Smooth Nonnegative Matrix and Tensor Factorizations for Robust Multi-way Data Analysis

Tatsuya Yokota[a], Rafal Zdunek[b], Andrzej Cichocki[a], Yukihiko Yamashita[c]

[a] *RIKEN Brain Science Institute, Japan*
[b] *Wroclaw University of Technology, Poland*
[c] *Tokyo Institute of Technology, Japan*

Abstract

In this paper, we discuss a new efficient algorithm for nonnegative matrix factorization (NMF) with smooth constraints imposed on nonnegative components or factors. Such constraints allow us to extend the applicability of NMF techniques, and to extract unique components with some 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 algorithm is more robust than the existing smooth and sparse NMF algorithms. Moreover, we extend the proposed approach to 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.

Keywords: Nonnegative Matrix Factorization (NMF), Nonnegative CP decomposition (NCPD), Nonnegative Tucker decomposition (NTD), smooth component analysis, blind source separation (BSS), multi-way data analysis, Gaussian radial basis function

1. Introduction

Nonnegative matrix/tensor factorization (NMF/NTF) plays an important role in feature extraction, classification, blind source separation (BSS), denoising, the completion of missing values, and clustering for nonnegative signals.

The standard NMF model is given by

$$\boldsymbol{Y} \simeq \boldsymbol{A}\boldsymbol{X} \in \mathbb{R}_{+}^{I \times J}, \tag{1}$$

where $\boldsymbol{A} = [\boldsymbol{a}_1,\ \boldsymbol{a}_2,\ \ldots,\ \boldsymbol{a}_R] \in \mathbb{R}_{+}^{I \times R}$, $\boldsymbol{X} \in \mathbb{R}_{+}^{R \times J}$, and $\boldsymbol{Y} = [\boldsymbol{y}_1,\ \boldsymbol{y}_2,\ \ldots,\ \boldsymbol{y}_J]$ is an observation matrix consisting of J observations. The goal of NMF is to compute $\boldsymbol{A}$ and $\boldsymbol{X}$ from the observation matrix $\boldsymbol{Y}$ for a given parameter R. For example, we consider $R \leq J \ll I$ in BSS problems [5]. In this case, we want to find R latent source signals from J mixed observations. NMF gives $\boldsymbol{A}$ as an estimator of the latent source signals. In the case of extracting parts of facial images [20], 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 [33], $\boldsymbol{A}$ is the set of cluster centroids, and $\boldsymbol{X}$ represents the weight parameters of the clusters. The nonnegativity constraint plays an important role in the physical interpretation of decomposition and the extraction of independent signals from physically mixed observations. Basically, luminance signals, spectral signals, text data, and financial data should be nonnegative, and their latent components are often preferred to be nonnegative for the interpretation of feature vectors. For the BSS problem in particular, the latent signals should be nonnegative. Furthermore, the physically linear mixing system is often given by the nonnegative mixing matrix, such as for audio signals, luminance signals, optical waves, and wavelength spectra. The nonnegative constraint on the feature vector and mixing system plays an important role in separating independent features. In fact, NMF can be applied to a wide range of real-world data analyses, such as document clustering, blind image separation, and image/video denoising [33, 18, 36, 11].

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 [16, 18, 14]. In this paper, we investigate another fundamental constraint: smoothness. Obviously, a signal may be smooth and

sparse in separate domains. For example, harmonic signals are smooth in the time domain and sparse in the frequency domain. Moreover, many physical latent variables (e.g., event related potential) are often relatively smooth. In fact, the smooth NMF is useful for analyzing temporally or spatially smooth signals (e.g., natural image data, brain waves, and financial data) [4, 35, 36, 12, 13, 34].

The many smooth NMF methods can generally be separated into two approaches. The first adds some smooth constraint term into 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 the Kullback–Leibler (KL) divergence-based NMF for electroencephalography (EEG) analysis, and Zdunek and Cichocki [35, 36] 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.

The second approach approximates the feature vectors by a linear combination of several smooth basis vectors. This approach was first proposed by Zdunek [34], 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 the many promising potential applications. 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:

- Simplify the GRBF-NMF method and develop a new, practical algorithm (i.e., reduce the computational cost).
- Extend the method to the nonnegative Tucker and canonical polyadic (CP) decompositions.

For this purpose, we modify the original problem and propose a new, fast algorithm based on the hierarchical alternating least-squares (HALS) method [8, 6], 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 involves two-dimensional Gaussian functions for processing image data. We call this extension GRBF-NMF-2Dbasis.

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 '2Dbasis' case. Note that we can select the target modes on which to impose the smooth constraint. For example, for a 3D tensor with a temporally smooth domain (the first mode), spatially smooth domain (the second mode), and trial domain (the third mode), the smooth 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 state-of-the-art methods. In Section 6, we discuss several aspects of our work,

including its application to 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 [34]. According to this method, a feature vector $\boldsymbol{a}_r$ is represented as

$$\boldsymbol{a}_r = \sum_{n=1}^{N} \boldsymbol{\phi}_n w_{nr}, \quad (r = 1, 2, ..., R) \tag{2}$$

where $\{w_{nr}\}$ are real-valued coefficients, and $\boldsymbol{\phi}_n$ is a smooth basis function (e.g., Gaussian function). Let $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_N] \in \mathbb{R}_+^{I \times N}$, and $\boldsymbol{W} = [w_{nr}] \in \mathbb{R}^{N \times R}$. Note that $\boldsymbol{W}$ is not resticted to be non-negative. Then, we have the following model for smooth NMF:

$$\boldsymbol{Y} \simeq \boldsymbol{\Phi}\boldsymbol{W}\boldsymbol{X}, \text{ s.t. } \boldsymbol{\Phi}\boldsymbol{W} \geq 0, \text{ and } \boldsymbol{X} \geq 0. \tag{3}$$

In this model, the feature matrix $\boldsymbol{A}$ is approximated by $\boldsymbol{\Phi}\boldsymbol{W}$, and the objective is to estimate $\boldsymbol{W}$ and $\boldsymbol{X}$. When the observed data $\boldsymbol{Y}$ includes some noise, this model can reduce its influence via smoothing constraints. For optimization, we estimate the two parameter matrices $\boldsymbol{W}$ and $\boldsymbol{X}$, since $\boldsymbol{\Phi}$ is known.

2.1. Selection of $\boldsymbol{\Phi}$

Zdunek [34] expressed $\boldsymbol{\Phi}$ using GRBF with a standard deviation σ as

$$\boldsymbol{\Phi}(i, n) = \exp\left[-\frac{(i - n\Delta t)^2}{2\sigma^2}\right], \tag{4}$$

where Δt is an interval satisfying $N = \lfloor (I-1)/\Delta t \rfloor + 1$ (see Fig. 1). When σ 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 σ is small, $\boldsymbol{\phi}_n$ will define orthogonal bases. This increases the flexibility of representation, but weakens the NMF for noisy data. Thus, σ can be regarded as a trade-off parameter. This method is known as GRBF-NMF.

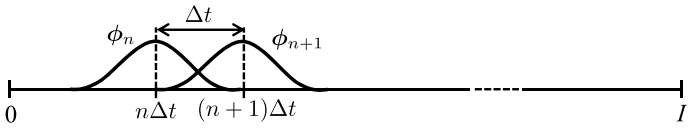


Figure 1: Basis function $\boldsymbol{\phi}_n$

2.2. Original GRBF-NMF Algorithm

In this section, we introduce the original GRBF-NMF algorithm. To estimate $\boldsymbol{W}$ and $\boldsymbol{X}$, the most popular criterion is to minimize the Frobenius norm as

$$\underset{\boldsymbol{W},\ \boldsymbol{X}}{\text{minimize}}\ \frac{1}{2}||\boldsymbol{Y}-\boldsymbol{\Phi}\boldsymbol{W}\boldsymbol{X}||_F^2,\ \text{s.t.}\ \boldsymbol{\Phi}\boldsymbol{W}\geq 0,\ \boldsymbol{X}\geq 0. \tag{5}$$

Since this optimization is not convex, we separate this criterion into the following two sub-problems, which are solved alternately and iteratively:

$$\underset{\boldsymbol{X}}{\text{minimize}}\ \frac{1}{2}||\boldsymbol{Y}-\boldsymbol{\Phi}\boldsymbol{W}\boldsymbol{X}||_F^2,\ \text{s.t.}\ \boldsymbol{X}\geq 0, \tag{6}$$

$$\underset{\boldsymbol{W}}{\text{minimize}}\ \frac{1}{2}||\boldsymbol{Y}-\boldsymbol{\Phi}\boldsymbol{W}\boldsymbol{X}||_F^2,\ \text{s.t.}\ \boldsymbol{\Phi}\boldsymbol{W}\geq 0. \tag{7}$$

To solve (6), we can apply the basic alternating least-squares (ALS) approach [9]. The regularized fast combinatorial nonnegative least-squares (FC-NNLS) algorithm [30] can also be used. This approach is based on the active-set algorithm. To solve (7), we transform the objective function to the following vectorized form:

$$\begin{aligned}\frac{1}{2}||\boldsymbol{Y}-\boldsymbol{\Phi}\boldsymbol{W}\boldsymbol{X}||_F^2 &= \frac{1}{2}||\bar{\boldsymbol{y}}-(\boldsymbol{X}^T\otimes\boldsymbol{\Phi})\bar{\boldsymbol{w}}||^2\\ &= \frac{1}{2}\bar{\boldsymbol{y}}^T\bar{\boldsymbol{y}}-\bar{\boldsymbol{y}}^T(\boldsymbol{X}^T\otimes\boldsymbol{\Phi})\bar{\boldsymbol{w}}+\frac{1}{2}\bar{\boldsymbol{w}}^T(\boldsymbol{X}\boldsymbol{X}^T\otimes\boldsymbol{\Phi}^T\boldsymbol{\Phi})\bar{\boldsymbol{w}},\end{aligned} \tag{8}$$

where $\bar{\boldsymbol{w}}=\text{vec}(\boldsymbol{W})\in\mathbb{R}^{RN}$ and $\bar{\boldsymbol{y}}=\text{vec}(\boldsymbol{Y})\in\mathbb{R}^{IJ}$ are vectorized forms of the matrices $\boldsymbol{W}$ and $\boldsymbol{Y}$, respectively, and $\otimes$ denotes the Kronecker product. Finally, the problem is transformed to the following QP problem:

$$\underset{\bar{\boldsymbol{w}}}{\text{minimize}}\ \frac{1}{2}\bar{\boldsymbol{w}}^T\boldsymbol{Q}\bar{\boldsymbol{w}}+\boldsymbol{c}^T\bar{\boldsymbol{w}},\ \text{s.t.}\ (\boldsymbol{I}_R\otimes\boldsymbol{\Phi})\bar{\boldsymbol{w}}\geq 0, \tag{9}$$

where $\boldsymbol{Q}=\boldsymbol{X}\boldsymbol{X}^T\otimes\boldsymbol{\Phi}^T\boldsymbol{\Phi}\in\mathbb{R}^{RN\times RN}$, $\boldsymbol{c}=-(\boldsymbol{X}\otimes\boldsymbol{\Phi}^T)\bar{\boldsymbol{y}}\in\mathbb{R}^{RN}$, and $\boldsymbol{I}_R\in\mathbb{R}^{R\times R}$ is an identity matrix.

