Smooth nonnegative matrix and tensor factorizations for robust multi-way data analysis

Tatsuya Yokota a,*, Rafal Zdunek b, Andrzej Cichocki a, d, Yukihiko Yamashita c

a RIKEN Brain Science Institute, Japan
b Wroclaw University of Technology, Poland
c Tokyo Institute of Technology, Japan
d Systems Research Institute, Polish Academy of Science, Poland

Abstract

In this paper, we discuss new efficient algorithms for nonnegative matrix factorization (NMF) with smoothness constraints imposed on nonnegative components or factors. Such constraints allow us to alleviate certain ambiguity problems, which facilitates better physical interpretation or meaning. In our approach, various basis functions are exploited to flexibly and efficiently represent the smooth nonnegative components. For noisy input data, the proposed algorithms are robust to the existing smooth and sparse NMF algorithms. Moreover, we extend the proposed approach to the smooth nonnegative Tucker decomposition and smooth nonnegative canonical polyadic decomposition (also called smooth nonnegative tensor factorization). Finally, we conduct extensive experiments on synthetic and real-world multi-way array data to demonstrate the advantages of the proposed algorithms.

1. Introduction

Nonnegative matrix/tensor factorization (NMF/NTF) plays an important role in feature extraction, classification, blind source separation (BSS), denoising, completion of missing values, and clustering of nonnegative signals [4,8,9,11–13,18,22,25,30,37,40].

The standard NMF model is given by

\[ Y \cong AX \in \mathbb{R}^{J \times L}, \tag{1} \]

where \( A = [a_1, a_2, \ldots, a_L] \in \mathbb{R}^{R \times L}, X \in \mathbb{R}^{L \times J}, \) and \( Y = [y_1, y_2, \ldots, y_J] \) is an input matrix consisting of \( J \) observed signals. The goal of NMF is to compute \( A \) and \( X \) from the matrix \( Y \) for a given parameter \( R \). For example, we consider \( R \ll J \ll L \) in BSS problems [5]. In this case, we want to find \( R \) latent source signals from \( J \) mixed observations. NMF gives \( A \) as an estimator of the latent source signals. In the case of extracting parts of facial images [22], \( I \) and \( J \) denote the number of pixels in an image and the number of images, respectively. NMF then represents each facial image as a linear combination of \( R \) nonnegative parts. In the case of clustering tasks [37], \( A \) is the set of cluster centroids, and \( X \) represents the weight parameters of the clusters.

The standard criterion for NMF based on the Euclidean distance is given by

\[ \min \|Y - AX\|_F^2, \quad \text{s.t. } A \geq 0, \ X \geq 0, \tag{2} \]

where \( \| \cdot \|_F \) stands for the Frobenius norm. Obviously, it minimizes the Euclidean distance between the observed signals \( Y \) and the model \( AX \), imposing nonnegativity constraints onto latent source and mixing matrices. When there is no constraint, this decomposition model has an unlimited number of solutions, and it cannot provide any meaningful decomposition. However, nonnegativity constraints narrow down the set of the solutions to these which have some
meaning for the latent components and the mixing systems. Such constraints imposed to both factors $A$ and $X$ can be justified by a non-subtracting mixing system of nonnegative signals, which can reduce redundancy of the solution.

Moreover, non-subtracting mixing systems are used for modeling light, sound, electromagnetic spectra, probability density functions, and image/video brightness. The observed signals, mixing matrices, and latent components are nonnegative, and NMF represents them using the linear superposition system $1$. Hence NMF is a useful tool to analyze such kinds of data. The nonnegativity constraint plays an important role in physical interpretation of the decomposition and extraction of non-redundant signals from physically mixed observations. Basically, luminance signals, spectral signals, textual data, and financial data should be nonnegative, and their latent components are often preferred to be nonnegative for meaningful interpretation of the feature vectors.

In general, NMF/NTF is not unique. Thus, for many types of data, we need to impose some additional constraints to relax the problem of non-uniqueness and obtain physically meaningful components. To date, most researchers have imposed sparsity constraints $14,16,18$. In this paper, we investigate another fundamental constraint: smoothness. The smoothness means that the differences between neighboring values are small in some domain. For example, harmonic signals are smooth in the time domain and sparse in the frequency domain. Considering the nonnegative signals, natural image signals are smooth in the spatial domain, probability density functions may be often smooth in some domain, and spectral intensity of optical waves is smooth in the wavelength domain. The latent components of the above signals may be smooth and nonnegative, and the mixing matrices could be assumed as nonnegative based on non-subtracting physical mixing systems. For this reason, NMF/NTF with smoothness constraints is very useful to analyze such kinds of data. When we assume the non-smooth noise is included in observations, this version of NMF, which is referred to as the smooth NMF, may reduce the effects of the noise on the estimators of nonnegative and smooth latent components. In other words, it should be robust to non-smooth noise. In fact, smooth NMF is useful for analyzing temporally or spatially smooth signals (e.g., natural image data, brain waves, and financial data $4,12,13,38–40$).

Many smooth NMF methods can generally be separated into two groups. In the first one, a smoothness constraint term is added to the NMF criterion. For example, Chen et al. $4$ proposed the addition of a temporal smoothness constraint and a spatial decorrelation constraint into the Frobenius norm, and into the Kullback–Leibler (KL) divergence-based NMF for electroencephalography (EEG) analysis. Zdunek and Cichocki $39,40$ added a Gibbs regularization term for smooth NMF. Drakakis et al. $12$ incorporated a sparseness constraint into the mixing matrix, and a smoothness constraint was added to the feature matrix in the Frobenius norm and KL divergence-based NMF for the analysis of financial data. Essid and Fevotte $13$ applied the KL divergence-based smooth NMF for audiovisual document structuring, and Dong and Li $11$ reported the application of smooth NMF using Laplacian regularization for incomplete matrix factorization.

In the second group, the feature vectors are approximated by a linear combination of several smooth basis vectors. This approach was first proposed by Zdunek $38$, where Gaussian radial basis functions (GRBFs) were used with a single standard deviation parameter. This GRBF-NMF method provides effective performance for robust data analysis with respect to noise. However, the original algorithm was relatively slow, because it employed quadratic programming (QP) optimization and the active-set algorithm. The computational cost of QP optimization increases exponentially for large-scale problems. Thus, the original GRBF-NMF algorithm is not practical for large-scale data.

Another problem is that research into smooth nonnegative ‘tensor’ factorization is not sufficiently well progressed, despite many promising potential applications exist. One reason for this is that most existing algorithms for smooth NMF are quite complex and have a very high computational cost. In this paper, we address the following objectives: to simplify the GRBF-NMF method and develop a new practical algorithm (i.e., reduce the computational cost); to extend the method to the nonnegative Tucker and canonical polyadic (CP) decompositions with additional smoothness constraints. For this purpose, we modify the original problem and propose a new fast algorithm based on the hierarchical alternating least-squares (HALS) method, $6,8$, which is a fast and stable algorithm for general NMF/NTF. Furthermore, we propose two extensions for GRBF-NMF. The first uses more flexible basis functions that consist of Gaussian functions with multiple standard deviation parameters. The second one involves two-dimensional Gaussian functions for processing image data. We call this extension GRBF-NMF-2D basis.

For the second objective, we propose two algorithms for smooth nonnegative Tucker decomposition (NTD) and smooth nonnegative CP decomposition (NCPD). These are extensions of our HALS-based GRBF-NMF algorithm. We call these extensions GRBF-NTD and GRBF-NCPD. Furthermore, the NTF methods are extended to the ‘2D basis’ case. Note that we can select the target modes on which to impose the smoothness constraint. For example, for a 3D tensor with the temporally smooth domain (the first mode), the spatially smooth domain (the second mode), and the trial non-smooth domain (the third mode), the smoothness constraint can be applied to only the first and the second modes.

