}}$. (b) Mean of $S_{\mathbf{H}}$.

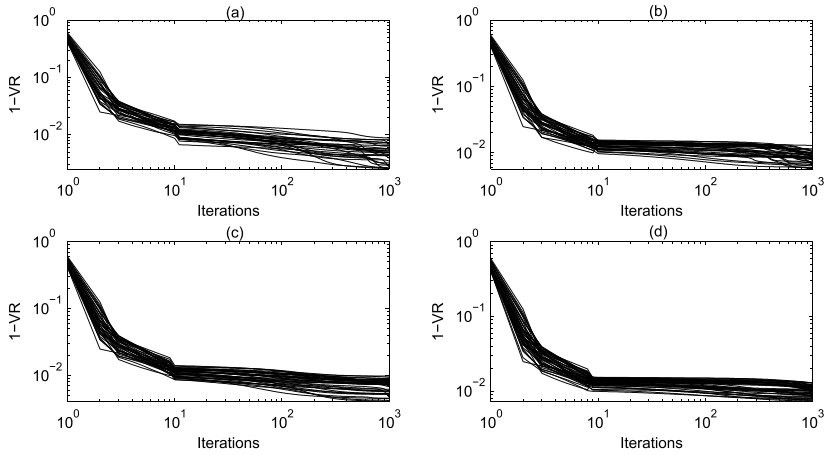


Fig. 3. Values of $1 - \mathrm{VR}$ versus the number of iterations under four typical parameter pairs $\rho_{\mathbf{H}}$ and $\rho_{\mathbf{W}}$, where the axes are logarithmic. (a) $\rho_{\mathbf{H}} = 0$ and $\rho_{\mathbf{W}} = 0$. (b) $\rho_{\mathbf{H}} = 0.45$ and $\rho_{\mathbf{W}} = 0$. (c) $\rho_{\mathbf{H}} = 0$ and $\rho_{\mathbf{W}} = 0.45$. (d) $\rho_{\mathbf{H}} = 0.45$ and $\rho_{\mathbf{W}} = 0.45$.

entry $c_{ij}, \forall i, j$ takes the maximum of a_{ij} and b_{ij}. Since the values of $\mathbf{\Phi}$ and $\mathbf{\Psi}$ are related to the boundary parameters $\delta_{\mathbf{H}}$ and $\delta_{\mathbf{W}}$, for the convenience of implementation, we replace $\delta_{\mathbf{H}}$ and $\delta_{\mathbf{W}}$ by the ratios $\rho_{\mathbf{H}}$ and $\rho_{\mathbf{W}}$ to the maximum values of the entries of $\mathbf{H}$ and $\mathbf{W}$, respectively, i.e., $\delta_{\mathbf{H}} = \max(\mathbf{H})\rho_{\mathbf{H}}$, $\delta_{\mathbf{W}} = \max(\mathbf{W})\rho_{\mathbf{W}}$, where $\rho_{\mathbf{H}}, \rho_{\mathbf{W}} \in [0, 1)$.

4) Convergence and Computational Complexity: Regarding the convergence, the proposed algorithm is based on the well-known alternatively multiplicative update scheme, which has been theoretically analyzed in [2] and [24]. Actually, in the case that $\mathbf{S}$ is fixed, the convergence property in updating $\mathbf{H}$ and $\mathbf{W}$ by (10) and (11) is the same to that of the existing Ns-NMF in [23]. As for the update of $\mathbf{S}$, it needs to solve a series of LP problems, which are convergent. Thus, we infer that the convergence of our Ans-NMF behaves similar to that of Ns-NMF.

As far as the computational complexity is concerned, the main time cost of the proposed algorithm is spent on the computation of $\mathbf{F}$ by solving (35) and $\mathbf{G}$ by solving (38). It is known that using a primal–dual interior point method, each LP problem [or the problem in (35)] can be solved with a worst case complexity of $O(n^{0.5}(n(r-1)+(r-1)^3)) \simeq O(n^{1.5}(r-1))$ for $n \gg r$ [37]. Since the algorithm needs to solve r LP problems to get $\mathbf{F}$ by solving (35), it can be concluded that its worst case complexity is $O(n^{1.5}r^2)$ approximately. Note that the factor $n^{0.5}$ is the theoretical worst case number of iterations for an interior point optimization algorithm to solve an LP. Practically, it is much smaller and seems like a constant [37]. Hence, to get $\mathbf{F}$, our algorithm works more like $O(n^q r^2)$ where $q \in [1, 1.5]$ is closer to one. Similarly, to get $\mathbf{G}$, the computational complexity is $O(m^q r^2)$. Furthermore, in updating (10) and (11), it needs the complexity of $O(mrn + 2mr^2 + 2nr^2)$. In addition, to solve (36) and (39)–(41), the computational complexities are $O(r^3)$, $O(r^3)$, $O(r^2 n + r^3)$, and $O(2r^2 m + r^3)$, respectively. Thus, in each iteration, the computational complexity of the proposed algorithm is the summation of the complexities above, i.e., $O(n^q r^2 + m^q r^2 + mrn + 2mr^2 + 2nr^2 + r^3 + r^3 + r^2 n + r^3 + 2r^2 m + r^3) \simeq O((n^q + m^q)r^2 + mrn)$.