2.3. Characteristics of the Model

This function approximation model is inspired by the regression methods employed for 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 [32]. However, since the single Boltzmann distribution function model is quite limited, it is difficult to apply this for our objectives. Next, Ding et al. [10] estimated the feature vectors according to $\boldsymbol{A} \simeq \boldsymbol{YW}$, where the dataset $\boldsymbol{Y}$ includes positive and negative entries, and $\boldsymbol{W} \geq 0$ is a nonnegative multiplier matrix. They claimed there was a close relation between this model and k-means clustering. Similarly, we claim that there exists a relation between their model and our model. They employ a weighted summation as $\boldsymbol{YW}$ for clustering, whereas we employ a linear combination of smooth functions as $\boldsymbol{\Phi W}$ for smooth features. Next, Jiang and Yin [17] proposed to estimate the feature vectors using a wavelet function model for sparse NMF. However, this was intended for sparse representations. On the other hand, the GRBF-NMF method can be characterized as a Gaussian mixture model or some kernel regression model [31] 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 σ, since the smoothness of results is directly dependent on this parameter. According to a previous study [34], GRBF-NMF gives robust results with respect to noisy data when an appropriate value of σ is used.

2.4. Computational Issues

The original algorithm employs the active-set algorithm for the matrix $\boldsymbol{X}$ and QP optimization for the matrix $\boldsymbol{W}$. 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 to evaluate the optimality of the

current active set. 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 the 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 computational efficiency to reduce the computational cost of each step. The total computational efficiency is then evaluated experimentally.

First, we modify the GRBF-NMF optimization problem (5) as

$$\begin{aligned} &\underset{\boldsymbol{W},\boldsymbol{X}}{\text{minimize}} \ ||\boldsymbol{Y}-\boldsymbol{\Phi}\boldsymbol{W}\boldsymbol{X}||_F^2, \\ &\text{s.t. } \boldsymbol{W}\geq 0,\ \boldsymbol{X}\geq 0,\ ||\boldsymbol{x}_r||^2=1 \text{ for } r=1,...,R, \end{aligned} \tag{10}$$

where $\boldsymbol{x}_r^T$ is the r-th row vector of $\boldsymbol{X}=[\boldsymbol{x}_1,...,\boldsymbol{x}_R]^T$. In this problem, the constraint $\boldsymbol{\Phi}\boldsymbol{W}\geq 0$ is replaced by $\boldsymbol{W}\geq 0$. Since $\boldsymbol{\Phi}\geq 0$, $\boldsymbol{W}\geq 0$ is a sufficient condition for $\boldsymbol{\Phi}\boldsymbol{W}\geq 0$. Under this constraint, the flexibility of the model decreases slightly; however, it becomes more robust to noise. 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 this over-fitting, one solution is to use the proposed simpler model, as we do not allow negative values in $\boldsymbol{W}$. Thus, the proposed model can be considered to be more robust for noise and outliers than the original model. In addition, we impose the constraint $||\boldsymbol{x}_r||^2=1$. This does not alter the flexibility, but normalizes each $\boldsymbol{x}_r$.

To obtain a solution to (10), we separate (10) into sub-problems based on the HALS method [8]. Since the GRBF-NMF model can be decomposed to $\boldsymbol{\Phi}\boldsymbol{W}\boldsymbol{X}=\boldsymbol{\Phi}\boldsymbol{w}_1\boldsymbol{x}_1^T+\boldsymbol{\Phi}\boldsymbol{w}_2\boldsymbol{x}_2^T+\cdots+\boldsymbol{\Phi}\boldsymbol{w}_R\boldsymbol{x}_R^T$ (see Fig. 2), problem (10) can be

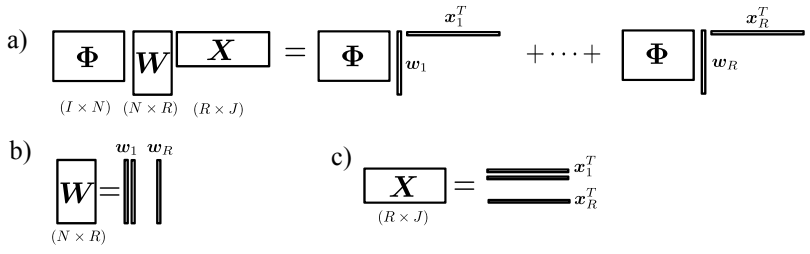


Figure 2: Decomposition of the GRBF-NMF model

separated into the following sub-problems:

$$\underset{\boldsymbol{x}_r}{\text{minimize}} \ ||\boldsymbol{Y}_r - \boldsymbol{\Phi}\boldsymbol{w}_r\boldsymbol{x}_r^T||_F^2, \ \text{s.t.} \ \boldsymbol{x}_r \geq 0, \ ||\boldsymbol{x}_r||^2 = 1. \tag{11}$$

$$\underset{\boldsymbol{w}_r}{\text{minimize}} \ ||\boldsymbol{Y}_r - \boldsymbol{\Phi}\boldsymbol{w}_r\boldsymbol{x}_r^T||_F^2, \ \text{s.t.} \ \boldsymbol{w}_r \geq 0, \tag{12}$$

where $\boldsymbol{Y}_r := \boldsymbol{Y} - \sum_{k \neq r} \boldsymbol{\Phi}\boldsymbol{w}_k\boldsymbol{x}_k^T$.

First, we consider problem (11). Since $\boldsymbol{\Phi}\boldsymbol{w}_r$ is currently fixed, it is equivalent to a simple least-squares problem with a nonnegativity constraint. This problem has been studied in many papers [6, 8, 9]. The simplest update rule is given by

$$\boldsymbol{x}_r \leftarrow [\boldsymbol{Y}_r^T\boldsymbol{\Phi}\boldsymbol{w}_r]_+, \tag{13}$$

$$\boldsymbol{x}_r \leftarrow \boldsymbol{x}_r / ||\boldsymbol{x}_r||, \tag{14}$$

where $[x]_+ := \max(x, \varepsilon)$, and ε is a very small positive value (e.g., $\varepsilon = 10^{-16}$).

Next, we consider problem (12). The objective function can be transformed to

$$\begin{aligned} &||\boldsymbol{Y}_r - \boldsymbol{\Phi}\boldsymbol{w}_r\boldsymbol{x}_r^T||_F^2 \\ &= \mathrm{tr}(\boldsymbol{Y}_r^T\boldsymbol{Y}_r) - 2\mathrm{tr}(\boldsymbol{Y}_r^T\boldsymbol{\Phi}\boldsymbol{w}_r\boldsymbol{x}_r^T) + \mathrm{tr}(\boldsymbol{w}_r^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{w}_r), \end{aligned} \tag{15}$$

since $\boldsymbol{x}_r^T\boldsymbol{x}_r = 1$. By computing the partial derivative of the objective function with respect to $\boldsymbol{w}_r$, the stationary condition for a solution is given by

$$\frac{\partial}{\partial \boldsymbol{w}_r}||\boldsymbol{Y}_r - \boldsymbol{\Phi}\boldsymbol{w}_r\boldsymbol{x}_r^T||_F^2 = 2\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{w}_r - 2\boldsymbol{\Phi}^T\boldsymbol{Y}_r\boldsymbol{x}_r = 0. \tag{16}$$

This direct solution will be $\boldsymbol{w}_r = [(\boldsymbol{\Phi}^T\boldsymbol{\Phi})^{-1}\boldsymbol{\Phi}^T\boldsymbol{Y}_r\boldsymbol{x}_r]_+$; however, this formula is unstable and has a high computational cost. Thus, we do not employ this update

rule, but propose a more effective version. Note that the following problem is equivalent to (12) via the same partial differentiation with respect to $\boldsymbol{w}_r$, since $||\boldsymbol{x}_r|| = 1$:

$$\underset{\boldsymbol{w}_r}{\text{minimize}} \ ||\boldsymbol{Y}_r\boldsymbol{x}_r - \boldsymbol{\Phi}\boldsymbol{w}_r||^2, \ \text{s.t.} \ \boldsymbol{w}_r \geq 0. \tag{17}$$

Moreover, the objective function in (17) can be transformed to

$$||\boldsymbol{Y}_r\boldsymbol{x}_r - \boldsymbol{\Phi}\boldsymbol{w}_r||^2 = \text{tr}(\boldsymbol{x}_r\boldsymbol{x}_r^T\boldsymbol{Y}_r^T\boldsymbol{Y}_r) - 2\text{tr}(\boldsymbol{Y}_r^T\boldsymbol{\Phi}\boldsymbol{w}_r\boldsymbol{x}_r^T) + \text{tr}(\boldsymbol{w}_r^T\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{w}_r). \tag{18}$$

The only difference from (15) is in the first term; however, neither first term depends on $\boldsymbol{w}_r$. Thus, we can ignore this difference from the viewpoint of optimization with respect to $\boldsymbol{w}_r$. Problem (17) can be solved by some nonnegativity-constrained least-squares (NNLS) algorithms (see [30]); however, such algorithms have a high computational complexity for large-scale problems. To further reduce the computational cost, we propose the following multiplicative update rule:

$$\boldsymbol{w}_r \leftarrow \boldsymbol{w}_r \circledast [\boldsymbol{\Phi}^T\boldsymbol{Y}_r\boldsymbol{x}_r]_+ \oslash (\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{w}_r), \tag{19}$$

where $\circledast$ and $\oslash$ denote element-wise multiplication and element-wise division, respectively.

Finally, the proposed algorithm is summarized in Algorithm 1.

3.1. Computational Cost

In this section, we discuss the computational cost of our algorithm. 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 $R \leq J \leq I = N$, the maximum computational complexity of the proposed fast algorithm is $\mathcal{O}(I^2R)$ of arithmetic operations in each iteration. For comparison, the maximum computational complexity of the original algorithm is $\mathcal{O}(I^3R^3)$, assuming that Cholesky decomposition is used in the QP optimization. Thus, our modification dramatically improves the computational efficiency. In Section 5, we not only confirm that the proposed method

Algorithm 1 Fast algorithm for GRBF-NMF

Input: $\boldsymbol{Y}$, R, and $\boldsymbol{\Phi}$
Initialize: $\boldsymbol{W}$ and $\boldsymbol{X}$
$\boldsymbol{E} = \boldsymbol{Y} - \boldsymbol{\Phi}\boldsymbol{W}\boldsymbol{X}$;
repeat
  for $r = 1, \ldots, R$ **do**
    $\boldsymbol{Y}_r \leftarrow \boldsymbol{E} + (\boldsymbol{\Phi}\boldsymbol{w}_r)\boldsymbol{x}_r^T$;
    $\boldsymbol{x}_r \leftarrow [\boldsymbol{Y}_r^T(\boldsymbol{\Phi}\boldsymbol{w}_r)]_+$;
    $\boldsymbol{x}_r \leftarrow \boldsymbol{x}_r / ||\boldsymbol{x}_r||$;
    $\boldsymbol{w}_r \leftarrow \boldsymbol{w}_r \circledast [\boldsymbol{\Phi}^T(\boldsymbol{Y}_r\boldsymbol{x}_r)]_+ \oslash \{\boldsymbol{\Phi}^T(\boldsymbol{\Phi}\boldsymbol{w}_r)\}$;
    $\boldsymbol{E} \leftarrow \boldsymbol{Y}_r - (\boldsymbol{\Phi}\boldsymbol{w}_r)\boldsymbol{x}_r^T$;
  end for
until $||\boldsymbol{E}||_F^2$ converges
Output: $\boldsymbol{W}$ and $\boldsymbol{X}$

provides an improvement in computational efficiency, but also demonstrate its robustness to noisy data.