The remainder of this paper is organized as follows. Section 2 introduces the original GRBF-NMF algorithm for a smooth representation. In Section 3, we propose a novel fast algorithm for GRBF-NMF, and discuss its extensions. Section 4 explores the tensor versions of our approach based on the Tucker and CP models. In Section 5, we investigate the performance and applications of our new HALS-based GRBF-NMF/NTF algorithms, and compare them with some state-of-the-art methods. In Section 6, we discuss several aspects of our work, including potential applications and open problems. Finally, we give our conclusions in Section 7.

2. Smooth nonnegative matrix factorization with function approximation

In this section, we review the basic smooth NMF model using the function approximation proposed by Zdunek $38$. According to this method, a feature vector $a$, is
represented as
\[
\mathbf{a}_r = \sum_{n=1}^{N_r} \phi_n w_{nr} \quad (r = 1, 2, \ldots, R)
\]
where \(\{w_{nr}\}\) are real-valued coefficients, and \(\phi_n\) is a smooth basis function (e.g., Gaussian function). Let \(\Phi = [\phi_1, \ldots, \phi_N] \in \mathbb{R}_+^{1 \times N}\), and \(\mathbf{W} = [w_{nr}] \in \mathbb{R}_+^{N \times R}\). Note that \(\mathbf{W}\) is not restricted to be non-negative. Then, we have the following model for smooth NMF:
\[
\mathbf{Y} \approx \mathbf{W}\Phi, \quad \text{s.t. } \mathbf{W} \succeq 0, \text{ and } \mathbf{X} \succeq 0.
\]
In this model, the feature matrix \(\mathbf{A}\) is approximated by \(\Phi\mathbf{W}\), and the objective is to estimate \(\mathbf{W}\) and \(\mathbf{X}\). When the observed data \(\mathbf{Y}\) includes some noise, this model can reduce its influence via smoothing constraints. For optimization, we estimate the two parameter matrices \(\mathbf{W}\) and \(\mathbf{X}\), since \(\Phi\) is known.

### 2.1. Selection of \(\Phi\)

According to [38], \(\Phi\) using GRBF with a standard deviation \(\sigma\) can be expressed as
\[
\Phi(i, n) = \exp \left( -\frac{(i - n\Delta t)^2}{2\sigma^2} \right),
\]
where \(\Delta t\) is an interval satisfying \(N = [(l - 1) / \Delta t] + 1\) (see Fig. 1). This method is known as GRBF-NMF. When \(\sigma\) is large, the flexibility of this representation decreases, but it is expected that the NMF will then be robust for noisy data. On the other hand, when \(\sigma\) is small, \(\phi_n\) will define orthogonal bases. This increases the flexibility of representation, but weakens the NMF for noisy data. Thus, \(\sigma\) can be regarded as a trade-off parameter. The hyper-parameters \(\Delta t\) and \(\sigma\) can be intuitively interpretable unlike some hyper-parameters of classical penalized methods. For example, \(\Delta t = 1\) and \(\sigma = 1\) can be intuitively justified because they mean clearly the interval and the bandwidth of Gaussian functions. In contrast, a regularization parameter cannot be set so intuitively, and some additional methods for parameter assessing should be used prior to the component estimation.

This approximation approach can be interpreted to narrow down the feasible area of \(\mathbf{A}\) into a subset \(\mathcal{F}\) which consists of only smooth components. It is important to know how to define appropriate \(\mathcal{F}\) which includes various types of smooth components without non-smooth components. In GRBF-NMF, a subset \(\mathcal{F}_{\text{GRBF}} = \{\mathbf{a} \mid \mathbf{a} = \Phi \mathbf{w}, \mathbf{w} \in \mathbb{R}_+^N\}\) is defined, and we find optimal smooth components from the subset \(\mathcal{F}\) for each input data. From this interpretation, smooth components obtained by GRBF-NMF could be robust to non-smooth noise because the feasible solution is limited to only smooth components.

### 2.2. Original GRBF-NMF algorithm

In this section, we introduce the original GRBF-NMF algorithm. To estimate \(\mathbf{W}\) and \(\mathbf{X}\), the most popular criterion is to minimize the Frobenius norm as
\[
\min_{\mathbf{W}, \mathbf{X}} \frac{1}{2} \|\mathbf{Y} - \Phi\mathbf{WX}\|^2_F, \quad \text{s.t. } \Phi\mathbf{W} \succeq 0, \mathbf{X} \succeq 0.
\]
Since this optimization is not convex, we separate this criterion into the following two sub-problems, which are solved alternately and iteratively:
\[
\begin{align*}
&\min_{\mathbf{X}} \frac{1}{2} \|\mathbf{Y} - \Phi\mathbf{WX}\|^2_F, \quad \text{s.t. } \mathbf{X} \succeq 0, \quad (7) \\
&\min_{\mathbf{W}} \frac{1}{2} \|\mathbf{Y} - \Phi\mathbf{WX}\|^2_F, \quad \text{s.t. } \mathbf{W} \succeq 0. \quad (8)
\end{align*}
\]
To solve (7), we can apply the basic alternating least-squares (ALS) approach [9]. The regularized fast combinatorial nonnegative least-squares (FC-NLNS) algorithm [34] can also be used. The FC-NLNS is based on the active-set algorithm [21] that iteratively searches for the set of active variables that satisfy the equality constraints (zero-value). The remaining variables are updated by solving an unconstrained linear least-squares problem. The iterative updates are terminated when the Karush–Kuhn–Tucker (KKT) optimality conditions are met. To solve (8), we transform the objective function to the following vectorized form:
\[
\frac{1}{2} \|\mathbf{Y} - \Phi\mathbf{WX}\|^2_F = \frac{1}{2} \|\mathbf{y} - (\mathbf{X}^T \otimes \Phi) \mathbf{w}\|^2_F
\]
\[
= \frac{1}{2} \mathbf{y}^T \mathbf{y} - \mathbf{y}^T (\mathbf{X}^T \otimes \Phi) \mathbf{w} + \frac{1}{2} \mathbf{w}^T (\mathbf{XX}^T \otimes \Phi^T \Phi) \mathbf{w},
\]
where \(\mathbf{w} = \text{vec}(\mathbf{W}) \in \mathbb{R}^{RN}\) and \(\mathbf{y} = \text{vec}(\mathbf{Y}) \in \mathbb{R}^N\) are vectorized forms of the matrices \(\mathbf{W}\) and \(\mathbf{Y}\), respectively, and \(\otimes\) denotes the Kronecker product. Finally, the problem is transformed to the following QP problem:
\[
\min_{\mathbf{W}} \frac{1}{2} \mathbf{w}^T Q \mathbf{w} + c^T \mathbf{w}, \quad \text{s.t. } (I_R \otimes \Phi) \mathbf{w} \succeq 0, \quad (10)
\]
where \(Q = \mathbf{XX}^T \otimes \Phi^T \Phi \in \mathbb{R}^{RN \times RN}\), \(c = -(\Phi \otimes \Phi^T) \mathbf{y} \in \mathbb{R}^{RN}\), and \(I_R \in \mathbb{R}^{R \times R}\) is an identity matrix.