In comparison, in each iteration, the computational complexity of NMF [5], new efficient nonnegative matrix factorization (Ne-NMF) [4], Ns-NMF [23], and Sc-NMF [15] are $O(mrn+mr^2+nr^2)$, $O(mr^2+mrn+nr^2)+K \times O(r^2 n + r^2 m)$, $O(mrn + 2mr^2 + 2nr^2)$, and $O(Kmrn)$, respectively, where K denotes the number of inner iterations.

IV. Simulations

In this section, we use both computer-generated data and real-world data to illustrate the performance of the proposed

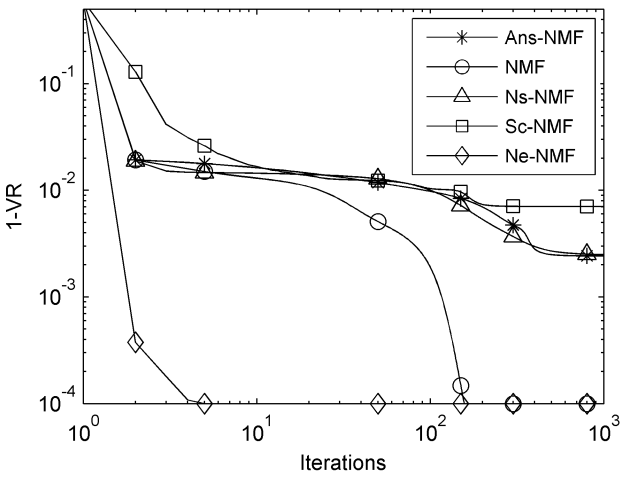


Fig. 4. Values of $1-\text{VR}$ versus the number of iterations using the compared algorithms.

Ans-NMF method, in terms of faithfulness and interpretability. We also compare the performance of our method with that of the traditional NMF method [5], the newly developed Ne-NMF method [4], the Ns-NMF method based on explicit smoothness constraint (with smoothness parameter θ) [23], and the Sc-NMF method using explicit sparsity constraint (with constraint parameters $\alpha_{\mathbf{W}}$ and $\alpha_{\mathbf{H}}$ for $\mathbf{W}$ and $\mathbf{H}$, respectively) [15]. Each method is implemented using MATLAB R2011b installed in a personal computer with Intel(R) Celeron(R) 2.4-GHz CPU, 2-GB memory, and the Microsoft Windows 7 operational system.

The faithfulness performance is measured by the variance ratio (VR) index, defined by

$$\text{VR} = \frac{\|\mathbf{V}\|_F^2 - \|\mathbf{V} - \mathbf{WSH}\|_F^2}{\|\mathbf{V}\|_F^2}. \tag{42}$$

Clearly, the larger the index VR, the better the faithfulness performance. As for the interpretability, it is application-dependent,[3] and thus there is generally no unified measure for numerical comparison. However, when it refers to the sparseness, we use the following measure, which is directly derived from Hoyer's method [15]:

$$S_{\mathbf{X}} = \frac{\sqrt{mr} - \left(\sum_{i=1}^{m}\sum_{j=1}^{r} x_{ij}\right) \Big/ \left(\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{r} x_{ij}^2}\right)}{\sqrt{mr} - 1} \tag{43}$$

where $\mathbf{X} \in \Re^{m\times r}$ is nonnegative. The larger the index $S_{\mathbf{X}}$, the sparser the matrix $\mathbf{X}$.

A. Computer-Generated Data

In this subsection, the computer-generated data are used to evaluate the performance of the proposed method. Similar to [23], small data are utilized for the convenience of computation. The data matrix $\mathbf{V}$ is generated by multiplying two rank-3 matrices in $\Re^{6\times 3}$ and $\Re^{3\times 20}$ (i.e., $m = 6$ and $n = 20$), respectively, whose entries are uniformly distributed within [0, 1]. Obviously, this data set can be perfectly decomposed into two nonnegative products, and the sparseness levels of the underlying products are low. To get a highly faithful representation, the number of the features is set to be three (i.e., $r = 3$).

[3]In different applications, the learned results may be required to have different features, such as sparsity, locality, oriented style, and so on.

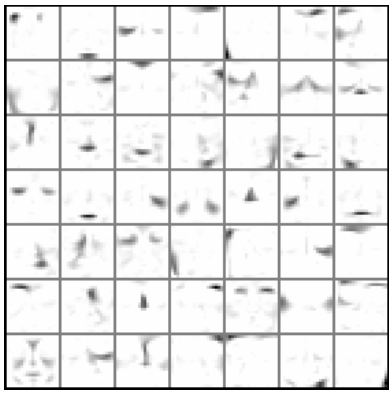


Fig. 5. Features extracted from the CBCL data set by using the standard NMF.

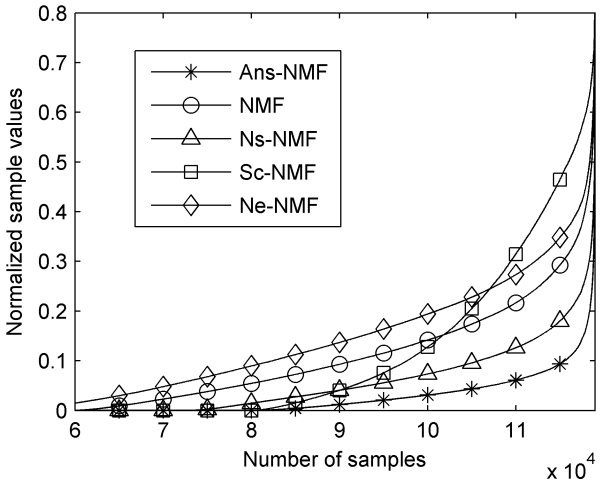


Fig. 6. Normalized coefficients corresponding to the features extracted from the CBCL data set by different methods.