3.2. Theoretical Guarantee of Monotonic Non-increasing Property

The proposed update rule (19) has the important property that the objective function of (17) is monotonically non-increasing. In this section, we propose and prove a theorem.

Theorem 3.1. *Let* $\boldsymbol{v} \in \mathbb{R}^I$, $\boldsymbol{\Phi} \in \mathbb{R}_+^{I \times N}$, $\boldsymbol{w} \in \mathbb{R}_+^N$, *and* $F(\boldsymbol{w}) := \frac{1}{2}||\boldsymbol{v} - \boldsymbol{\Phi}\boldsymbol{w}||^2$ *be an objective function. Then, the update rule* $\boldsymbol{w}^{t+1} = \boldsymbol{w}^t \circledast [\boldsymbol{\Phi}^T\boldsymbol{v}]_+ \oslash (\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{w}^t)$ *does not increase the objective function. Thus,* $F(\boldsymbol{w}^{t+1}) \leq F(\boldsymbol{w}^t)$.

Note that the theorem is generally true, and the update rule can be applied to other problems. Substituting $\boldsymbol{Y}_r\boldsymbol{x}_r$ into $\boldsymbol{v}$, the theorem can be applied to our problem. Theorem 3.1 can be proved by the following definition and lemmas:

Definition 3.1. $G(\boldsymbol{w}, \boldsymbol{w}')$ *is an auxiliary function for* $F(\boldsymbol{w})$ *if the following*

conditions hold:

$$G(\boldsymbol{w}, \boldsymbol{w}') \geq F(\boldsymbol{w}), \quad G(\boldsymbol{w}, \boldsymbol{w}) = F(\boldsymbol{w}). \tag{20}$$

Lemma 3.1. *We define the diagonal matrix* $\boldsymbol{K}(\boldsymbol{w}')$ *as*

$$[\boldsymbol{K}(\boldsymbol{w}')]_{ij} = \delta_{ij}(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{w}')_i/w'_i. \tag{21}$$

Then,

$$G(\boldsymbol{w}, \boldsymbol{w}') = F(\boldsymbol{w}') + (\boldsymbol{w} - \boldsymbol{w}')^T\nabla F(\boldsymbol{w}') + \frac{1}{2}(\boldsymbol{w} - \boldsymbol{w}')^T\boldsymbol{K}(\boldsymbol{w}')(\boldsymbol{w} - \boldsymbol{w}') \tag{22}$$

is an auxiliary function for $F(\boldsymbol{w})$.

Lemma 3.1 has already been proved [21]. This auxiliary function is quadratic with respect to $\boldsymbol{w}$, and the minimization of $G(\boldsymbol{w}, \boldsymbol{w}')$ subject to nonnegative $\boldsymbol{w}$ produces an update rule for the minimization of $F(\boldsymbol{w})$. Next we introduce t to indicate the number of updates of $\boldsymbol{w}$, then, $\boldsymbol{w}^t$ is after t-times updates of $\boldsymbol{w}^0$. The following lemmas express an appropriate update rule to obtain $\boldsymbol{w}^{t+1}$ from $\boldsymbol{w}^t$.

Lemma 3.2. *The expression*

$$\begin{aligned}\boldsymbol{w}^{t+1} &= [\boldsymbol{w}^t - \boldsymbol{K}(\boldsymbol{w}^t)^{-1}\nabla F(\boldsymbol{w}^t)]_+ \\ &= \boldsymbol{w}^t \circledast [\boldsymbol{\Phi}^T\boldsymbol{v}]_+ \oslash (\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{w}^t)\end{aligned}$$

is a solution of

$$\underset{\boldsymbol{w}}{\text{minimize}}\ G(\boldsymbol{w}, \boldsymbol{w}^t), \text{ subject to } \quad \boldsymbol{w} \geq 0. \tag{23}$$

Proof. The Lagrange function of problem (23) is given by

$$L(\boldsymbol{w}) = G(\boldsymbol{w}, \boldsymbol{w}^t) - \boldsymbol{w}^T\boldsymbol{\gamma}, \tag{24}$$

where $\boldsymbol{\gamma} \geq 0$ is a vector of Lagrangian multipliers. Its Karush-Kuhn-Tucker (KKT) conditions are given by

$$\frac{\partial L}{\partial \boldsymbol{w}} = \nabla F(\boldsymbol{w}^t) + \boldsymbol{K}(\boldsymbol{w}^t)(\boldsymbol{w} - \boldsymbol{w}^t) - \boldsymbol{\gamma} = 0, \tag{25}$$

$$w_i \geq 0,\ \gamma_i \geq 0,\ w_i\gamma_i = 0 \text{ for } i = 1, 2, ..., N. \tag{26}$$

We can rewrite (25) as

$$\gamma_i = [\nabla F(\boldsymbol{w}^t)]_i + [\boldsymbol{K}(\boldsymbol{w}^t)(\boldsymbol{w} - \boldsymbol{w}^t)]_i = w_i(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{w}^t)_i/w_i^t - (\boldsymbol{\Phi}^T\boldsymbol{v})_i.$$

If $(\boldsymbol{\Phi}^T\boldsymbol{v})_i \leq 0$, then $w_i^{t+1} = w_i^t \cdot 0 \cdot (\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{w}^t)_i = 0$ and $\gamma_i = 0 \cdot (\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{w}^t)_i/w_i^t + |(\boldsymbol{\Phi}^T\boldsymbol{v})_i| \geq 0$. If $(\boldsymbol{\Phi}^T\boldsymbol{v})_i > 0$, then $w_i^{t+1} \geq 0$ and $\gamma_i = w_i^{t+1}(\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{w}^t)_i/w_i^t - (\boldsymbol{\Phi}^T\boldsymbol{v})_i = (\boldsymbol{\Phi}^T\boldsymbol{v})_i - (\boldsymbol{\Phi}^T\boldsymbol{v})_i = 0$. Therefore, w_i^{t+1} satisfies the KKT conditions for all i, and is a global optimal solution to problem (23). □

Lemma 3.3. *If $G(\boldsymbol{w}^{t+1}, \boldsymbol{w}^t) \leq G(\boldsymbol{w}^t, \boldsymbol{w}^t)$, then $F(\boldsymbol{w}^{t+1}) \leq F(\boldsymbol{w}^t)$.*

Proof. $F(\boldsymbol{w}^{t+1}) \leq G(\boldsymbol{w}^{t+1}, \boldsymbol{w}^t) \leq G(\boldsymbol{w}^t, \boldsymbol{w}^t) = F(\boldsymbol{w}^t)$. □

3.3. Extensive Bases for GRBF-NMF

In this section, we discuss the efficient selection of $\boldsymbol{\Phi}$. We can use various bases $\boldsymbol{\phi}$ to make $\boldsymbol{\Phi}$; for example, multiple bases with various σ can be mixed. Thus, we propose to construct $\boldsymbol{\Phi}$ as

$$\boldsymbol{\Phi} = [\boldsymbol{\Phi}_{\sigma_1}, \boldsymbol{\Phi}_{\sigma_2}, ..., \boldsymbol{\Phi}_{\sigma_U}] \in \mathbb{R}^{I \times N}, \tag{27}$$

where U is the number of σ_u and $\boldsymbol{\Phi}_{\sigma_u}$ denotes the basis matrix with standard deviation σ_u. For example, if we set

$$\sigma_1 = \sigma,\ \sigma_2 = 2\sigma,\ \sigma_3 = 4\sigma,\ ...,\ \sigma_U = 2^{U-1}\sigma, \tag{28}$$

$$\Delta t_1 = \delta t,\ \Delta t_2 = 2\delta t,\ \Delta t_3 = 4\delta t,\ ...,\ \Delta t_U = 2^{U-1}\delta t, \tag{29}$$

then $N < 2I/\delta t$ holds for any U. Let us set the horizontal size of $\boldsymbol{\Phi}_{\sigma_1}$ as N_0. Then, the horizontal size of $\boldsymbol{\Phi}$ is roughly bounded by $N = N_0 + \frac{1}{2}N_0 + \frac{1}{4}N_0 + \cdots + \frac{1}{2^{U-1}N_0} < 2N_0$ for any U. Furthermore, we propose to add an additional direct-current (DC) component, such that $\boldsymbol{\Phi}$ is given by

$$\boldsymbol{\Phi} = [\boldsymbol{\Phi}_{\sigma_1}, \boldsymbol{\Phi}_{\sigma_2}, ..., \boldsymbol{\Phi}_{\sigma_U}, \boldsymbol{1}]. \tag{30}$$

Finally, we have $N = \sum_{u=1}^{U}[\lfloor (I-1)/(2^{u-1}\delta t) \rfloor + 1] + 1$. This extension can be applied to various resolutions of data.

3.3.1. 2D basis

We propose another kind of Gaussian basis function for smooth image signals. Let the observed signal $\boldsymbol{y}_j \in \mathbb{R}^I$ be an unfolded vector of an $(I_1 \times I_2)$ matrix $\boldsymbol{Y}_j \in \mathbb{R}^{I_1 \times I_2}$ representing the image, where $I = I_1 I_2$. In this case, using $\boldsymbol{\phi}_n$ as defined in (4) represents only the vertical smoothness. However, natural images usually have both vertical and horizontal smoothness. Therefore, we consider the following 2D basis:

$$\boldsymbol{\Phi}_n^{(2)}(i_1, i_2) := \exp\left[-\frac{1}{2\sigma^2}\left\|\begin{pmatrix} i_1 \\ i_2 \end{pmatrix} - \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}\right\|^2\right] \in \mathbb{R}^{I_1 \times I_2}, \tag{31}$$

where

$$n_1 = \{(n\Delta t - 1) \bmod I_1\} + 1, \tag{32}$$

$$n_2 = \lfloor (n\Delta t - 1)/I_1 \rfloor + 1. \tag{33}$$

Thus, we have $n\Delta t = (n_2 - 1)I_1 + n_1$, where $1 \leq n_1 \leq I_1$ and $1 \leq n_2 \leq I_2$. In practice, we unfold the matrix $\boldsymbol{\Phi}_n^{(2)}$ into a vector as $\boldsymbol{\phi}_n^{(2)} = \mathrm{vec}(\boldsymbol{\Phi}_n^{(2)}) \in \mathbb{R}^I$, and construct the basis matrix $\boldsymbol{\Phi}^{(2)}$ as

$$\boldsymbol{\Phi}^{(2)} = [\boldsymbol{\phi}_1^{(2)}, \boldsymbol{\phi}_2^{(2)}, ..., \boldsymbol{\phi}_N^{(2)}] \in \mathbb{R}^{I \times N}. \tag{34}$$

The multi-σ version described in the previous section can be formed in a similar way:

$$\boldsymbol{\Phi} = [\boldsymbol{\Phi}_{\sigma_1}^{(2)}, \boldsymbol{\Phi}_{\sigma_2}^{(2)}, ..., \boldsymbol{\Phi}_{\sigma_U}^{(2)}, \mathbf{1}]. \tag{35}$$

3.4. Dimensionality Reduction of the Basis Matrix

When the size of the basis matrix $\boldsymbol{\Phi}$ is very large, function approximation can occupy a lot of memory. For example, when we factorize images with a resolution of 256×256 pixels, each image must be unfolded to a 65536-dimensional vector. Then, $\boldsymbol{\Phi}$ becomes a (65536×65536) 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 $\boldsymbol{\Phi}$.