### 2.3. Characteristics of model

This function approximation model is inspired by the regression methods applied to nonlinear function models. In an earlier study of function approximation by NMF, it was proposed that the feature vectors are fitted by the single Boltzmann distribution function model to factorize the data from fluorescence correlation spectroscopy [36]. However, since the single Boltzmann distribution function model is quite limited, it is difficult to apply it for our objectives. Next, Ding et al. [10] estimated the feature vectors according to \(\mathbf{A} \succeq \mathbf{YW}\), where the dataset \(\mathbf{Y}\) includes positive and negative entries, and \(\mathbf{W} \succeq 0\) is a nonnegative multiplier matrix. They claimed there was a close relation between this model and the k-means clustering. Similarly, we claim that there exists the relation between their model and our model. They employ a weighted
summation as $YW$ for clustering, whereas we employ a linear combination of smooth functions as $FW$ for smooth features. Next, Jiang and Yin [17] proposed to estimate the feature vectors using a wavelet function model for sparse NMF. However, this approach was intended for sparse representations. On the other hand, the GRBF-NMF method can be characterized as a Gaussian mixture model or a kernel regression model [35] to represent smooth and nonnegative feature vectors; this seems to be appropriate for our objectives (i.e., part-based representation and BSS). Thus, the key to this model is the choice of a suitable value for $\sigma$, since the smoothness of results is directly dependent on this parameter. According to the previous study [38], GRBF-NMF gives robust results with respect to noisy data when an appropriate value of $\sigma$ is used.

2.4. Computational issues

The original algorithm employs the active-set algorithm for the matrix $X$ and QP optimization for the matrix $W$. Each algorithm is an excellent optimization method; however, their combined alternate use does not result in an efficient optimization algorithm. This is because the parameter-space dimension for the QP optimization will be large (i.e., $RN$), and the active-set method has to iteratively solve the least-squares problem. Thus, the computational cost becomes very high. The use of strict optimization methods is not indispensable in each step of an iterative algorithm. A low-cost approximation step is often better, even though it does not give an exact solution. Based on this approach, a new fast GRBF-NMF algorithm is proposed in Section 3.

3. Fast algorithm for GRBF-NMF

As stated above, the problem with the original algorithm is its high computational cost of each step because of QP optimization and the active-set algorithm in each iteration (i.e., double loop). In this section, we focus on an algorithmic approach to reduce the computational cost of each step. The total computational efficiency is then evaluated experimentally.

First, we modify the optimization problem in (6) as

$$\min_{W,X} \|Y - FWX\|_F^2;$$

s.t. $W \succeq 0$, $X \succeq 0$, $\|x_r\| = 1$ for $r = 1, \ldots, R,$

(11)

where $x_r$ is the $r$-th row vector of $X = [x_1, \ldots, x_R]^T$. In this problem, the constraint $FW \succeq 0$ is replaced by $W \succeq 0$. Since $\Phi \succeq 0$, $W \succeq 0$ is a sufficient condition for $FW \succeq 0$. This modification provides two benefits for GRBF-NMF: fast algorithm and enhanced smoothness. Because of the smoothness constraint, the complexity of the model decreases slightly; however, the original problem is simplified, the uniqueness of a solution is improved, and a new efficient algorithm can be developed. Moreover, the proposed method with this constraint can find smoother latent components often with meaningful physical representations (see Fig. 5). In many regression models, a highly flexible model may suffer from over-fitting without any regularization. When the data is noisy, such a model often represents not only the main feature, but also noise and outliers. To prevent a such over-fitting, one solution is to use the proposed simpler model, as we do not allow negative values in $W$. Using the Gaussian bases, the convex cone constraint $W \succeq 0$ narrows down the space of feasible solutions more than only the constraint $FW \succeq 0$. In other words, when we assume the smooth and nonnegative latent components, which can be represented under the proposed constraint, and non-smooth noise, the proposed model is more robust for noise and outliers than using the original constraint. The nonnegativity constraint of $W$ in this model is related to the kernel density estimation (KDE) and the Gaussian mixture model of probability density functions. Both methods do not allow negative weights of the Gaussians, and are widely used. The bandwidth of KDE and $\sigma$ of GRBF-NMF are essentially the same because both parameters affect the width of all Gaussian functions. For this reason, some bandwidth selection methods for KDE could be also applicable to GRBF-NMF. Furthermore, some physical signals such as spectra in the Raman spectroscopy, gas chromatography mass spectroscopy, and hyperspectral imaging can be regarded as nonnegative combinations of nonnegative, smooth, and unimodal signals. Our model with the nonnegative $W$ usually quite well reflects the physical constraints, and hence it might be useful to analyze such kinds of data. In addition, we impose the constraint $\|x_r\| = 1$. This does not alter the flexibility, but normalizes each $x_r$.

To obtain a solution to the problem (11), we separate it into sub-problems based on the HALS method [8]. Since the GRBF-NMF model can be decomposed as $FWX = \Phi w_1 x_1^T + \Phi w_2 x_2^T + \cdots + \Phi w_R x_R^T$ (see Fig. 2), we have

$$\min_{w_r} \|Y_r - \Phi w_r x_r^T\|_F^2, \quad \text{s.t. } x_r \geq 0, \quad \|x_r\| = 1.$$  \hspace{1cm} (12)

$$\min_{w_r} \|Y_r - \Phi w_r x_r^T\|_F^2, \quad \text{s.t. } w_r \geq 0.$$  \hspace{1cm} (13)

where $Y_r = Y - \sum_{k \neq r} \Phi w_k x_k^T$.

First, we consider the optimization problem (12). Since $\Phi w_r$ is currently fixed, it is equivalent to a simple least-squares problem with a nonnegativity constraint. This problem has been studied in several papers [6,8,9]. The simplest update rule is given by

$$x_r \leftarrow \left[ Y_r^T \Phi w_r \right]_+,$$

$$\Phi w_r \leftarrow x_r / \|x_r\|,$$

(14)

(15)

where $\left[ x \right]_+ = \max(x, 0)$, and $\varepsilon$ is a very small positive value (typically, $\varepsilon = 10^{-16}$).

Next, we consider the optimization problem (13). The objective function can be transformed to

$$\|Y_r - \Phi w_r x_r^T\|_F^2 = \text{tr}(Y_r^T Y_r) - 2 \text{tr}(Y_r^T \Phi w_r x_r^T) + \text{tr}(w_r^T \Phi^T \Phi w_r),$$

(16)

since $x_r^T x_r = 1$. By computing the gradients of the objective function with respect to $w_r$, the stationary condition for a solution is given by

$$\frac{\partial}{\partial w_r} \|Y_r - \Phi w_r x_r^T\|_F^2 = 2\Phi^T \Phi w_r - 2\Phi^T Y_r x_r = 0.$$  \hspace{1cm} (17)
The closed-form solution leads to update formula $w_r \leftarrow [(\Phi^T \Phi)^{-1} \Phi^T Y x_r^T]_r$; however, this formula is unstable and has a high computational cost. Thus, we do not employ this update rule, but propose a more efficient version. Note that the following problem is equivalent to (13) via the same partial differentiation with respect to $w_r$, since $\|x_r\| = 1$:

$$\min_{w_r} \|Y x_r - \Phi w_r\|^2 \quad \text{s.t.} \quad w_r \geq 0. \quad (18)$$

Moreover, the objective function in (18) can be transformed to

$$\|Y x_r - \Phi w_r\|^2 = \text{tr}(x_r^T Y^T Y x_r) - 2 \text{tr}(Y^T \Phi w_r x_r^T) + \text{tr}(w_r^T \Phi^T \Phi w_r). \quad (19)$$

The only difference with respect to (16) persists in the first term; however, it does not depend on $w_r$. Thus, we can ignore this difference from the viewpoint of optimization with respect to $w_r$. The problem (18) can be solved by some nonnegativity-constrained least-squares (NNLS) algorithms (see [34]); however, such algorithms have a high computational complexity for large-scale problems. To further reduce the computational cost, we propose to use the following multiplicative update rule:

$$w_r \leftarrow w_r \odot [(\Phi^T Y x_r^T) \odot (\Phi^T \Phi w_r)]_r. \quad (20)$$

where $\odot$ and $\odot$ denote element-wise multiplication (Hadamard product) and element-wise division, respectively. The above update rule can be considered as a special case of the multiplicative update rule for the nonnegative quadratic programming proposed by Sha et al. [31], so the monotonic convergence property to the global optimal solution of the problem (18) is guaranteed [31]. The vector $w_r$ is updated by (20), and $x_r$ by (14) and (15) for each iteration step $r$ until convergence. Finally, the proposed algorithm is summarized in Algorithm 1.