We first test the performance of the proposed Ans-NMF against the parameter pair of $\rho_{\mathbf{W}}$ and $\rho_{\mathbf{H}}$, both of which vary from 0 to 0.5. For each pair of $\rho_{\mathbf{W}}$ and $\rho_{\mathbf{H}}$, 50 independent runs are performed. Fig. 1 shows the mean of the VR index under different values of $\rho_{\mathbf{W}}$ and $\rho_{\mathbf{H}}$, where VR is replaced by $1 - \text{VR}$ for better visualization. One can see that the satisfactory faithfulness that has been achieved as VR is close to 100% (or $1-\text{VR}$ is close to 0). The means of $S_{\mathbf{W}}$ and $S_{\mathbf{H}}$ are shown in Fig. 2(a) and (b), respectively. From Fig. 2(a), one can see that the sparseness of $\mathbf{W}$ enhances with the increase of $\rho_{\mathbf{W}}$ but it is much less affected by $\rho_{\mathbf{H}}$. Meanwhile, it is shown in Fig. 2(b) that the sparseness of $\mathbf{H}$ is almost completely determined by $\rho_{\mathbf{H}}$. This implies that the sparseness of the products can be separately constrained in our method.

Then, we test the convergence of Ans-NMF, under four different parameter pairs $\rho_{\mathbf{H}}$ and $\rho_{\mathbf{W}}$. For each pair of $\{\rho_{\mathbf{H}}, \rho_{\mathbf{W}}\}$, 50 independent runs are carried out, where a different random initial value is used in each run. Fig. 3 shows the VR indices versus the number of iterations under $\{\rho_{\mathbf{H}} = 0, \rho_{\mathbf{W}} = 0\}$,

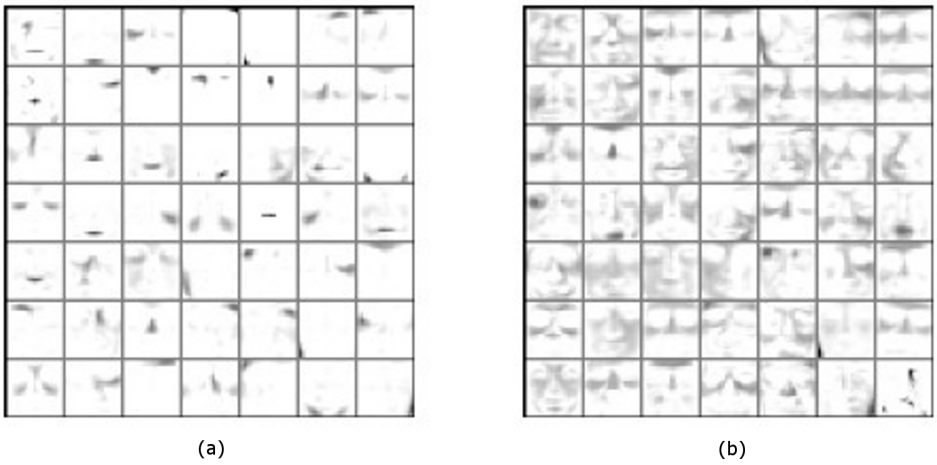


Fig. 7. Local and nonlocal features extracted from the CBCL data set by using Ans-NMF. (a) Local features under $\rho_{\mathbf{W}} = 0.5$ and $\rho_{\mathbf{H}} = 0$. (b) Nonlocal features under $\rho_{\mathbf{W}} = 0$ and $\rho_{\mathbf{H}} = 0.5$.

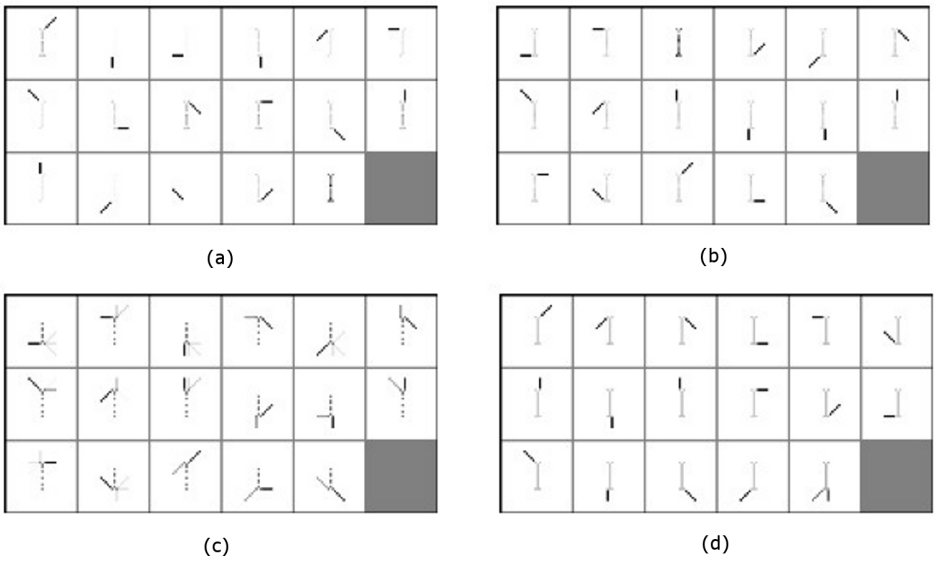


Fig. 8. Features extracted from the swimmer data set by using (a) Ans-NMF, (b) Ns-NMF, (c) Sc-NMF, and (d) Ne-NMF.