First, the observed signal $\boldsymbol{y}_j$ is folded into a matrix $\boldsymbol{Y}^{(j)}$. For example, a 1000-dimensional vector is folded into a (100×10) matrix. Next, we replace the observed matrix with a block matrix $\widetilde{\boldsymbol{Y}} = [\boldsymbol{Y}^{(1)}, ..., \boldsymbol{Y}^{(J)}]$. For instance, if we fold 1000-dimensional vectors into (100×10) block matrices, we can reduce the matrix size of $\boldsymbol{\Phi}$ 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 $\boldsymbol{Y}_j$ is given by a block matrix

$$\boldsymbol{Y}_j = \begin{pmatrix} \boldsymbol{D}_{11}^{(j)} & \cdots & \boldsymbol{D}_{1q}^{(j)} \\ \vdots & \ddots & \vdots \\ \boldsymbol{D}_{p1}^{(j)} & \cdots & \boldsymbol{D}_{pq}^{(j)} \end{pmatrix} \in \mathbb{R}^{I_1 \times I_2}. \tag{36}$$

This can be folded as $\boldsymbol{Y}^{(j)} = [\mathrm{vec}(\boldsymbol{D}_{11}^{(j)}),\ \mathrm{vec}(\boldsymbol{D}_{12}^{(j)}),\ ...,\ \mathrm{vec}(\boldsymbol{D}_{pq}^{(j)})] \in \mathbb{R}^{I_1 I_2/(pq) \times pq}$. 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 a 2D discrete cosine transform (2D-DCT) for image compression, which splits an image into 8×8 blocks and factorizes each block by 64 2D cosine bases. If we apply the 2D cosine bases to $\boldsymbol{\Phi}$, we obtain a similar effect to the block transform. However, our method differs from the block transform, because we not only obtain coefficients of the bases, but also optimize the bases via linear model using $\boldsymbol{\Phi}$. Furthermore, the DCT block transform is lossless, whereas our method is generally lossy.

One problem concerns folding the large-scale vector when the dimension 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 repeat and symmetry methods for the expansion approach. Let us consider expanding $\boldsymbol{y} \in \mathbb{R}^N$ to $\boldsymbol{z} \in \mathbb{R}^{N+M}$. When $\boldsymbol{y}$ has a cyclic feature, it is appropriate to use the repeat-type expansion from $\boldsymbol{y} = [y_1, y_2, ..., y_N]^T$ to $\boldsymbol{z} = [y_1, y_2, ..., y_N, y_1, y_2, ..., y_M]^T$. The symmetry-type expansion takes $\boldsymbol{y} = [y_1, y_2, ..., y_N]^T$ and produces $\boldsymbol{z} =$

Algorithm 2 Fast GRBF-NTD algorithm

Input: $\underline{\boldsymbol{Y}}$, R_1, R_2, R_3, and $\boldsymbol{\Phi}$

Initialize: $\boldsymbol{W}$ randomly, and $\boldsymbol{B}$, $\boldsymbol{C}$, $\underline{\boldsymbol{G}}$ by some initialization method for NTD

$\underline{\boldsymbol{E}} \leftarrow \underline{\boldsymbol{Y}} - \underline{\boldsymbol{G}} \times_1 \boldsymbol{\Phi}\boldsymbol{W} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C}$;

repeat

  $\boldsymbol{\Xi} \leftarrow \boldsymbol{G}_{(1)}(\boldsymbol{C}^T \otimes \boldsymbol{B}^T)$;

  for $r = 1, \ldots, R_1$ **do**

    $\boldsymbol{Z}_r \leftarrow \boldsymbol{E}_{(1)} + (\boldsymbol{\Phi}\boldsymbol{w}_r)\boldsymbol{\xi}_r^T$;

    $\boldsymbol{\xi}_r \leftarrow [\boldsymbol{Z}_r^T(\boldsymbol{\Phi}\boldsymbol{w}_r)]_+$;

    $\boldsymbol{\xi}_r \leftarrow \boldsymbol{\xi}_r / ||\boldsymbol{\xi}_r||$;

    $\boldsymbol{w}_r \leftarrow [\boldsymbol{w}_r \circledast \{\boldsymbol{\Phi}^T(\boldsymbol{Z}_r\boldsymbol{\xi}_r)\} \oslash \{\boldsymbol{\Phi}^T(\boldsymbol{\Phi}\boldsymbol{w}_r)\}]_+$;

    $\boldsymbol{E}_{(1)} \leftarrow \boldsymbol{Z}_r - (\boldsymbol{\Phi}\boldsymbol{w}_r)\boldsymbol{\xi}_r^T$;

  end for

  $\boldsymbol{A} \leftarrow \boldsymbol{\Phi}\boldsymbol{W}$;

  $\boldsymbol{B} \leftarrow \boldsymbol{B} \circledast (\boldsymbol{Y}_{(2)}(\boldsymbol{A} \otimes \boldsymbol{C})\boldsymbol{G}_{(2)}^T) \oslash (\boldsymbol{B}\boldsymbol{G}_{(2)}(\boldsymbol{A}^T\boldsymbol{A} \otimes \boldsymbol{C}^T\boldsymbol{C})\boldsymbol{G}_{(2)}^T)$;

  $\boldsymbol{C} \leftarrow \boldsymbol{C} \circledast (\boldsymbol{Y}_{(3)}(\boldsymbol{A} \otimes \boldsymbol{B})\boldsymbol{G}_{(3)}^T) \oslash (\boldsymbol{C}\boldsymbol{G}_{(3)}(\boldsymbol{A}^T\boldsymbol{A} \otimes \boldsymbol{B}^T\boldsymbol{B})\boldsymbol{G}_{(3)}^T)$;

  $\underline{\boldsymbol{G}} \leftarrow \underline{\boldsymbol{G}} \circledast (\underline{\boldsymbol{Y}} \times_1 \boldsymbol{A}^T \times_2 \boldsymbol{B}^T \times_3 \boldsymbol{C}^T) \oslash (\underline{\boldsymbol{G}} \times_1 \boldsymbol{A}\boldsymbol{A}^T \times_2 \boldsymbol{B}\boldsymbol{B}^T \times_3 \boldsymbol{C}\boldsymbol{C}^T)$;

  $\underline{\boldsymbol{E}} \leftarrow \underline{\boldsymbol{Y}} - \underline{\boldsymbol{G}} \times_1 \boldsymbol{\Phi}\boldsymbol{W} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C}$;

until $||\boldsymbol{E}||_F^2$ converges

Output: $\boldsymbol{W}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\underline{\boldsymbol{G}}$

$[y_1, y_2, ..., y_N, y_N, y_{N-1}, ..., y_{N-M+1}]^T$.

4. Smooth Nonnegative Tensor Factorizations and Decompositions

Nonnegative tensor decompositions have already found numerous applications in positron emission tomography (PET), EEG, spectroscopy, chemometrics, and environmental science [9, 2, 26]. There are two basic models of tensor factorization/decomposition: Tucker and CP decomposition (CPD).

The Tucker model [29] for third-order tensors can be described as

$$\begin{aligned}\underline{\boldsymbol{Y}} &\simeq \underline{\boldsymbol{G}} \times_1 \boldsymbol{A} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C} \\ &= \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} g_{r_1,r_2,r_3} \boldsymbol{a}_{r_1} \circ \boldsymbol{b}_{r_2} \circ \boldsymbol{c}_{r_3}, \end{aligned} \tag{37}$$

where $\underline{\boldsymbol{Y}} \in \mathbb{R}_+^{I \times J \times K}$ is an observed data tensor, $\underline{\boldsymbol{G}} \in \mathbb{R}_+^{R_1 \times R_2 \times R_3}$ is a core tensor, $\boldsymbol{A} \in \mathbb{R}_+^{I \times R_1}$, $\boldsymbol{B} \in \mathbb{R}_+^{J \times R_2}$, $\boldsymbol{C} \in \mathbb{R}_+^{K \times R_3}$ are factor matrices, $\times_k$ is the k-th way tensor-matrix product, and $\circ$ denotes the outer product. The nonnegativity-constrained decomposition is called the nonnegative Tucker decomposition (NTD). The Tucker model can be rewritten in matrix and vector forms as

$$\boldsymbol{Y}_{(1)} \simeq \boldsymbol{A}\boldsymbol{G}_{(1)}(\boldsymbol{C}^T \otimes \boldsymbol{B}^T), \tag{38}$$

$$\boldsymbol{Y}_{(2)} \simeq \boldsymbol{B}\boldsymbol{G}_{(2)}(\boldsymbol{C}^T \otimes \boldsymbol{A}^T), \tag{39}$$

$$\boldsymbol{Y}_{(3)} \simeq \boldsymbol{C}\boldsymbol{G}_{(3)}(\boldsymbol{B}^T \otimes \boldsymbol{A}^T), \tag{40}$$

$$\bar{\boldsymbol{y}} \simeq (\boldsymbol{C} \otimes \boldsymbol{B} \otimes \boldsymbol{A})\bar{\boldsymbol{g}}, \tag{41}$$

where $\boldsymbol{Y}_{(1)} \in \mathbb{R}_+^{I \times JK}$ and $\boldsymbol{G}_{(1)} \in \mathbb{R}_+^{R_1 \times R_2 R_3}$ are the mode-1 matrix forms of the tensors $\underline{\boldsymbol{Y}}$ and $\underline{\boldsymbol{G}}$, respectively. We define modes 2 and 3 similarly. If we set $\boldsymbol{\Xi} := \boldsymbol{G}_{(1)}(\boldsymbol{C}^T \otimes \boldsymbol{B}^T)$, the factorization $\boldsymbol{Y}_{(1)} \simeq \boldsymbol{A}\boldsymbol{\Xi}$ can be regarded as NMF. Thus, the nonnegative factor matrix $\boldsymbol{A}$ can be updated by NMF-based update rules. In a similar way, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\underline{\boldsymbol{G}}$ can also be updated by NMF-based update rules (e.g., by applying ALS) [19].