**Algorithm 1.** Fast algorithm for GRBF-NMF.

```
1: Input: Y, R, and $\Phi$
2: Initialize: W and X
3: $E = Y - \Phi WX$
4: repeat
5:   for $r = 1, \ldots, R$ do
6:     $Y_r, E + (\Phi w_r)x_r^T$;
7:     $x_r \leftarrow |Y_r^T (\Phi w_r)|_+$;
8:     $x_r \leftarrow x_r / \|x_r\|_2$;
9:     $w_r \leftarrow w_r \odot |(\Phi^T Y x_r^T)|_+ \odot |(\Phi^T \Phi w_r)|_+$;
10:   $E \leftarrow Y_r - (\Phi w_r)x_r^T$;
11:   end for
12: until $\|E\|^2$ converges
13: Output: W and X
```

### 3.1. Computational cost

In this section, we discuss the computational cost of our algorithm. The essence of our contribution is to employ the HALS approach [5,8], and separate the main problem into several sub-problems. Since $W$ is "sandwiched" between $\Phi$ and $X$ in the main problem, its Hessian matrix with respect to $W$ becomes very large, i.e. $RN \times RN$. In contrast, the size of the Hessian in the separated sub-problem is only $N \times N$. The computational cost of one step of the new algorithm is very low, because it consists of only four operations and the thresholding without any complex optimization procedure. If we assume $R \leq J \leq 1 = N$, the maximum computational complexity of the proposed fast algorithm is $O((J^2 R^2)$, assuming that the Cholesky decomposition is used in the QP optimization. Thus, our modification considerably improves the computational efficiency. In Section 5, we confirm experimentally that the proposed method improves the computational efficiency.

### 3.2. Extensive bases for GRBF-NMF

In this section, we discuss the efficient selection of $\Phi$. We can use various bases $\phi$ to make $\Phi$; for example, multiple bases with various $\sigma$ can be mixed. Thus, we propose to construct $\Phi$ as $\Phi \leftarrow \{\Gamma_{\sigma_1}, \Gamma_{\sigma_2}, \ldots, \Gamma_{\sigma_U}\} \in \mathbb{R}^{J \times N}$, where $U$ is the number of $\sigma_u$ and $\Gamma_{\sigma_u}$ denotes the basis matrix with standard deviation $\sigma_u$. For example, if we set $\sigma_1 = \sigma_2 = 2\sigma, \sigma_3 = 3\sigma, \ldots, \sigma_U = 2^{U-1}\sigma$, (21)

$$\Delta t_1 = \delta t, \Delta t_2 = 2\delta t, \Delta t_3 = 4\delta t, \ldots, \Delta t_U = 2^{U-1}\delta t, \quad (22)$$

then $N < 2U / \delta t$ holds for any $U$. Let us set the horizontal size of $\Gamma_{\sigma_u}$ as $N_0$. Then, the horizontal size of $\Phi$ is roughly bounded by

$$N = N_0 + \frac{1}{2} N_0 + \frac{1}{4} N_0 + \cdots + \frac{1}{2^{U-1}} N_0 < 2N_0$$

as $\Delta t = \delta t$.
for any U. Furthermore, we propose to add an additional direct-current (DC) component, such that Φ is given by $\Phi = \{\Gamma_{\sigma_1}, \Gamma_{\sigma_2}, \ldots, \Gamma_{\sigma_n}, 1\}$. Finally, we have $N = \sum_{\sigma} = 1 [(l-1)/2^{n+2} - 1] + 1$.

The use of this extension has several benefits. First, this extension obviously increases flexibility of the model, while keeping the smoothness. Basically, the addition of the Gaussian bases with larger $\sigma$ increases the smoothness. Second, the problem of the parameter selection of $\sigma$ can be reduced by using this extension because the model includes several values of $\sigma$ and redundant bases could be removed automatically in its optimization process. Although this extension increases the computational cost of the algorithm, it is maximally twice higher than for the normal bases. The comparison of the computational time between the normal and extended bases (Fig. 3) shows that the difference is quite small.

### 3.2.1. 2D basis

In this section, we propose another kind of Gaussian basis functions for smooth images. Let the observed signal $y \in \mathbb{R}^l$ be an unfolded vector of an $(l_1 \times l_2)$ matrix $Y \in \mathbb{R}^{l_1 \times l_2}$ representing a two dimensional (2D) signal, where $l = l_1 l_2$. In this case, the use of $\phi_{\sigma_n}$, as defined in (5), affects only the vertical smoothness. However, natural images usually have both vertical and horizontal smoothness. Therefore, we consider the following 2D bases:

$$\Pi_{\sigma_n}^{2D}(i_1, i_2) = \exp \left[ \frac{-1}{2\sigma^2} \left( \frac{i_1}{n_1} \right)^2 \right] \in \mathbb{R}^{l_1 \times l_2},$$

where $n_1 = (n \Delta l - 1) / l_1$ and $n_2 = (n \Delta l - 1) / l_1$ + 1. Thus, we have $n \Delta l = (n_2 - 1)l_1 + 1$, where $1 \leq n_1 \leq l_1$ and $1 \leq n_2 \leq l_2$. In practice, we uniformly fold the matrix $\Pi_{\sigma_n}^{2D}$ into a vector as $\phi_{\sigma_n}^{2D} = \text{vec}(\Pi_{\sigma_n}^{2D}) \in \mathbb{R}^l$, and construct the basis matrix $\Gamma^{2D}$ as $\Phi = \Gamma^{2D} = \{\phi_{\sigma_1}^{2D}, \phi_{\sigma_2}^{2D}, \ldots, \phi_{\sigma_n}^{2D}\} \in \mathbb{R}^{l \times N}$. The multi-$\sigma$ version described in the previous section can be formed in a similar way: $\Phi = \{\Gamma_{\sigma_1}^{2D}, \Gamma_{\sigma_2}^{2D}, \ldots, \Gamma_{\sigma_n}^{2D}, 1\}$.

The 2D Gaussian bases with various $\sigma$ are relative simple to implement, and could be efficient for images which are horizontally and/or vertically smooth. They do not change the computational cost because the size of $\Phi$ is the same.

### 3.3. Dimensionality reduction of the basis matrix

When the size of the basis matrix $\Phi$ is very large, the function approximation can be quite memory-demanding. For example, when we factorize images with the resolution of $256 \times 256$ pixels, each image must be unfolded to a 65,536-dimensional vector. Then, $\Phi$ becomes a $(65,536 \times 65,536)$ matrix, which would occupy about 34.36 GB of memory. This is a critical issue of the GRBF-NMF methods. In this section, we propose a method that drastically reduces the size of the basis matrix $\Phi$. First, the observed signal $y$ is folded into a matrix $Y^{(1)}$. For example, a 1000-dimensional vector is folded into a $(100 \times 10)$ matrix. Next, we replace the observed matrix with a block matrix $Y = [Y^{(1)}, \ldots, Y^{(p)}]$. For instance, if we fold 1000-dimensional vectors into $(100 \times 10)$ block matrices, we can reduce the matrix size of $\Phi$ by almost 99% compared to the conventional method.