$\{\rho_{\mathbf{H}} = 0.45, \rho_{\mathbf{W}} = 0\}$, $\{\rho_{\mathbf{H}} = 0, \rho_{\mathbf{W}} = 0.45\}$, and $\{\rho_{\mathbf{H}} = 0.45, \rho_{\mathbf{W}} = 0.45\}$, respectively. It can be seen that the proposed algorithm converges in all considered scenarios.

For comparison, the convergence curves of the benchmark algorithms NMF, Ns-NMF, Sc-NMF, Ne-NMF, and our Ans-NMF are shown in Fig. 4. Since there is no sparseness constraint in NMF and Ne-NMF, they have better convergence performance than the other algorithms. Among the methods with constraints, our Ans-NMF is comparable to Ns-NMF, and better than Sc-NMF.

Furthermore, we compare the results obtained by Ans-NMF, NMF, Ns-NMF, Sc-NMF, and Ne-NMF. The 50 independent runs are performed for each method. Table II shows the means of VR, $S_{\mathbf{W}}$, $S_{\mathbf{H}}$, and S_{avg}, where S_{avg} is defined as $S_{\text{avg}} = (S_{\mathbf{W}} + S_{\mathbf{H}})/2$. As we can see from Table II, while NMF and Ne-NMF yield high faithfulness (with VR > 99%), they also produce the lowest sparseness (with the smallest $S_{\mathbf{W}}$, $S_{\mathbf{H}}$, and S_{avg}). Among the other three methods, Ns-NMF and Ans-NMF result in much larger VR values than Sc-NMF, showing the great potential of embedding smoothness constraint to faithful sparse decompositions. It is worth noting that our Ans-NMF generates sparser products than Ns-NMF, besides the similar high faithfulness. The reason is that the embedding smoothness factor is adaptively learned from the data set in Ans-NMF, rather than manually created as in Ns-NMF.

TABLE II
MEANS OF VR, $S_{\mathbf{W}}$, $S_{\mathbf{H}}$, AND S_{avg} OBTAINED BY DIFFERENT METHODS

	Ans-NMF	NMF	Ns-NMF	Sc-NMF	Ne-NMF
VR (%)	99.00	99.99	98.99	92.02	99.99
$S_{\mathbf{W}}$	0.4890	0.1700	0.3968	0.4322	0.1939
$S_{\mathbf{H}}$	0.4364	0.1698	0.3894	0.3566	0.1707
S_{avg}	0.4627	0.1699	0.3931	0.3944	0.1823

B. Face Data

The real-world center for biological and computational learning face data set is utilized in this group of simulations, which can be downloaded from [38]. It contains 2429 19×19