The CPD model is a special case of the Tucker model with $R_1 = R_2 = R_3 = R$ and diagonal tensor $\underline{\boldsymbol{G}} = \underline{\boldsymbol{\Lambda}} \in \mathbb{R}_+^{R \times R \times R}$ with entries λ_r on the main diagonal given by

$$\underline{\boldsymbol{Y}} \simeq \sum_{r=1}^{R} \lambda_r \boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r. \tag{42}$$

The CPD model is also called PARAFAC [15] or CANDECOMP [3]. It can be regarded as a straightforward tensor extension of NMF, since NMF can be rewritten using $\boldsymbol{Y} \simeq \sum_{r=1}^{R} \lambda_r \boldsymbol{a}_r \circ \boldsymbol{b}_r$. 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. In some cases, we assume $\boldsymbol{a}_r$, $\boldsymbol{b}_r$, and/or $\boldsymbol{c}_r$ are smooth, and approximate these as $\boldsymbol{\Phi}_a\boldsymbol{w}_r$, $\boldsymbol{\Phi}_b\boldsymbol{v}_r$, and/or $\boldsymbol{\Phi}_c\boldsymbol{h}_r$, respectively. The update rule for $\boldsymbol{w}_r$ can be applied to $\boldsymbol{v}_r$ and $\boldsymbol{h}_r$; hence, we only provide an update rule for $\boldsymbol{w}_r$ in Sections 4.1 and 4.2.

4.1. Tucker Model

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

$$\begin{aligned} &\underset{\underline{\boldsymbol{G}},\boldsymbol{W},\boldsymbol{B},\boldsymbol{C}}{\text{minimize}} \ ||\underline{\boldsymbol{Y}} - \underline{\boldsymbol{G}} \times_1 \boldsymbol{\Phi W} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C}||_F^2, \\ &\text{s.t. } \underline{\boldsymbol{G}} \geq 0, \ \boldsymbol{W} \geq 0, \ \boldsymbol{B} \geq 0, \ \boldsymbol{C} \geq 0. \end{aligned} \tag{43}$$

We call this the GRBF-NTD method. The key to this problem is the update of $\boldsymbol{W}$. This is because the other factors $(\boldsymbol{B}, \boldsymbol{C}, \underline{\boldsymbol{G}})$ can be updated by the ALS-based algorithm for nonnegative Tucker decomposition [24, 25, 27, 19]. Focusing on $\boldsymbol{W}$ and using a matrix form, the problem can be rewritten as

$$\underset{\boldsymbol{W}}{\text{minimize}} \, ||\boldsymbol{Y}_{(1)} - \boldsymbol{\Phi W G}_{(1)}(\boldsymbol{C}^T \otimes \boldsymbol{B}^T)||_F^2, \ \text{s.t. } \boldsymbol{W} \geq 0. \tag{44}$$

Setting $\boldsymbol{Z}_r := \boldsymbol{Y}_{(1)} - \sum_{k \neq r} \boldsymbol{\Phi w}_k \boldsymbol{\xi}_k^T$ and $\boldsymbol{\Xi} = [\boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_{R_1}]^T := \boldsymbol{G}_{(1)}(\boldsymbol{C}^T \otimes \boldsymbol{B}^T) \in \mathbb{R}^{R_1 \times JK}$, we consider the problem for $\boldsymbol{w}_r$ as

$$\underset{\boldsymbol{w}_r}{\text{minimize}} \ ||\boldsymbol{Z}_r - \boldsymbol{\Phi w}_r \boldsymbol{\xi}_r^T||_F^2, \ \text{s.t. } \boldsymbol{w}_r \geq 0. \tag{45}$$

Problem (45) is essentially equivalent to (12). Since the condition that $||\boldsymbol{\xi}_r|| = 1$ must be satisfied, the update rule for $\boldsymbol{w}_r$ is given by

$$\boldsymbol{\xi}_r \leftarrow [\boldsymbol{Z}_r^T \boldsymbol{\Phi w}_r]_+; \tag{46}$$

$$\boldsymbol{\xi}_r \leftarrow \boldsymbol{\xi}_r / ||\boldsymbol{\xi}_r||; \tag{47}$$

$$\boldsymbol{w}_r \leftarrow [\boldsymbol{w}_r \circledast (\boldsymbol{\Phi}^T \boldsymbol{Z}_r \boldsymbol{\xi}_r) \oslash (\boldsymbol{\Phi}^T \boldsymbol{\Phi w}_r)]_+. \tag{48}$$

Note that update rules (46) and (47) are important steps for the accurate update of $\boldsymbol{w}_r$.

Algorithm 2 summarizes the GRBF-NTD process. Lines 5–13 give the update procedure for $\boldsymbol{W}$. If we apply the function approximation to factor matrices $\boldsymbol{B}$ and/or $\boldsymbol{C}$, the 14th and/or 15th lines must be modified in a similar way to $\boldsymbol{W}$.

4.2. CP Model

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

$$\begin{aligned} \underset{\lambda_r, \boldsymbol{w}_r, \boldsymbol{b}_r, \boldsymbol{c}_r}{\text{minimize}} \quad & \left|\left| \underline{\boldsymbol{Y}} - \sum_{r=1}^{R} \lambda_r (\boldsymbol{\Phi w}_r) \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r \right|\right|_F^2, \\ \text{s.t.} \quad & \lambda_r \geq 0,\ \boldsymbol{w}_r \geq 0,\ \boldsymbol{b}_r \geq 0,\ \boldsymbol{c}_r \geq 0, \\ & ||\boldsymbol{\Phi w}_r|| = ||\boldsymbol{b}_r|| = ||\boldsymbol{c}_r|| = 1 \end{aligned} \tag{49}$$

for $r = 1,\ 2,\ ...,\ R$. We refer to this as the GRBF-NCPD method. Problem (49) can also be solved with the HALS algorithm. Setting $\underline{\boldsymbol{Y}}_r := \underline{\boldsymbol{Y}} - \sum_{k \neq r} \lambda_k (\boldsymbol{\Phi w}_k) \circ \boldsymbol{b}_k \circ \boldsymbol{c}_k$ and using the first-way matrix form, the sub-problem for $\boldsymbol{w}_r$ can be rewritten as

$$\begin{aligned} \underset{\boldsymbol{w}_r}{\text{minimize}} \quad & ||\boldsymbol{Y}_{r(1)} - \lambda_r (\boldsymbol{\Phi w}_r)(\boldsymbol{c}_r^T \otimes \boldsymbol{b}_r^T)||_F^2, \\ \text{s.t.} \quad & ||\boldsymbol{\Phi w}_r|| = 1, \boldsymbol{w}_r \geq 0. \end{aligned} \tag{50}$$

By analogy to the relation between (12) and (17), problem (50) can be transformed to

$$\begin{aligned} \underset{\boldsymbol{w}_r}{\text{minimize}} \quad & ||\boldsymbol{Y}_{r(1)} (\boldsymbol{c}_r \otimes \boldsymbol{b}_r) - \lambda_r \boldsymbol{\Phi w}_r||_F^2, \\ \text{s.t.} \quad & ||\boldsymbol{\Phi w}_r|| = 1, \boldsymbol{w}_r \geq 0, \end{aligned} \tag{51}$$

where we have $(\boldsymbol{c}_r^T \otimes \boldsymbol{b}_r^T)(\boldsymbol{c}_r \otimes \boldsymbol{b}_r) = (\boldsymbol{c}_r^T \boldsymbol{c}_r \otimes \boldsymbol{b}_r^T \boldsymbol{b}_r) = 1$. Since there is a constraint $||\boldsymbol{\Phi w}_r|| = 1$ in (51), its solution is not dependent on λ_r. Thus, the update rule for $\boldsymbol{w}_r$ is finally given by

$$\boldsymbol{w}_r \leftarrow [\boldsymbol{w}_r \circledast \{\boldsymbol{\Phi}^T \boldsymbol{Y}_{r(1)} (\boldsymbol{c}_r \otimes \boldsymbol{b}_r)\} \oslash \{\boldsymbol{\Phi}^T \boldsymbol{\Phi w}_r\}]_+, \tag{52}$$

$$\boldsymbol{w}_r \leftarrow \boldsymbol{w}_r / ||\boldsymbol{\Phi w}_r||. \tag{53}$$

Algorithm 3 Fast GRBF-NCPD algorithm

1: **Input:** $\underline{\boldsymbol{Y}}$, R, and $\boldsymbol{\Phi}$
2: **Initialize:** $\boldsymbol{W}$ randomly, and $\boldsymbol{B}$, $\boldsymbol{C}$, $\underline{\boldsymbol{G}}$ by some initialization method for nonnegative CP decomposition
3: $\underline{\boldsymbol{E}} \leftarrow \underline{\boldsymbol{Y}} - \underline{\boldsymbol{G}} \times_1 \boldsymbol{\Phi}\boldsymbol{W} \times_2 \boldsymbol{B} \times_3 \boldsymbol{C}$;
4: **repeat**
5: **for** $r = 1, \ldots, R$ **do**
6: $\underline{\boldsymbol{Y}}_r \leftarrow \underline{\boldsymbol{E}} + \lambda_r(\boldsymbol{\Phi}\boldsymbol{w}_r) \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r$;
7: $\boldsymbol{w}_r \leftarrow [\boldsymbol{w}_r \circledast \{\boldsymbol{\Phi}^T(\underline{\boldsymbol{Y}}_r \times_2 \boldsymbol{b}_r^T \times_3 \boldsymbol{c}_r^T)\} \oslash \{\boldsymbol{\Phi}^T\boldsymbol{\Phi}\boldsymbol{w}_r\}]_+$;
8: $\boldsymbol{w}_r \leftarrow \boldsymbol{w}_r / ||\boldsymbol{\Phi}\boldsymbol{w}_r||$;
9: $\boldsymbol{b}_r \leftarrow [\underline{\boldsymbol{Y}}_r \times_1 (\boldsymbol{\Phi}\boldsymbol{w}_r)^T \times_3 \boldsymbol{c}_r^T]_+$;
10: $\boldsymbol{b}_r \leftarrow \boldsymbol{b}_r / ||\boldsymbol{b}_r||$;
11: $\boldsymbol{c}_r \leftarrow [\underline{\boldsymbol{Y}}_r \times_1 (\boldsymbol{\Phi}\boldsymbol{w}_r)^T \times_2 \boldsymbol{b}_r^T]_+$;
12: $\boldsymbol{c}_r \leftarrow \boldsymbol{c}_r / ||\boldsymbol{c}_r||$;
13: $\lambda_r \leftarrow [\underline{\boldsymbol{Y}}_r \times_1 (\boldsymbol{\Phi}\boldsymbol{w}_r)^T \times_2 \boldsymbol{b}_r^T \times_3 \boldsymbol{c}_r^T]_+$;
14: $\underline{\boldsymbol{E}} \leftarrow \underline{\boldsymbol{Y}}_r - \lambda_r(\boldsymbol{\Phi}\boldsymbol{w}_r) \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r$;
15: **end for**
16: **until** $||\boldsymbol{E}||_F^2$ converge
17: **Output:** $\boldsymbol{W}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\underline{\boldsymbol{G}}$

Algorithm 3 summarizes the GRBF-NCPD process. Lines 7–8 give the update procedure for $\boldsymbol{w}_r$. If we also apply the function approximation to $\boldsymbol{b}_r$ and/or $\boldsymbol{c}_r$, lines 9–10 and/or 11–12 should be modified in a similar way to $\boldsymbol{w}_r$.