For image data, it is effective to split the image into several blocks to be vectorized. We assume that the image $Y_j$ is given by a block matrix

$$Y_j = \begin{pmatrix} D_{11}^{(j)} & \cdots & D_{1q}^{(j)} \\ \vdots & \ddots & \vdots \\ D_{pq}^{(j)} & \cdots & D_{pq}^{(j)} \end{pmatrix} \in \mathbb{R}^{l_1 \times l_2}.$$  

This can be folded as $Y^{(j)} = [\text{vec}(D_{11}^{(j)}), \text{vec}(D_{12}^{(j)}), \ldots, \text{vec}(D_{pq}^{(j)})] \in \mathbb{R}^{l_1 l_2 / (pq) \times pq}$. In this case, the 2D basis is easily applicable. We call this the GRBF-block-NMF method.

Note that this is related to the block transform using the 2D discrete cosine transform (2D-DCT) for image compression, which splits an image into 8 $\times$ 8 blocks and factorizes each block by 64 2D cosine bases. If we apply the 2D cosine bases to $\Phi$, we obtain a similar effect to the block transform. However, our method differs from the block transform, because we do not only obtain coefficients of the bases, but also optimize the bases via a linear model using $\Phi$. Furthermore, the DCT block transform is loss-less, whereas our method is generally lossy.

One problem concerns folding a large-scale vector when its length is a prime number. In this case, there are two approaches: we can either reduce or expand the dimension. Although the reduction approach may lose some information, it entails a lower computational cost. While the expansion approach does not lose any information, it increases the computational cost and the amount of redundant information. We propose to exploit repeating and symmetry techniques for the expansion approach. Let us expand $y \in \mathbb{R}^{N}$ to $z \in \mathbb{R}^{N + M}$. When $y$ has a cyclic feature, it is appropriate to use the repeat-type expansion from $y = [y_1, y_2, \ldots, y_N]^T$ to $z = [y_1, y_2, \ldots, y_N, y_1, y_2, \ldots, y_M]^T$. The symmetry-type expansion takes $y = [y_1, y_2, \ldots, y_N]^T$ and produces $z = [y_1, y_2, \ldots, y_N, y_{N-1}, \ldots, y_{N-M+1}]^T$.
4. Smooth nonnegative tensor factorizations and decompositions

Nonnegative tensor factorization (NTF) has already found numerous applications in positron emission tomography (PET), EEG, spectroscopy, chemometrics, and environmental science [2,9,29]. There are two basic models of tensor factorization/decomposition: Tucker and CP decomposition.

The Tucker model [33] with a nonnegativity constraint for a third-order tensor can be described as

\[
y 
\geq \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} g_{r_1,r_2,r_3} a_{r_1} \odot b_{r_2} \odot c_{r_3},
\]

(25)

where \( Y \in \mathbb{R}^{I \times J \times K} \) is an observed data tensor, \( G \in \mathbb{R}^{R_1 \times R_2 \times R_3} \) is a core tensor, \( A = [a_1, a_2, \ldots, a_{R_1}] \in \mathbb{R}^{I \times R_1}, B = [b_1, b_2, \ldots, b_{R_2}] \in \mathbb{R}^{J \times R_2}, C = [c_1, c_2, \ldots, c_{R_3}] \in \mathbb{R}^{K \times R_3} \) are factor matrices, and \( \odot \) denotes the outer product. The nonnegativity-constrained decomposition is called the nonnegative Tucker decomposition (NTD). The NTD model can be rewritten in matrix and vector forms as \( Y_{(1)} \geq AG_{(1)}(C^T \otimes B^T), Y_{(2)} \geq BG_{(2)}(C^T \otimes A^T), Y_{(3)} \geq CG_{(3)}(B^T \otimes A^T), \) and \( Y = (C \otimes B \otimes A)G \), where \( Y_{(1)} \in \mathbb{R}^{I \times J \times K} \) and \( G_{(1)} \in \mathbb{R}^{R_1 \times R_2 \times R_3} \) are the mode-1 matrix forms of the tensors \( Y \) and \( G \), respectively. The mode-1 matrix form of tensor \( Y \) is defined as \( Y_{(i)}(i, k-1) + j \) for all \( i, j, \) and \( k \). We define the modes 2 and 3 similarly. If we set \( \Xi = G_{(1)}(C^T \otimes B^T) \), the factorization \( Y_{(1)} \geq A \Xi \) can be regarded as NMF. Thus, the nonnegative factor matrix \( A \) can be updated by NMF-based update rules. In a similar way, \( B, C, \) and \( G \) can also be updated by NMF-based update rules (e.g., by applying ALS) [19].

The CP model [3,15] with nonnegativity constraints is a special case of the NTD model with \( R_1 = R_2 = R_3 = R \) and the diagonal tensor \( G = \Lambda = \mathbb{R}^{I \times J \times K} \) with entries \( \lambda_r \) on the main diagonal given by \( Y \geq \sum_{r=1}^{R} \lambda_r a_r \odot b_r \odot c_r \). The nonnegative CP model can be regarded as a straightforward tensor extension of NMF, since \( Y \) can be rewritten using \( Y \geq \sum_{r=1}^{R} \lambda_r a_r \odot b_r \odot c_r \). Thus, the CP model gives an \( R \)-rank approximation of the observed data tensor.

In this section, we discuss the extension of GRBF-NMF to tensor decompositions. We often assume that vectors \( a_r, b_r, \) and/or \( c_r \) are smooth, and can be approximated as \( \Phi_r a_r, \Phi_r b_r, \) and/or \( \Phi_r c_r \), respectively. The update rule for \( w_r \) can be applied to \( v_r, b_r, \) and hence, we only provide the update rule for \( w_r \) in Sections 4.1 and 4.2.

4.1. Tucker model

In this section, we extend GRBF-NMF to the smooth NTD to obtain \( A \) by the function approximation \( \Phi W \). For this purpose, we consider the following optimization criterion:

\[
\min_{G, W, B, C} \left\| Y - G \times_1 \Phi W \times_2 B \times_3 C \right\|_F^2,
\]

\[\text{s.t. } G \geq 0, \quad W \geq 0, \quad B \geq 0, \quad C \geq 0.\]

(26)

We call this model as GRBF-NTD. The key to this problem is the update of \( W \). This is because the other factors \( (B, C, G) \) can be updated by the existing ALS-based algorithm developed for the standard NTD [19,27,28,30]. Focusing on \( W \) and using a matrix form, the problem can be rewritten as

\[
\min_{\mathbf{W}} \left\| Y - \Phi W G_{(1)}(C^T \otimes B^T) \right\|_F^2, \quad \text{s.t. } \mathbf{W} \geq 0.
\]

(27)

Setting \( Z_r = Y_{(1)} - \sum_{k \neq r} \Phi_r B_r C_r^T \) and \( \Xi = [\xi_1, \ldots, \xi_R]^T \) for \( \mathbf{w}_r \) as

\[
\min_{\mathbf{w}_r} \left\| Z_r - \Phi_r B_r C_r^T \right\|_F^2, \quad \text{s.t. } \mathbf{w}_r \geq 0.
\]

(28)

The problem (28) is essentially equivalent to (13). Since the condition \( \left\| \xi_r \right\|_2 = 1 \) must be satisfied, the update rule for \( \mathbf{w}_r \) is given by

\[
\xi_r \leftarrow \left[ Z_r \Phi_r B_r C_r^T \right]_{+}.
\]

(29)

\[
\xi_r \leftarrow \left\| \xi_r \right\|_2.
\]

(30)

\[
\mathbf{w}_r \leftarrow \left[ \mathbf{w}_r \odot \Phi_r Z_r \xi_r \right] \odot \left( \Phi_r^T \Phi_r \mathbf{w}_r \right)_{+}.
\]

(31)

Note that update rules (29) and (30) are important steps for the accurate update of \( \mathbf{w}_r \). Algorithm 2 summarizes the GRBF-NTD. The lines 5–13 give the update procedure for \( W \). If we apply the function approximation to factor matrices \( B, C, \) and \( G \), the 14th and/or 15th line must be modified in a similar way to \( W \).