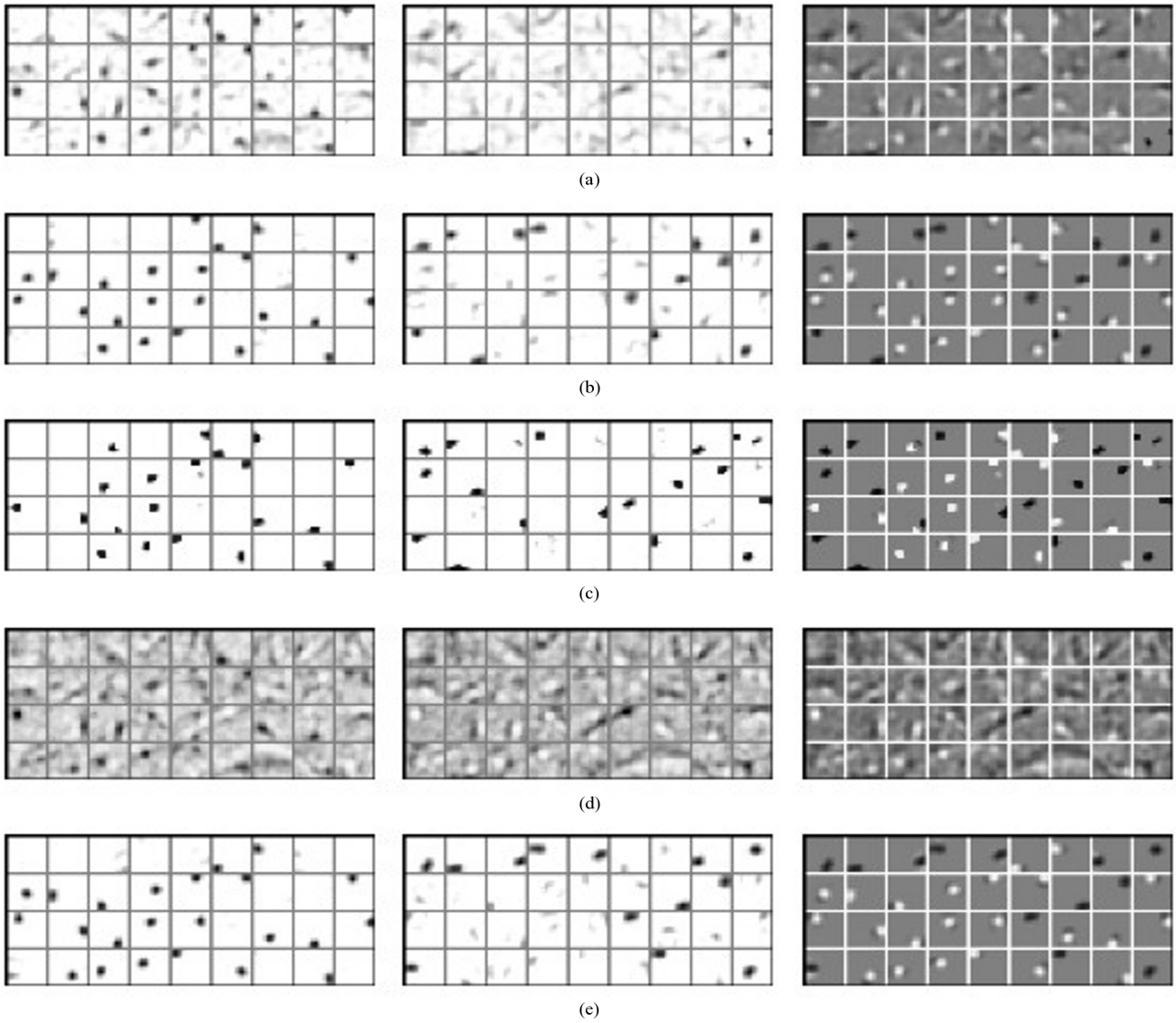


Fig. 9. Features extracted from ON/OFF-filtered natural image data set by (a) Ans-NMF, (b) NMF, (c) Ns-NMF, (d) Sc-NMF, and (e) Ne-NMF. From left to right, the features are for ON-channel, OFF-channel, and ON-channel minus OFF-channel, respectively.

facial low-resolution gray-level images (i.e., $m = 381$ and $n = 2429$). The same data set has been used in [5] to present the capacity of the standard NMF method to produce a part-based (or local) representation of the facial images. The learned features by the standard NMF are shown in Fig. 5, where the corresponding VR value is 96.01%. For comparison purpose, the same feature number $r = 49$ is used in our simulations.

First, we compare the sparsity of the results learned by Ans-NMF, NMF, Ns-NMF, Sc-NMF, and Ne-NMF, under similar faithfulness, i.e., 94.62%, 94.65%, 94.18%, 93.87%, and 94.69%, respectively. Here, we focus on the sparseness of the learned coefficients, which reflect the number of features needed for reconstructing an input image. As the sparseness of the coefficient matrix is not affected by the permutation and scaling of the entries, each of the matrices is reshaped into a vector, in which the elements are sorted with ascending order and then normalized into [0, 1]. The normalized coefficients are shown in Fig. 6, where the first 60 000 samples with very small values are removed for better visual comparison, and the values above 0.8 are discarded for the same reason. In addition, the corresponding sparseness degrees are 0.6320, 0.4764, 0.5489, 0.5595, and 0.4282. It shows that the proposed method generates the sparsest coefficient, implying that our method has great potential to explaining the input image with a minimum number of features.

Then, we show that Ans-NMF can generate both local and nonlocal representations for this data set, more flexible than the standard NMF. These variant representations or features can be obtained through adjusting the parameter pair $\rho_{\mathbf{W}}$ and $\rho_{\mathbf{H}}$. Fig. 7(a) and (b) shows the learned features under the parameter pairs $\rho_{\mathbf{W}} = 0.5, \rho_{\mathbf{H}} = 0$ and $\rho_{\mathbf{W}} = 0, \rho_{\mathbf{H}} = 0.5$, respectively. It can be seen that the proposed Ans-NMF has the capacity of achieving both local and nonlocal feature extractions. Interestingly, the nonlocal features show some similarity to those given by vector quantization [5], implying a strong interpretability of our method.

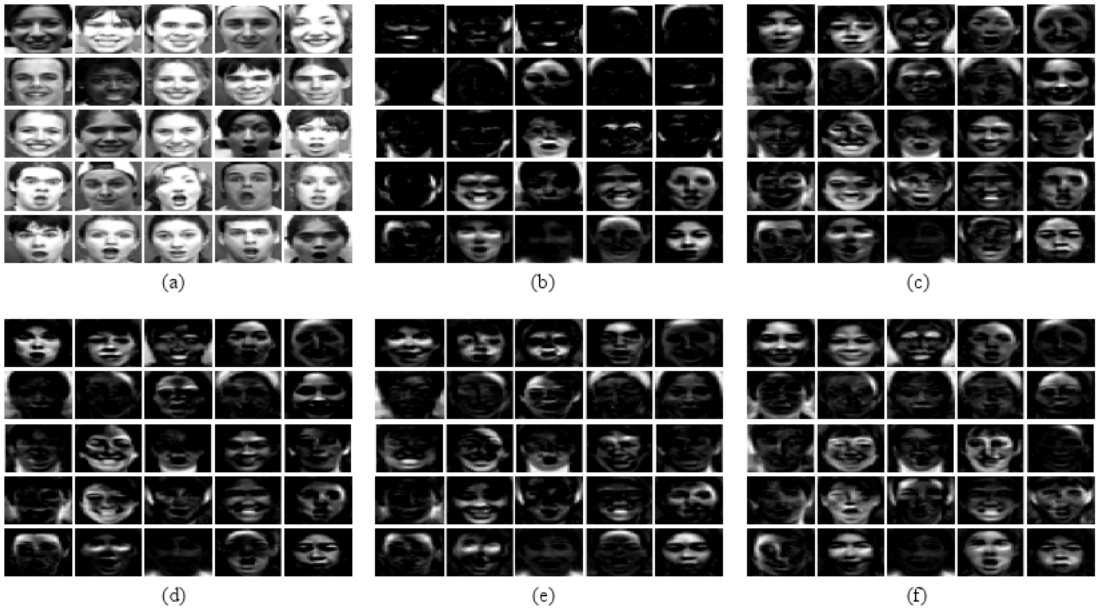


Fig. 10. (a) 25 initial happiness and surprise images. 25 basis images learned by (b) Ans-NMF, (c) NMF, (d) Ns-NMF, (e) Sc-NMF, and (f) Ne-NMF.