5. Experiments

5.1. Algorithmic Comparison

In this experiment, we compare the proposed algorithm with the original form in terms of computation time. We selected 10 facial images of one subject from the Yale Face Database [1]. As a result, we obtained a dataset in the form of a (99×10) matrix. The small image size is necessary because the original algorithm is slow, even when factorizing such a small matrix. For various

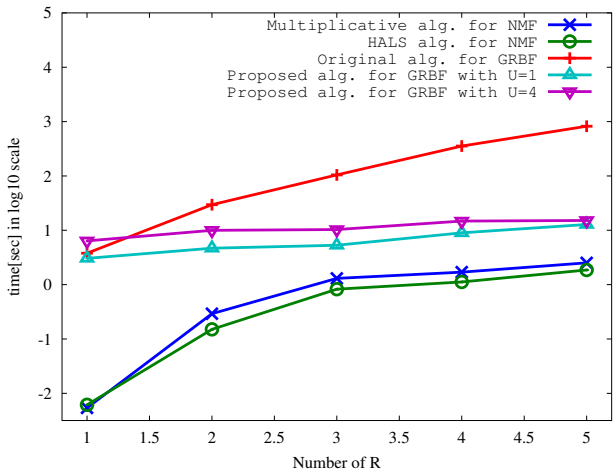


Figure 3: Computation time for various ranks

rank decompositions $R = \{1, 2, ..., 5\}$, we tested the following algorithms: multiplicative algorithm for NMF [20], HALS algorithm for NMF [8], the original GRBF-NMF algorithm, and the proposed GRBF-NMF algorithm. For GRBF-NMF, we set $\Delta t = \delta t = 1$, $\sigma = 1.0$, and $U = \{1, 4\}$. Figure 3 shows the average log 10-scale computation time over 10 runs for each rank decomposition. The computation time of the original algorithm increased exponentially. In contrast, the proposed algorithm required much less time to obtain a solution, and its computation time was relatively independent of U. The proposed algorithm is around 10–100 times faster than the original algorithm for $R = 3, 4, 5$. Although our proposed method may require many more iterations than the original algorithm, the overall computation time is dramatically improved by the new algorithm.

Next, we examined the convergence of the proposed algorithm. 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 $\boldsymbol{W}$ and $\boldsymbol{X}$ were varied at random. Figure 4 shows the functional boxplots [28] of the objective function values for all simulations. Simulations with values greater than 1.5 times the range of the central region

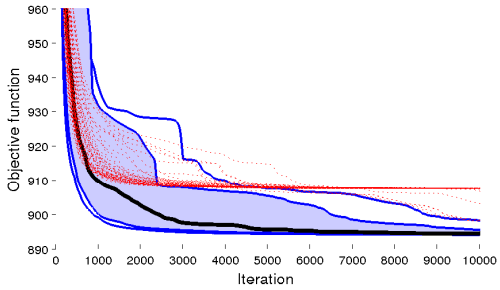


Figure 4: Functional boxplots of objective function for 1000 random simulations: black curve shows the median result, blue denotes the 50% central region, outer blue lines are maximum and minimum values for all non-outlying simulations, and red dashed lines are outlying simulations

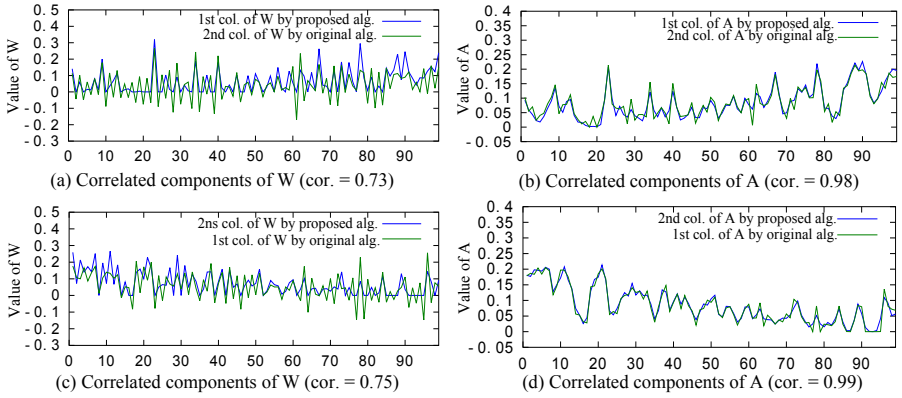


Figure 5: Estimates of $\boldsymbol{W}$ and $\boldsymbol{A}$ obtained by the proposed and original algorithms

were regarded as outliers. When the difference between backward and forward steps is smaller than $\epsilon = 10^{-3}$, the algorithm was assumed to have converged. From this criterion, we can see that all simulations converged to local optima. The average final value of the objective function for all simulations was 895.3 ± 3.0, and that for non-outlying simulations was 895.3 ± 0.2. The variation in the latter becomes very small. For all simulations and iterations, no increase in the objective function was observed. Thus, this experiment confirms the non-increasing property of the algorithm.

Next, we compared the estimates of $\boldsymbol{W}$ and $\boldsymbol{A}$ 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 algorithm (i.e., the same conditions). Fig. 5 shows the estimates of $\boldsymbol{W}$ and $\boldsymbol{A}$ obtained by the proposed algorithm and the original algorithm. Despite the difference between the constraints $\boldsymbol{W} \geq 0$ and $\boldsymbol{\Phi W} \geq 0$, the results for $\boldsymbol{A}$ are highly correlated. Furthermore, the proposed algorithm provides smoother components than the original algorithm. Many spikes in the estimates given by the original algorithm are smoothed by the proposed algorithm.

In the following sentences, GRBF-NMF stands for the method applying the proposed algorithm.

5.2. 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

$$\boldsymbol{Y} = [\boldsymbol{S}\boldsymbol{X}_0 + \boldsymbol{E}_0]_+, \tag{54}$$

where $\boldsymbol{S} \in \mathbb{R}_+^{I \times R}$ is an original source signal matrix, $\boldsymbol{X}_0 \in \mathbb{R}_+^{R \times J}$ is a mixing matrix, $\boldsymbol{E}_0 \in \mathbb{R}^{I \times J}$ is a Gaussian noise matrix, and $\boldsymbol{Y} \in \mathbb{R}_+^{I \times J}$ is an observed signal matrix. The signal-to-noise ratio (SNR) is defined as

$$\mathrm{SNR} := 10 \log_{10} \left[\frac{||\boldsymbol{S}\boldsymbol{X}_0||_F^2}{||\boldsymbol{S}\boldsymbol{X}_0 - \boldsymbol{Y}||_F^2} \right]. \tag{55}$$

Furthermore, we evaluated the estimated source $\boldsymbol{A} = \boldsymbol{\Phi W}$ using the mean signal-to-interference ratio (SIR) measure, which is calculated by Algorithm 4. This consists of several steps. First, each signal is normalized, because the NMF problem may not have a unique solution. Next, the SIR combination matrix is calculated as $\boldsymbol{M}(r_1, r_2) = \mathrm{SIR}(\boldsymbol{s}_{r_1}, \boldsymbol{a}_{r_2})$. For example, let $\boldsymbol{M}$ be

$$\boldsymbol{M} = \begin{pmatrix} 3.02 & 0.74 & 4.13 \\ 5.37 & 2.38 & 3.30 \\ 3.51 & 3.54 & 5.58 \end{pmatrix}. \tag{56}$$

Algorithm 4 Calculation of mean SIR

Input: $\boldsymbol{S}$, $\boldsymbol{A} \in \mathbb{R}^{I \times R}$

Initialize: $v = 0$

$\boldsymbol{s}_r \leftarrow \boldsymbol{s}_r / ||\boldsymbol{s}_r||$ for $r = 1, ..., R$;

$\boldsymbol{a}_r \leftarrow \boldsymbol{a}_r / ||\boldsymbol{a}_r||$ for $r = 1, ..., R$;

Calculate $\boldsymbol{M}(r_1, r_2) = 10 \log_{10} \left[\frac{||\boldsymbol{s}_{r_1}||^2}{||\boldsymbol{s}_{r_1} - \boldsymbol{a}_{r_2}||^2} \right]$ for all r_1, r_2;

for $r = 1, ..., R$ **do**

  $[r_1^*, r_2^*] \leftarrow \text{argmax}_{r_1, r_2} \boldsymbol{M}(r_1, r_2)$;

  $v \leftarrow v + \boldsymbol{M}(r_1^*, r_2^*)/R$;

  $\boldsymbol{M}(r_1^*, :) \leftarrow -\infty$;

  $\boldsymbol{M}(:, r_2^*) \leftarrow -\infty$;

end for

Output: v

The maximum element in this matrix is $\boldsymbol{M}(3, 3) = 5.58$. Next, $\boldsymbol{M}$ becomes

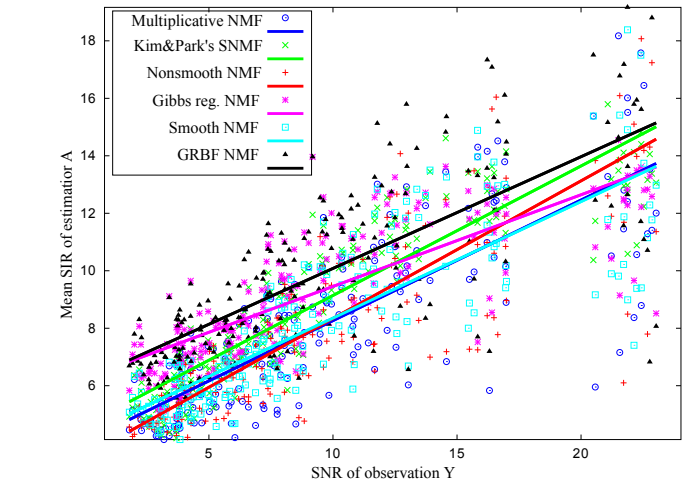
$$\boldsymbol{M} = \begin{pmatrix} 3.02 & 0.74 & -\infty \\ 5.37 & 2.38 & -\infty \\ -\infty & -\infty & -\infty \end{pmatrix}. \tag{57}$$

The maximum element is now $\boldsymbol{M}(2, 1) = 5.37$, and the next $\boldsymbol{M}$ is then

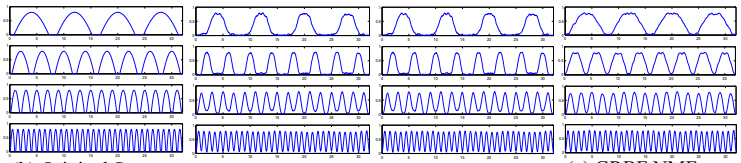
$$\boldsymbol{M} = \begin{pmatrix} -\infty & 0.74 & -\infty \\ -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty \end{pmatrix}. \tag{58}$$

Finally, the mean SIR is given by $(5.58 + 5.37 + 0.74)/3 = 3.90$. Actually, the maximum combination is given by $(5.37 + 4.13 + 3.54)/3 = 4.35$, but we need to calculate $R!$ combinations to obtain this solution. The mean SIR gives a useful approximation for the combinatorial maximization problem of the mean SIR for $\boldsymbol{A}$.