1: \textbf{Input: } \( Y, R_1, R_2, R_3, \) and \( \Phi \)
2: \textbf{Initialize: } \( W \) randomly, and \( B, C, G \) by some initialization method for NTD
3: \( E \leftarrow Y - G \times_1 \Phi W \times_2 B \times_3 C; \)
4: \textbf{repeat}
5: \( Z_r \leftarrow E_{(1)} + \left( \Phi_r B_r C_r^T \right); \)
6: \textbf{for } \( r = 1, \ldots, R_1 \textbf{do} \)
7: \( \xi_r \leftarrow \left[ Z_r \Phi_r B_r C_r^T \right]_{+}; \)
8: \( \xi_r \leftarrow \left\| \xi_r \right\|_2; \)
9: \( \mathbf{w}_r \leftarrow \left[ \mathbf{w}_r \odot \Phi_r Z_r \xi_r \right] \odot \left( \Phi_r^T \Phi_r \mathbf{w}_r \right)_{+}; \)
10: \( E_{(1)} \leftarrow Z_r - \left( \Phi_r B_r C_r^T \right); \)
11: \textbf{end for} \)
12: \( A \leftarrow \Phi W; \)
13: \( B \leftarrow B \odot Y_{(2)}(A \odot C) G_{(2)}^T \odot \left( B G_{(2)}(A^T A \odot C^T C) G_{(2)}^T \right); \)
14: \( C \leftarrow C \odot Y_{(3)}(A \odot B G_{(2)}(A^T A \odot C^T C) G_{(2)}^T \right); \)
15: \( G \leftarrow G \odot Y (X^2 A^T \times B^T \times C^T \otimes (G \times A^T A \otimes B B^T \times C C^T); \)
16: \( E \leftarrow Y - G \times_1 \Phi W \times_2 B \times_3 C; \)
18: \textbf{until } \|E\|_F \text{ converges}
19: \textbf{Output: } \( W, B, C, \) and \( G \)

4.2. CP model

In this section, we extend GRBF-NMF to nonnegative CP decomposition (NCPD) to obtain \( A \) by the function
approximation \( \Phi W \). The criterion is given by

\[
\min_{\lambda, w, b, c} \left\| Y - \sum_{r=1}^{R} \lambda_r (\Phi w_r) \otimes b_r \circ c_r \right\|_F^2,
\]

s.t. \( \lambda_r \geq 0, w_r \geq 0, b_r \geq 0, c_r \geq 0 \),
\( \| \Phi w_r \|_F = \| b_r \|_2 = \| c_r \|_2 = 1 \) \hspace{1cm} (32)

for \( r = 1, 2, \ldots, R \). We refer to this as the GRBF-NCPD method. The problem (32) can also be solved with the HALS algorithm. Setting \( Y := Y - \sum_{r=1}^{R} \lambda_r (\Phi w_r) \otimes b_r \circ c_r \) and using the first-way matrix form, the sub-problem for \( w_r \) can be rewritten as

\[
\min_{w_r} \left\| Y_{(n)} - \lambda_r (\Phi w_r) (c_r \otimes b_r) \right\|_F^2, \quad \text{s.t.} \quad \| \Phi w_r \|_F = 1, w_r \geq 0.
\] \hspace{1cm} (33)

Similarly, as regards the relation between (13) and (18), the problem (33) can be transformed to

\[
\min_{w_r} \left\| Y_{(n)} (c_r \otimes b_r) - \lambda_r (\Phi w_r) \right\|_F^2, \quad \text{s.t.} \quad \| \Phi w_r \|_F = 1, w_r \geq 0.
\] \hspace{1cm} (34)

where we have \( (c_r \otimes b_r) (c_r \otimes b_r) = (c_r \otimes c_r) \otimes (b_r \otimes b_r) = 1 \). Since there is a constraint \( \| \Phi w_r \|_F = 1 \) in (34), its solution is not dependent on \( \lambda_r \). Thus, the update rule for \( w_r \) is finally given by

\[
w_r \rightarrow \left[ w_r \cap (\Phi^T Y_{(n)} (c_r \otimes b_r)) \cap (\Phi^T \Phi w_r) \right]_+,
\]

\[\text{Algorithm 3. Fast GRBF-NCPD algorithm.}\]

1: \textbf{Input:} \( Y, R, \) and \( \Phi \)
2: \textbf{Initialize:} \( W \) randomly, and \( B, C, G \) by some initialization method for nonnegative CP decomposition
3: \( E := Y - C \times T \Phi W \times B \times C \)
4: \textbf{repeat}
5: \textbf{for} \( r = 1, \ldots, R \) \textbf{do}
6: \( Y_r \rightarrow E + \lambda_r (\Phi w_r) \otimes b_r \circ c_r \)
7: \( w_r \rightarrow [w_r \cap (\Phi^T Y \times T b_r \circ c_r)] \cap (\Phi^T \Phi w_r) \)
8: \( w_r \rightarrow w_r \cap (\Phi^T \Phi w_r) \)
9: \( b_r \rightarrow [Y \times T \Phi w_r] \cap (b_r \cap c_r) \)
10: \( b_r \rightarrow b_r \cap (b_r \cap c_r) \)
11: \( c_r \rightarrow [Y \times T \Phi w_r] \cap (b_r \cap c_r) \)
12: \( c_r \rightarrow c_r \cap (b_r \cap c_r) \)
13: \( \lambda_r \rightarrow [Y \times T \Phi w_r] \cap (b_r \cap c_r) \)
14: \( E \rightarrow Y - \lambda_r (\Phi w_r) \otimes b_r \circ c_r \)
15: \textbf{end for}
16: \textbf{until} \( \| E \|_F^2 \) converge
17: \textbf{Output:} \( W, B, C, \) and \( G \)

5. Experiments

5.1. Parameter setting of GRBF-NMF

In general, the parameters \( \Delta t \) or \( \delta t \) and \( U \) affect the representational ability (flexibility) of the model. When we do not have any prior information about the data, it is better to set \( \Delta t = \delta t = 1 \) and \( U \geq 1 \). We set \( U = 4 \) as default. Smaller \( \Delta t \) or \( \delta t \) would increase a number of parameters, leading to a decrease in the computational efficiency. Thus, \( \Delta t \) or \( \delta t \) can be considered as a trade-off parameter with respect to the computational cost and flexibility. The parameter \( \sigma \) has the highest impact on the smoothness. Its increase amplifies the smoothness as well as diminishes the flexibility of the model. We have found an optimal value of \( \sigma \) experimentally.