C. Swimmer Data

The performance of the considered methods is tested using the swimmer data set, which was created by an NMF-style generative model obeying some predefined rules [39]. It consists of a set of black-and-white images with one fixed part (torso) and four moving parts (limbs). Each moving part can exhibit four different positions (articulations). In total, there are 256 images of dimension 32×32 pixels (i.e., $m = 1024$ and $n = 256$), containing all possible limb positions/combinations. As in [23], 17 factors (i.e., $r = 17$) are used in our simulations. Since the simulation results in [23] have shown that Ns-NMF performs better than NMF, the latter is not compared here. That is, we compare the proposed Ans-NMF with Ns-NMF, Sc-NMF, and Ne-NMF.

Fig. 8 shows the best results (according to the faithfulness) of the compared methods selected from ten independent runs, each with random initializations. One can see from Fig. 8(a) and (b) that the smoothness-based methods Ans-NMF and Ns-NMF successfully extract all 16 limbs and the torso, showing a high interpretability. In contrast, Sc-NMF and Ne-NMF fail to extract all limbs and the torso, as shown in Fig. 8(c) and (d). Furthermore, for Ans-NMF and Ns-NMF, the corresponding VR values are 99.33 and 94.54, respectively. So, Ans-NMF gives a better faithfulness than Ns-NMF. This verifies the superiority of the adaptive selection of the smoothness factor.

D. Natural Images

In this simulation, we compare the methods of interest using another set of real-world data containing 10 000 12×12 natural images (i.e., $m = 144$ and $n = 10\,000$). Similar to the processing of visual information by the retina [14], [15], the image patches are high-pass filtered and subsequently split into positive (ON) and negative (OFF) contrast channels. As we know, in most of the modern image processing techniques, the cornerstone consists of oriented filters. This oriented style is regarded as the best type of low-level features for representing natural images, and also widely used in biological brain visual systems [40], [41]. To compare the interpretability of the results, we mainly analyze the shape of the features (with feature number $r = 36$ detected using the method in [42]) learned by Ans-NMF, NMF, Ns-NMF, Sc-NMF, and Ne-NMF.

The features extracted by these methods are shown in Fig. 9. As we can see, the proposed Ans-NMF method generates oriented, Gaborlike features, similar to the Sc-NMF method. Furthermore, the features yielded by Ans-NMF are much sparser than those resulted from Sc-NMF [see Fig. 9(a) and (d)]. In contrast, NMF and Ne-NMF represent these images using simple, dull, and circular blobs [see Fig. 9(b) and (e)]. In addition, based on Fig. 9(b) and (c), the features produced by Ns-NMF are similar to those given by the standard NMF. The reason is that Ns-NMF cannot separately sparsify the products, which restricts the interpretability of the learned results.

E. Cohn–Kanade Data

In this simulation, the proposed method is applied to image classification for facial expression recognition, and we aim at the recognition of two universal expressions: happiness and surprise. The real-world Cohn-Kanade database released in 2000 is tested. This database consists of 486 image sequences from 97 university students aged between 18 and 30 years, where 65% are female students, 15% are African–American, and 3% are from Asia or Latin American. Each image has 30×40 pixels, i.e., the dimensionality of the initial data is $m = 1200$. Fig. 10(a) shows 25 random initial images, where 13 are with happiness expression and 12 are with surprise expression.

In our experiments, the data dimensionality is reduced to $r = 25$ by using the algorithms Ans-NMF, NMF, Ns-NMF,

TABLE III
CLASSIFICATION ACCURACY (%) OF Ans-NMF+SVM, NMF+SVM, Ns-NMF+SVM, Sc-NMF+SVM, AND Ne-NMF+SVM ALGORITHMS FOR THE COHN–KANADE DATABASE

Ans-NMF+SVM	NMF+SVM	Ns-NMF+SVM	Sc-NMF+SVM	Ne-NMF+SVM
92.65%	90.95%	90.36%	90.39%	92.06%

Sc-NMF, and Ne-NMF. Fig. 10(b)–(f) shows the 25 features learned by these algorithms, respectively. It can be seen that the features obtained by our Ans-NMF are the sparsest. This property is very useful in information extraction, code, compression, and so on. For classification, the support vector machine (SVM) method is applied to the low-dimensional data, where the fivefold cross-validation scheme is utilized in the classification, i.e., each class data are partitioned into five complementary subsets, and in each iteration (of the total five), one subset is left for testing and the others are used for training. Finally, the classification accuracy of each compared algorithm is obtained by averaging the accuracies in each iteration. Table III gives the classification accuracies (%) of these algorithms in combination with SVM. It shows that our method obtains the greatest accuracy index, implying that it is the most effective one among the compared methods.