In the first BSS experiment, we used synthetic sparse and smooth nonnegative signals given by nonnegative sine curves and soft thresholding. Fig. 6 (b) shows the original sources $\boldsymbol{S} = [\boldsymbol{s}_1, ..., \boldsymbol{s}_R] \in \mathbb{R}_+^{I \times R}$. The individual lengths are



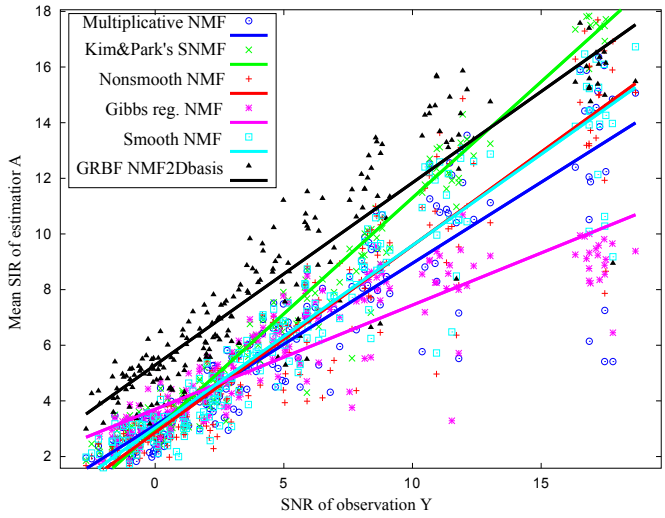
(a) Mark plots and linear regressions for various SNRs



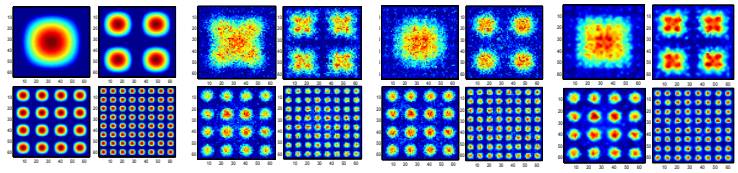
(b) Original Sources (c) Gibbs reg. NMF (12.6dB, 14.7 dB) (d) Kim&Park's SNMF (13.3 dB, 13.0 dB) (e) GRBF NMF (20.0 dB, 21.3dB)

Figure 6: Visualization of experimental results for synthetic signals

$R = 4$, $I = 1024$, and $J = 20$. Each element of the mixing matrix $\boldsymbol{X}_0$ is the absolute value of a sample generated from the normal distribution $\mathcal{N}(0, 1)$. We applied traditional and state-of-the-art NMF methods to the BSS problem with various noise levels: multiplicative NMF [20], Kim & Park's sparse NMF [18], nonsmooth NMF [23], Gibbs regularized NMF [35, 36], Chen's smooth NMF [4], and GRBF-NMF. The GRBF-NMF parameters were set to $U = 4$, $\delta t = 1$, and $\sigma = 1.0$. Figs. 6 (c), (d), and (e) illustrate the moving-average filtered signals after separation by the Gibbs regularized NMF, Kim & Park's SNMF, and GRBF-NMF, respectively. Each value in Figs. 6 (c)–(e) is the mean SIR of separated signals and filtered signals. Fig. 6 (a) shows the mean SIRs for all simulations and noise levels. The individual lines in Fig. 6 (a) 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.



(a) Mark plots and linear regressions for various SNRs



(b) Original Sources (c) Smooth NMF (9.7 dB) (d) Kim&Park's SNMF (10.9 dB) (e) GRBF NMF 2Dbasis (12.6 dB)

Figure 7: Visualization of experimental results for synthetic 2D signals

Next, we used the synthetic sparse and smooth nonnegative 2D signals illustrated in Fig. 7 (b). In this experiment, each signal was a $64 \times 64 = 4096$-dimensional vector. Figs. 7 (c), (d), and (e) illustrate the separated signals given by the smooth NMF, Kim & Park's SNMF, and GRBF-NMF-2Dbasis, respectively. Each value in Figs. 7 (c)–(e) is the mean SIR of the separated signals. Fig. 7 (a) plots the mean SIRs and the linear LSR results for various noise levels. Kim & Park's sparse NMF exhibited 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 non-mixed hyperspectral problem using four spectral signatures selected at random from the US Geological Survey (USGS) database. The angle between any pair of vectors $\{\boldsymbol{a}_i, \boldsymbol{a}_j\}$ is greater than 10°, and the reflectance values of the endmembers (i.e., source sig-

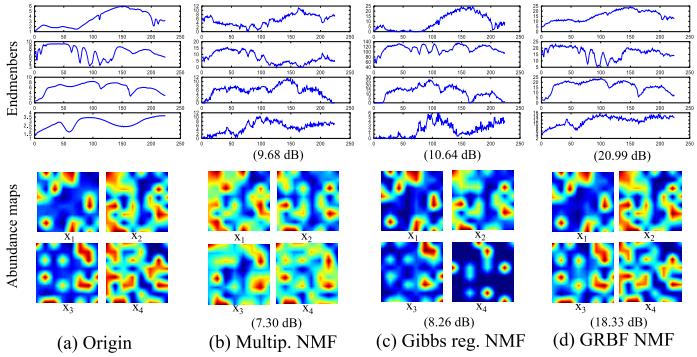


Figure 8: Endmember results and abundance maps

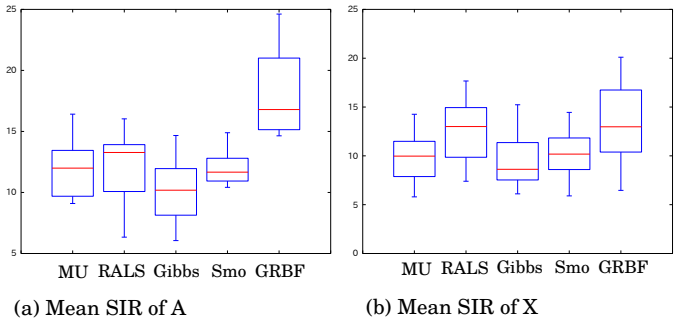


Figure 9: SIR_M statistics for matrix estimation (MU: multiplicative NMF, RALS: regularized ALS [7], Gibbs: Gibbs regularized NMF, Smo: Chen's smooth NMF, and GRBF: GRBF NMF)

nals $\boldsymbol{a}_r$) 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. We generated abundance (i.e., weight parameters $\boldsymbol{x}_r$) maps synthetically using a low-pass filtering strategy [22]; the resolution of each abundance map is 64×64 pixels. The mixtures are corrupted with an i.i.d zero-mean Gaussian noise with $SNR = 30\ dB$. The original endmembers and abundance maps are shown in Fig. 8 (a), and those estimated using multiplicative NMF [20], Gibbs regularized NMF [35, 36], and GRBF-NMF are presented in Figs. 8 (b)–(d), respectively. The mean SIR of these results was also evaluated. Fig. 9 illustrates the mean SIR statistics obtained for estimating matrices $\boldsymbol{A}$ (left) and $\boldsymbol{X}$ (right) with var-

ious NMF algorithms. The mean SIR samples were obtained for 20 uniformly distributed random initializations for the factors $\boldsymbol{A}^{(0)}$, $\boldsymbol{W}^{(0)}$, and $\boldsymbol{X}^{(0)}$ for each algorithm. From the results, we can see that GRBF-NMF outperforms the other NMF methods in terms of source estimation. There is no significant difference in the estimated mixture matrices given by RALS and GRBF. Since GRBF-NMF only imposes a nonnegativity constraint on $\boldsymbol{X}$, this implies that there is a possibility of further improvement by imposing some additional sparsity or norm-based regularization constraint.

5.3. Local Parts Analysis

The GRBF-block-NMF method was also applied to image analysis for parts-based feature extraction. The 3456×4608 image shown in Fig. 10 (b) was used with a noise level of 10 dB. The image can be transformed to a (1024×15552) nonnegative matrix by unfolding the individual 32×32 blocks. In this experiment, the GRBF-block-NMF was used to analyze local parts of this noisy image. Setting $R = 20$, the GRBF-block-NMF method extracts the 20 local parts-based feature images shown in Fig. 10 (c)–(h) (negative image). For comparison, we applied the standard multiplicative NMF [20], non-smooth NMF (nsNMF) [23], Chen's smooth NMF [4], and Gibbs regularized smooth NMF [35, 36]. Only the parts shown in Fig. 10 (c) were learned from the original noise-free image (a). This is used for the reference to evaluate the performances of the methods for parts-analysis in the presence of noise. The parts in (d)–(h) were learned from the noisy image (b), and mean SIRs between the reference parts and the individual obtained 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 SIR values: (d) 14.36 dB, (e) 16.65 dB, (f) 13.68 dB, (g) 14.42 dB, and (h) 16.99 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

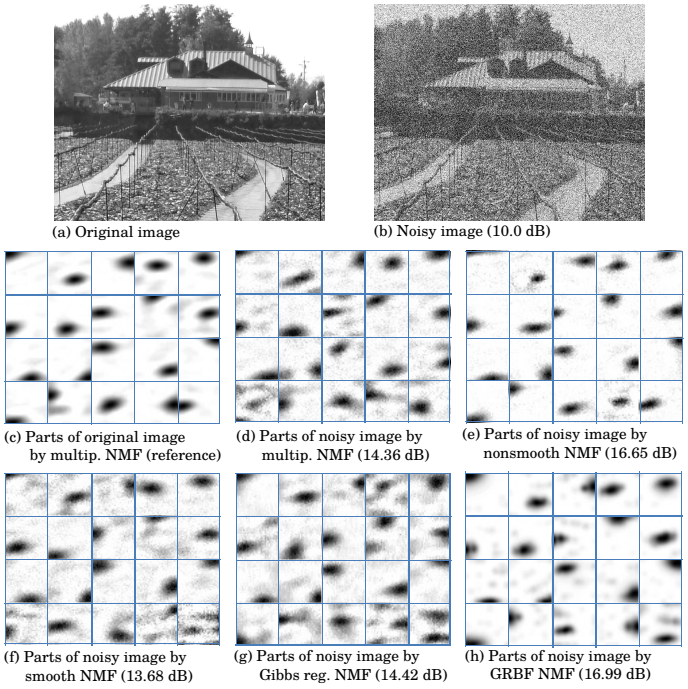


Figure 10: Parts-based feature extraction

known that sparseness works efficiently for the 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 $2048 \times 2048 \times 3$ color image (Fig. 11 (a)) and applied a noise level of 10 dB (Fig. 11 (b)). The data was separated into $64 \times 64 = 4096$ blocks, from which 20 local parts were extracted. Each part was a $32 \times 32 \times 3$ color image, and individual blocks in the image could be factorized as a linear combination of the 20 parts. We apply the matrix, CP, and Tucker factorization models. In the CP and Tucker models, we reformed the image as a $(32 \times 32 \times 3 \times 4092)$ tensor. In the matrix model, we reformed the image as a (3072×4092) matrix, and regarded each column as the vectorized form of a $(32 \times 32 \times 3)$ block. Multiplicative NMF, NCPD,