In the experiments, we compared several sparse and smooth NMF methods which have one or two hyper-parameters to control sparseness, smoothness, or a level of regularization. Kim and Park’s developed in [18] sparse NMF has two hyper-parameters for sparseness and regularization, non-smooth NMF [26] has one hyper-parameter for non-smoothness, Gibbs regularized NMF [39,40], Chen’s smooth NMF [4], the original GRBF-NMF, and the proposed GRBF-NMF with parameters: \( \Delta t = \delta t = 1, \sigma = 1.0 \), and \( U = 1,4 \). We employed the same strategy of stopping criterion for all the methods. When the difference in the values of the objective function between before and after updates is smaller than \( 10^{-4} \), the algorithm was stopped. Fig. 3 shows the log 10-scale computation time averaged over 10 runs for each rank decomposition. The computation time of the original algorithm increased exponentially, and for \( R \geq 6 \) failed on 32-bit hardware because of high memory requirement. In contrast, the proposed algorithm required much less time to obtain a solution, and its computation time was relatively independent of \( U \). It is around 10–100 times faster than the original algorithm for \( R = 3,4,5 \). Although it may require many more iterations than the original one and is still slower than the existing sparse or smooth NMF methods, except for the Kim&Park’s SNMF, the overall computation time is significantly improved, leading to real-time computations.

The convergence of the proposed algorithm is also examined. We used the same data and set \( R = 4, \delta t = 1, \sigma = 1.0 \), and \( U = 4 \). The objective function value was recorded for each iteration over 1000 simulations. In each simulation, the initial values of \( W \) and \( X \) were set randomly. Fig. 4 shows the functional boxplots [32] of the objective function values for all the simulations. The runs with the values greater than 1.5 times the range of the central region were regarded as outliers. When the difference between successive steps is
smaller than $\epsilon = 10^{-3}$, the algorithm was assumed to have converged. From this criterion, we can see that all the runs converged to local optima. The average value of the objective function is 895.3 ± 3.0, and that for the non-outlying runs is 895.3 ± 0.2. The variation in the latter becomes very small. Note that the objective function’s value is not high because each element in the data matrix lies in the range [0, 255] and the size of the data is $99 \times 10$. For all the runs and iterations, no increase in the objective function was observed, and the rate of outlying runs (i.e., getting stuck in local minima) was 5.9%.

Next, we compared the estimates of $W$ and $A$ obtained by the proposed and original algorithms. We set $U=1$, $\Delta t=1$, $\sigma=0.8$, and $R=2$ for the proposed algorithm, and $\Delta t=1$, $\sigma=0.8$, and $R=2$ for the original one (i.e., the same conditions). Fig. 5 shows the estimates of $W$ and $A$ obtained by both algorithms. Despite the difference between the constraints $W \succeq 0$ and $\Phi W \succeq 0$, the results for $A$ are highly correlated. Furthermore, the proposed algorithm provides smoother components than the original one. Many spikes in the estimates, which are not fully removed with the original algorithm, are smoothed using the proposed one.

5.3. Blind source separation

In this experiment, we applied GRBF-NMF to the BSS problem for both synthetic and real-world datasets. The generative model is given by $Y = [SX_0 + E_0]_+^R$, where $S \in \mathbb{R}^{1 \times R}$ is an original source signal matrix, $X_0 \in \mathbb{R}^{R \times J}$ is a mixing matrix, $E_0 \in \mathbb{R}^{R \times J}$ is a Gaussian noise matrix, and $Y \in \mathbb{R}^{1 \times J}$ is an observed signal matrix. The signal-to-noise ratio (SNR) is defined as $\text{SNR} = 10 \log_{10} \left( \frac{\|SX_0\|_F^2}{\|SX_0 - Y\|_F^2} \right)$. Furthermore, we evaluated the estimated source $A = \Phi W$ using the mean signal-to-distortion ratio (SDR) measure. First, the

---

**Fig. 4.** Functional boxplots of the objective function for 1000 random runs: the black curve shows the median result, the blue denotes the 50% central region, the outer blue lines are maximum and minimum values for all non-outlying simulations, and the red dashed lines are outlying simulations. The rate of outlying runs was 5.9%. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

**Fig. 5.** Estimates of $W$ and $A$ obtained by the proposed and original smooth NMF algorithms.
SDR is defined as \( SDR (s, a) = 10 \log_{10} \frac{\|s\|_2^2}{\|s - a\|_2^2} \), where \( s \) is the original source and \( a \) is the estimated source. The calculation of the mean SDR consists of several steps. First, each signal is normalized, because the NMF problem may not have a unique solution. Next, the SDR combination matrix is calculated as \( M(r_1, r_2) = SDR(s_{r_1}, a_{r_2}) \) for \( r_1, r_2 = 1, 2, \ldots, R \), and we obtain the optimal one-to-one correspondence of \( r_1 \) and \( r_2 \) to maximize the sum of SDR. To find the optimal correspondence, we employed the Hungarian method [20]. Finally, the mean SDR can be calculated by averaging the corresponding SDRs.

In the first BSS experiment, we used synthetic sparse and smooth nonnegative signals given by nonnegative sine curves and soft thresholding. Fig. 6b (left) shows the original sources \( S = \{s_1, \ldots, s_R\} \in \mathbb{R}_+^{R \times 1} \) with the parameters: \( R = 4 \) and \( I = 1024 \). The number of observed mixed signals is equal to \( J = 20 \). Each element of the mixing matrix \( X_0 \) is the absolute value of a sample generated from the normal distribution \( \mathcal{N}(0, 1) \). We applied the following baseline and state-of-the-art NMF methods: multiplicative NMF [22,23], Kim and Park's sparse NMF [18], non-smooth NMF [26], Gibbs regularized NMF [39,40], Chen's smooth NMF [4], and GRBF-NMF. The tests are carried out for various noise levels. The parameters in GRBF-NMF were set to \( U = 4, \delta t = 1, \) and \( \sigma = 1.0 \). Fig. 6b illustrates the moving-average filtered signals that are separated with the Gibbs regularized NMF, Kim & Park's SNMF, and GRBF-NMF, respectively. The mean-SDR values of the estimated signals are displayed in the titles of the corresponding figures. Fig. 6a shows the mean SDRs for all the simulations and noise levels. The individual lines in Fig. 6a are the linear least-squares regression (LSR) results of the individual methods. From this figure, we can confirm the robustness of the GRBF-NMF method against noise.

Next, we used the synthetic sparse and smooth nonnegative 2D signals illustrated in Fig. 7b (left). In this experiment, each signal was a \( 64 \times 64 = 4096 \)-dimensional vector. Fig. 7b illustrates the separated signals given by the smooth NMF, Kim & Park's SNMF, and GRBF-NMF-2D basis, respectively. Fig. 7a plots the mean SDRs and the linear LSR results for various noise levels. Kim & Park's sparse NMF gives the best performance with low-noise data, whereas GRBF-NMF outperformed the other methods for high-noise data.

Finally, we applied GRBF-NMF to the blind hyperspectral unmixing problem using four spectral signatures selected at random from the US Geological Survey (USGS) database. The angle between any pair of vectors \( \{a_i, a_j\} \) is...
greater than \( 10^\circ \), and the reflectance values of the endmembers (i.e., source signals \( a_r \)) are measured in 224 spectral bands, distributed in the interval \( 0.4 \)–\( 2.5 \) \( \mu m \). Note that all these spectral signals are strictly positive (no zero elements). Since the endmembers can be assumed to be smooth, we applied smooth and classical NMF methods \([4,7,22,23,39,40]\) without sparseness constraint. We generated abundance maps synthetically (i.e., weight parameters \( x_r \)) using a low-pass filtering strategy \([25]\); the resolution of each abundance map is \( 64 \times 64 \) pixels. The mixtures are corrupted with an i.i.d zero-mean Gaussian noise with \( \text{SNR} = 30 \) dB. The original endmembers and abundance maps are shown in Fig. 8, and those estimated using the multiplicative NMF \([22,23]\), Gibbs regularized NMF \([39,40]\), and GRBF-NMF are presented in Fig. 8. The mean SDRs of these results were also evaluated. Fig. 9 illustrates the mean SDR statistics obtained for estimating the matrices \( A \) (left) and \( X \) (right) with the smooth and classical NMF algorithms. The mean SDR samples were obtained for 20 uniformly distributed random initializations for the factors \( A^{(0)}, W^{(0)} \), and \( X^{(0)} \) for each algorithm. From the results, we can see that GRBF-NMF outperforms the other NMF methods in terms of the estimation for \( A \) and \( X \). Since GRBF-NMF only imposes a nonnegativity constraint on \( X \), this implies that there is a possibility of further improvement by imposing some additional sparsity or norm-based regularization constraints.