V. CONCLUSION AND DISCUSSION

Due to its ability in extracting human intelligible features, NMF has attracted much attention over the past few years. However, to generate desired results, it is appealing to explore some advanced schemes with explicit constraints for different scenarios. In this paper, an Ans-NMF method is proposed. It aims to yield desired results for particular applications, in which either the feature or the coefficient (or both) satisfies sparsity requirement, and the decomposition error is as small as possible. Simulation results based on both computer-generated data and real-world data show that the proposed Ans-NMF method is capable of achieving this goal and performs better than other NMF methods with explicit constraints.

Since the proposed method can generate sparse results, its extension has great potential to the promotion of group sparse code (GSC). This code scheme is proposed in [43], under the assumption that the data groups are predefined. In [44], the assumption is relaxed, and the data groups are learned automatically. By employing the spatial information, the extension of our method could further improve GSC in terms of better interpretability and smaller code error. It is also an interesting study to extend the proposed method to semisupervised or supervised learning, and the nonoverlapping property of the results is particularly useful for clustering [45] and source separation [46].

REFERENCES

[1] W. Ren, G. Li, D. Tu, and L. Jia, "Nonnegative matrix factorization with regularizations," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 4, no. 1, pp. 153–164, Mar. 2014.

[2] L.-X. Li, L. Wu, H.-S. Zhang, and F.-X. Wu, "A fast algorithm for nonnegative matrix factorization and its convergence," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 10, pp. 1855–1863, Oct. 2014.

[3] K. Huang, N. D. Sidiropoulos, and A. Swami, "Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition," *IEEE Trans. Signal Process.*, vol. 62, no. 1, pp. 211–224, Jan. 2014.

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

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

[6] Y.-X. Wang and Y.-J. Zhang, "Nonnegative matrix factorization: A comprehensive review," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 6, pp. 1336–1353, Jun. 2013.

[7] Z. Zhang and K. Zhao, "Low-rank matrix approximation with manifold regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 7, pp. 1717–1729, Jul. 2013.

[8] G. Zhou, A. Cichocki, and S. Xie, "Fast nonnegative matrix/tensor factorization based on low-rank approximation," *IEEE Trans. Signal Process.*, vol. 60, no. 6, pp. 2928–2940, Jun. 2012.

[9] N. Wang, B. Du, and L. Zhang, "An endmember dissimilarity constrained non-negative matrix factorization method for hyperspectral unmixing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 2, pp. 554–569, Apr. 2013.

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

[11] X. Pei, T. Wu, and C. Chen, "Automated graph regularized projective nonnegative matrix factorization for document clustering," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1821–1831, Oct. 2014.

[12] N. Yokoya, J. Chanussot, and A. Iwasaki, "Nonlinear unmixing of hyperspectral data using semi-nonnegative matrix factorization," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 2, pp. 1430–1437, Feb. 2014.

[13] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 13. 2001, pp. 556–562.

[14] P. O. Hoyer, "Modeling receptive fields with non-negative sparse coding," *Neurocomputing*, vols. 52–54, pp. 547–552, Jun. 2003.

[15] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraint," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, Nov. 2004.

[16] Z. Yang, G. Zhou, S. Xie, S. Ding, J.-M. Yang, and J. Zhang, "Blind spectral unmixing based on sparse nonnegative matrix factorization," *IEEE Trans. Image Process.*, vol. 20, no. 4, pp. 1112–1125, Apr. 2011.

[17] Z. He, S. Xie, S. Ding, and A. Cichocki, "Convolutive blind source separation in the frequency domain based on sparse representation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 5, pp. 1551–1563, Jul. 2007.

[18] O. Zoidi, A. Tefas, and I. Pitas, "Multiplicative update rules for concurrent nonnegative matrix factorization and maximum margin classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 3, pp. 422–434, Mar. 2013.

[19] H. Liu, Z. Wu, X. Li, D. Cai, and T. S. Huang, "Constrained nonnegative matrix factorization for image representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1299–1311, Jul. 2012.

[20] G. Zhou, Z. Yang, S. Xie, and J.-M. Yang, "Online blind source separation using incremental nonnegative matrix factorization with volume constraint," *IEEE Trans. Neural Netw.*, vol. 22, no. 4, pp. 550–560, Apr. 2011.

[21] 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*. Oxford, U.K.: Wiley, 2009.

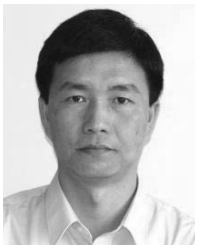
[22] Z. Yang, Y. Xiang, Y. Rong, and S. Xie, "Projection-pursuit-based method for blind separation of nonnegative sources," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 1, pp. 47–57, Jan. 2013.