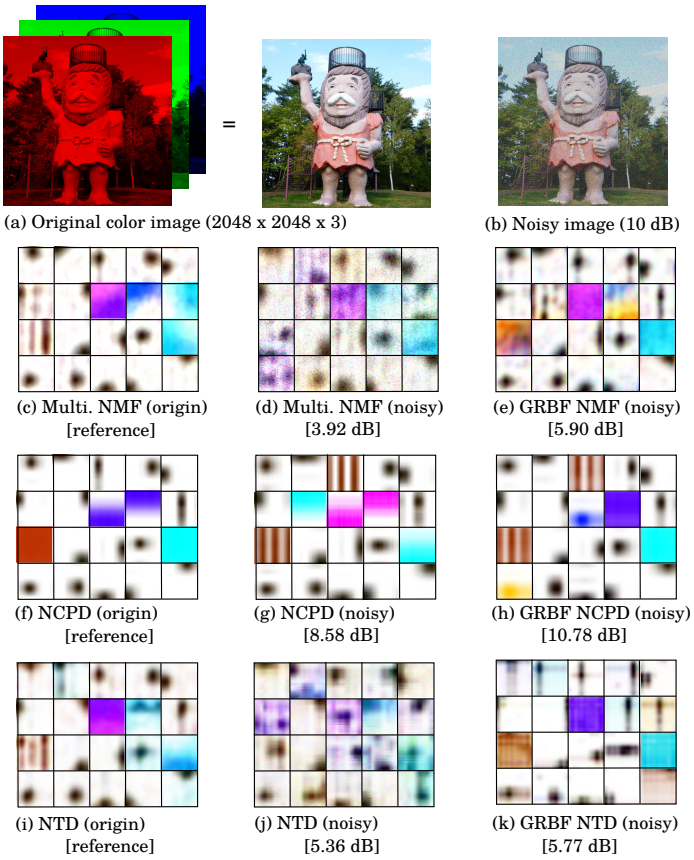


Figure 11: Parts-based feature extraction with a color image

NTD, GRBF-NMF-2Dbasis, 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, parts are given by the 1st–3rd factor matrices $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C})$ and a core tensor $(\underline{\boldsymbol{G}})$: the kth part is given by $g_k \boldsymbol{a}_k \circ \boldsymbol{b}_k \circ \boldsymbol{c}_k$ in CP decomposition, and $\sum_{r_1,r_2,r_3} g_{r_1 r_2 r_3 k} \boldsymbol{a}_{r_1} \circ \boldsymbol{b}_{r_2} \circ \boldsymbol{c}_{r_3}$ in Tucker decomposition. Fig. 11 (c)–(k) show the extracted local parts for each individual algorithm: Fig. 11 (c), (f), and (i) were 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 standard algorithms from the original image. From the mean SIR values, 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

Smooth NMF methods have been studied extensively, because smoothness and nonnegativity are physically meaningful 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 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 would appear to have promising applications in various areas of multi-way real-world data analysis, including brain, audio, and visual signal processing. For example, some brain signals are smooth in the time domain and sparse in the frequency domain. Thus, to apply our method to the analysis of brain signals, a combination of sparseness and smoothness is necessary. A proper combination with decorrelation, statistical independence, or other meaningful constraints should find some attractive applications.

6.2. Scalability Problem

In Section 3.4, we mentioned the scalability problem of $\mathbf{\Phi}$, and proposed a technique for reducing the matrix size. This was applied to the local parts analysis of a large-scale image. However, it still cannot be used for low-rank approximation and BSS problems when the dimension 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 the Updating

In this paper, we focused on reducing the computational cost of updating $\boldsymbol{X}$ and $\boldsymbol{W}$. Thus, the proposed update rules (13) and (19) are very simple and low cost. In 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 $\boldsymbol{X}$ and $\boldsymbol{W}$, giving better convergence to exact solutions with realistic computational costs. It will be worthwhile to use different options according to the situation.

7. Conclusions

In this paper, we proposed a new fast GRBF-NMF algorithm, multiple-σ 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. BSS and parts-based feature extraction experiments were conducted to compare the proposed methods with state-of-the-art NMF methods. The proposed algorithm does not require matrix inversion or mathematical programming techniques, such as QP optimization. Thus, it is much faster than the original algorithm (about 10–100 times faster). Moreover, we proposed the use of the 2D basis for unfolded 2D array data (i.e., vector signals), and demonstrated that the GRBF-NMF-2Dbasis algorithm gave improved results. Furthermore, the GRBF-based methods work well for Tucker and CP models with both single and multi-way smooth representations. In the BSS experiments, the GRBF method exhibited significantly better results compared to those of conventional nonnegative matrix factorization methods, including state-of-the-art methods for various noise levels. Finally, GRBF-block-NMF/NCPD/NTD also produced good results in the local parts-based feature extraction.

References

[1] Belhumeur, P., Hespanha, J., and Kriegman, D. (1997). Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720.

[2] Bro, R. (1998). *Multi-way analysis in the food industry: models, algorithms, and applications*. PhD thesis, University of Amsterdam, Amsterdam, Holland.

[3] Carroll, J. and Chang, J.-J. (1970). Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35:283–319.

[4] Chen, Z., Cichocki, A., and Rutkowski, T. M. (2006). Constrained non-negative matrix factorization method for EEG analysis in early detection of Alzheimer disease. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, 5:893–896.

[5] Cichocki, A., Amari, S.-i., et al. (2002). *Adaptive blind signal and image processing*. John Wiley Chichester.

[6] Cichocki, A. and Phan, A. (2009). Fast local algorithms for large scale non-negative matrix and tensor factorizations. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 92(3):708–721.

[7] Cichocki, A. and Zdunek, R. (2007). Regularized alternating least squares algorithms for non-negative matrix/tensor factorization. In *Advances in Neural Networks–ISNN 2007*, 793–802. Springer.

[8] Cichocki, A., Zdunek, R., and Amari, S.-i. (2007). Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization. In *Independent Component Analysis and Signal Separation*, 169–176. Springer.

[9] Cichocki, A., Zdunek, R., Phan, A. H., and Amari, S. (2009). *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation*. Wiley Publishing.

[10] Ding, C., Li, T., and Jordan, M.I. (2010). Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):45–55.

[11] Dong, Q. and Li, L. (2013) Smooth incomplete matrix factorization and its applications in image/video denoising. *Neurocomputing*, 112:458–469.

[12] Drakakis, K., Rickard, S., Frein, R. D., and Cichocki, A. (2008). Analysis of financial data using non-negative matrix factorization. In *International Mathematical Forum*, 38:1853–1870.

[13] Essid, S. and Fevotte, C. (2013). Smooth nonnegative matrix factorization for unsupervised audiovisual document structuring. *IEEE Transactions on Multimedia*, 15(2):415–425.

[14] Gillis, N. and Glineur, F. (2010) Using underapproximations for sparse nonnegative matrix factorization *Pattern Recognition*, 43(4):1676–1502. Elsevier.

[15] Harshman, R. (1970). Foundations of the parafac procedure: Model and conditions for an 'explanatory' multi-mode factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84.

[16] Hoyer, P. O. (2004). Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469.

[17] Jiang, L., and Yin, H. (2012). Bregman iteration algorithm for sparse non-negative matrix factorizations via alternating l_1-norm minimization. *Multidimensional Systems and Signal Processing*, 23(3):315–328.

[18] Kim, J. and Park, H. (2008). Sparse nonnegative matrix factorization for clustering. *Technical Report, Georgia Institute of Technology.*

[19] Kim, Y.-D. and Choi, S. (2007). Nonnegative Tucker decomposition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2007)*, 1–8.

[20] Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):789.

[21] Lee, D. D. and Seung, H. S. (2001). Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems*, 13:556–562.

[22] Miao, L. and Qi, H. (2007). Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 45(3):765–777.

[23] Pascual-Montano, A., Carazo, J. M., Kochi, K., Lehmann, D., and Pascual-Marqui, R. D. (2006). Nonsmooth nonnegative matrix factorization (nsNMF). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):403–415.

[24] Phan, A. and Cichocki, A. (2008). Fast and efficient algorithms for nonnegative Tucker decomposition. In *Advances in Neural Networks - ISNN 2008*, *Lecture Notes in Computer Science*, 5264:772–782. Springer Berlin Heidelberg.

[25] Phan, A. and Cichocki, A. (2009). Local learning rules for nonnegative Tucker decomposition. In *Neural Information Processing*, *Lecture Notes in Computer Science*, 5863:538–545. Springer Berlin Heidelberg.

[26] Phan, A. and Cichocki, A. (2010). Tensor decompositions for feature extraction and classification of high dimensional datasets. *IEICE, NOLTA*, 1(1):37–68.

[27] Phan, A. H. and Cichocki, A. (2011). Extended hals algorithm for nonnegative tucker decomposition and its applications for multiway analysis and classification. *Neurocomputing*, 74(11):1956–1969.

[28] Sun, Y. and Genton, M.G. (2011). Functional boxplots. *Journal of Computational and Graphical Statistics*, 20(2).

[29] Tucker, L. R. (1963). Implications of factor analysis of three-way matrices for measurement of change. In Harris, C. W., editor, *Problems in measuring change*, 122–137, University of Wisconsin Press.

[30] Van Benthem, M. H. and Keenan, M. R. (2004). Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems. *Journal of Chemometrics*, 18(10):441–450.

[31] Vapnik., V. (1998). *Statistical Learning Theory*. Wiley, New York.

[32] Watanabe, K., Hidaka, A., and Kurita, T. (2008). Automatic factorization of biological signals by using Boltzmann non-negative matrix factorization. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, IJCNN2008, 1122–1128.

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

[34] Zdunek, R. (2012). Approximation of feature vectors in nonnegative matrix factorization with Gaussian radial basis functions. In *Proceedings of the 19th International Conference on Neural Information Processing - Volume I*, ICONIP'12, 616–623. Springer-Verlag.

[35] Zdunek, R. and Cichocki, A. (2007). Gibbs regularized nonnegative matrix factorization for blind separation of locally smooth signals. In *15th IEEE International Workshop on Nonlinear Dynamics of Electronic Systems (NDES 2007)*, 317–320.

[36] Zdunek, R. and Cichocki, A. (2008). Blind image separation using non-negative matrix factorization with Gibbs smoothing. In *Neural Information Processing*, 519–528. Springer.