5.4. Local parts analysis

GRBF-block-NMF was used for parts-based feature extraction. The \( 3456 \times 4608 \) image, shown in Fig. 10b, was corrupted with the noise level of 10 dB. The noisy image was then transformed to a \( (1024 \times 15552) \) nonnegative matrix by unfolding the individual \( 32 \times 32 \) blocks. Setting \( R = 20 \), GRBF-block-NMF extracts 20 local parts-based feature images shown in Fig. 10c–h (negative image). For comparison, we applied the standard multiplicative NMF \([22,23]\) non-smooth NMF (nNMF) \([26]\), Chen’s smooth NMF \([4]\), and Gibbs regularized smooth NMF \([39,40]\). In this experiment, we did not test Kim and Park’s sparse NMF because of its high computational time. Only the parts shown in Fig. 10c were
learned from the original noise-free image (a). This is used for the reference to evaluate the performance of the methods for the parts-based analysis in presence of noise. The parts in Fig. 10d–h were learned from the noisy image (b), and the mean peak-SDRs between the reference and the estimated parts were calculated. We can see that almost all the features are corrupted with strong noise, except for those extracted with nsNMF and GRBF-NMF, which are fairly clear. Comparing the estimates obtained with multiplicative NMF from the noise-free image with those given by the other methods, we obtain the following SDR values: (d) 16.5 dB, (e) 18.95 dB, (f) 17.12 dB, (g) 16.8 dB, and (h) 20.13 dB. Thus, GRBF-NMF produced the best result. In fact, some noise remains in the features obtained with nsNMF. To further improve GRBF-NMF, other constraints can be considered. It is well known that sparseness works efficiently for extraction of parts-based features, but GRBF-NMF does not include any sparseness constraints. Thus, the proposed algorithms can be extended by combining smoothness with sparseness constraints for feature extraction problems.

Next, we applied the nonnegative matrix and tensor factorization techniques to the analysis of a color image. We used a (2048 × 2048 × 3) color image (Fig. 11a) perturbed with the noise of 10 dB (Fig. 11b). The data was separated into (64 × 64 = 4096) blocks, from which 20 local parts were extracted. Each part was a (32 × 32 × 3) color image, and individual blocks in the image could be represented by a linear combination of 20 parts. We apply the matrix, CP, and Tucker factorization models. In the CP and Tucker models,
we reshaped the image to a \((32 \times 32 \times 3 \times 4092)\) tensor. In the matrix model, the image was reshaped to a 
\((3072 \times 4092)\) matrix, and each column was regarded as the vectorized form of a 
\((32 \times 32 \times 3)\) block. The multiplicative NMF, NCPD, NTD, GRBF-NMF-2D basis, 
GRBF-NCPD-2way, and GRBF-NTD-2way were applied to the individual data. We set 
\(R = 20\) for the NMF and CP models, and \(R_1 = R_2 = 8, R_3 = 3, R_4 = 20\) for the Tucker model. In the 
CP and Tucker models, the parts are given by the 1st–3rd 
factor matrices \((A, B, C)\) and the core tensor \((G)\). The \(k\)th part is 
given by \(\lambda_k a_k \odot b_k \odot c_k\) in the CP decomposition, and 
\(\sum_{r_1, r_2, k} b_{r_1, r_2} a_{r_1} \odot b_k \odot c_{r_2}\) in the Tucker decomposition. 

Fig. 11c–k shows the extracted local parts for each individual algorithm. Fig. 11c, f, and i was extracted from the 
original color image, the others were extracted from the 
noisy image. The GRBF-based methods and NCPD extracted 
smooth and sparse local parts from the noisy image, and 
gave similar results to those extracted by the standard 
algorithms from the original image. From the mean SDR 
results, we can confirm that the individual GRBF-based 
methods outperformed the other methods for each model.

6. Discussion

6.1. Nonnegative and smooth multi-way analysis

The smooth NMF methods have been studied exten-
sively. Smoothness and nonnegativity are physically mean-
ingful in many applications. Many of these approaches are 
based on additional penalty terms, and are only applicable
to matrix factorization. The contributions of this research are twofold: first, we have employed and improved a new function approximation approach for smooth NMF; second, we extended this approach to multi-way nonnegative and smooth component feature analysis. The proposed smooth NMF/NTF outperformed the other NMF/NTF methods in the BSS and parts extraction experiments with noisy data. The nonnegative and smooth multi-way analysis is a somewhat novel technique. It has appeared to have promising applications in various areas of multi-way real-world data analysis, including brain, audio, and visual signal processing. A proper combination with decorrelation, statistical independence, or other meaningful constraints should find some attractive applications.

Moreover, the estimation of missing values in an incomplete matrix/tensor (image/video) [11, 24] can be considered as an interesting and potential application of GRBF-NMF/NTF because image/video signals are generally smooth and...
missing elements can be distributed randomly. However, our algorithm cannot be directly applied to this problem since some special techniques, which the standard NMF framework does not include, are required to treat incomplete data. Hence, the matrix/tensor completion using GRBF-NMF/NTF could be included in a future work.

6.2. Scalability problem

In Section 3.3, we mentioned the scalability problem of $\Phi$, and proposed a technique for reducing the matrix size. This was applied to the local parts-based analysis of a large image. However, it still cannot be used for low-rank approximation and BSS problems when the number of observations is very large. To address such problems, we may need to consider a new scheme, or some preprocessing step to reduce the dimensionality. This scalability challenge is an open problem.

6.3. Another option for updating

In this paper, we focused on reducing the computational cost of updating $X$ and $W$. Thus, the proposed update rules (14) and (20) are very simple and have a low computational complexity. In the cases where the data size is not large (e.g., $I,J \leq 100$), some well-studied nonnegativity-constrained least-squares algorithms based on the active-set or interior point methods could be employed to update $X$ and $W$, giving better convergence to exact solutions with moderate computational costs. It will be worthwhile to use different options depending on the scale of the problem.

7. Conclusions

In this paper, we proposed the family of new fast GRBF-NMF algorithms, multiple-$\sigma$ extension for various degrees of data resolution, 2D basis extension for image processing, and block-wise extension to reduce the size of the basis matrix. These techniques were extended to nonnegative tensor decompositions with the Tucker and CP models. The BSS and ‘parts-based’ feature extraction experiments were carried out to compare the proposed methods with state-of-the-art NMF methods. The proposed algorithms do not require matrix inversion or mathematical programming techniques, such as $\ell_1$-norm optimization. Thus, they are much faster than the original algorithm. Moreover, we proposed the use of the 2D basis for unfolded 2D array data (i.e., vector signals), and demonstrated that the GRBF-NMF-2D basis algorithm gives improved results. Furthermore, the GRBF-based methods work well for the nonnegative Tucker and CP models with both single and multi-way smooth representations. In the BSS experiments, the GRBF-NMF method provided significantly better results than those obtained with the conventional NMF methods, including state-of-the-art methods for various noise levels. Finally, GRBF-block-NMF/NCMP/NTF also produced good results in local ‘parts-based’ feature extraction.

Appendix A. Supplementary data

Supplementary data associated with this paper can be found in the online version at: http://dx.doi.org/10.1016/j.sigpro.2015.02.003.

References