[23] A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmann, and R. D. Pascual-Marqui, "Nonsmooth nonnegative matrix factorization (nsNMF)," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 3, pp. 403–415, Mar. 2006.
[24] R. Badeau, N. Bertin, and E. Vincent, "Stability analysis of multiplicative update algorithms and application to nonnegative matrix factorization," *IEEE Trans. Neural Netw.*, vol. 21, no. 12, pp. 1869–1881, Dec. 2010.
[25] N. Guan, D. Tao, Z. Luo, and B. Yuan, "Online nonnegative matrix factorization with robust stochastic approximation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1087–1099, Jul. 2012.
[26] A. Cichocki and R. Zdunek, "Multilayer nonnegative matrix factorisation," *Electron. Lett.*, vol. 42, no. 16, pp. 947–948, Aug. 2006.
[27] C. Ding, T. Li, and M. I. Jordan, "Convex and semi-nonnegative matrix factorizations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 45–55, Jan. 2010.
[28] Z. He, S. Xie, R. Zdunek, G. Zhou, and A. Cichocki, "Symmetric nonnegative matrix factorization: Algorithms and applications to probabilistic clustering," *IEEE Trans. Neural Netw.*, vol. 22, no. 12, pp. 2117–2131, Dec. 2011.
[29] B. Li, G. Zhou, and A. Cichocki, "Two efficient algorithms for approximately orthogonal nonnegative matrix factorization," *IEEE Signal Process. Lett.*, vol. 22, no. 7, pp. 843–846, Jul. 2015.
[30] D. Kong, C. Ding, and H. Huang, "Robust nonnegative matrix factorization using L21-norm," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, Glasgow, Scotland, Oct. 2011, pp. 673–682.
[31] B. Gao, W. L. Woo, and S. S. Dlay, "Variational regularized 2-D nonnegative matrix factorization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 5, pp. 703–716, May 2012.
[32] S. Essid and C. Févotte, "Smooth nonnegative matrix factorization for unsupervised audiovisual document structuring," *IEEE Trans. Multimedia*, vol. 15, no. 2, pp. 415–425, Feb. 2013.
[33] Z. Yang, Y. Xiang, S. Xie, S. Ding, and Y. Rong, "Nonnegative blind source separation by sparse component analysis based on determinant measure," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 10, pp. 1601–1610, Oct. 2012.
[34] F. Y. Wang, C.-Y. Chi, T.-H. Chan, and Y. Wang, "Nonnegative least-correlated component analysis for separation of dependent sources by volume maximization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 875–888, May 2010.
[35] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York, NY, USA: Cambridge Univ. Press, 1985.
[36] S. Mehrotra, "On the implementation of a primal-dual interior point method," *SIAM J. Optim.*, vol. 2, no. 4, pp. 575–601, 1992.
[37] I. J. Lustig, R. E. Marsten, and D. F. Shanno, "Interior point methods for linear programming: Computational state of the art," *ORSA J. Comput.*, vol. 6, no. 1, pp. 1–14, 1994.
[38] (2005). *MIT Center for Biological and Computation Learning, CBCL Face Database 1*. [Online]. Available: http://www.ai.mit.edu/projects/cbcl
[39] D. Donoho and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts?" in *Proc. Conf. Adv. Neural Inf. Process. Syst. (NIPS)*, 2003, pp. 1–8.
[40] A. J. Bell and T. J. Sejnowski, "The 'independent components' of natural scenes are edge filters," *Vis. Res.*, vol. 37, no. 23, pp. 3327–3338, 1997.
[41] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.
[42] Z. He, A. Cichocki, S. Xie, and K. Choi, "Detecting the number of clusters in n-way probabilistic clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 11, pp. 2006–2021, Nov. 2010.
[43] S. Bengio, F. Pereira, Y. Singer, and D. Strelow, "Group sparse coding," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 22. 2009, pp. 82–89.
[44] F. Wang, N. Lee, J. Sun, J. Hu, and S. Ebadollahi, "Automatic group sparse coding," in *Proc. 25th AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, Aug. 2011, pp. 495–500.
[45] F. Wang, T. Li, and C. Zhang, "Semi-supervised clustering via matrix factorization," in *Proc. Int. Conf. SIA Data Mining (SDM)*, Atlanta, GA, USA, Apr. 2008, pp. 1–12.
[46] F. G. Germain and G. J. Mysore, "Stopping criteria for non-negative matrix factorization based supervised and semi-supervised source separation," *IEEE Signal Process. Lett.*, vol. 21, no. 10, pp. 1284–1288, Oct. 2014.

Zuyuan Yang (M'15) received the B.E. degree from the Hunan University of Science and Technology, Xiangtan, China, in 2003, and the Ph.D. degree from the South China University of Technology, Guangzhou, China, in 2010.

He is currently a Researcher with the Faculty of Automation, Guangdong University of Technology, Guangzhou. His current research interests include blind source separation, nonnegative matrix factorization, and image processing.

Dr. Yang won the Excellent Ph.D. Thesis Award Nomination of China. He joined the National Program for New Century Excellent Talents in University and received the Guangdong Distinguished Young Scholar Award.



Yong Xiang (SM'12) received the Ph.D. degree in electrical and electronic engineering from The University of Melbourne, Melbourne, VIC, Australia.

He is currently an Associate Professor and the Director of Artificial Intelligence and Image Processing Research Cluster with the School of Information Technology, Deakin University, Melbourne. His current research interests include signal and system estimation, information and network security, multimedia (speech/image/video) processing, and wireless sensor networks.

Dr. Xiang has served as the Program Chair, TPC Chair, Symposium Chair, and Session Chair for a number of international conferences. He serves as an Associate Editor of IEEE ACCESS.

Kan Xie was born in Hubei, China. He received the M.S. degree in software engineering from the South China University of Technology, Guangzhou, China, in 2009. He is currently pursuing the Ph.D. degree in intelligent signal and information processing with the Guangdong University of Technology, Guangzhou.

His current research interests include machine learning, nonnegative signal processing, blind signal processing, and biomedical signal processing.

Yue Lai received the B.E. and Ph.D. degrees from the South China University of Technology, Guangzhou, China, in 2005 and 2012, respectively.

He is currently a Researcher with the Faculty of Automation, Guangdong University of Technology, Guangzhou. His current research interests include data mining, privacy preserve, and real-time data analysis.