PAPER

Gauss-Seidel HALS Algorithm for Nonnegative Matrix Factorization with Sparseness and Smoothness Constraints

Takumi KIMURA[†a)], *Nonmember and* **Norikazu TAKAHASHI**[†b)], *Senior Member*

SUMMARY Nonnegative Matrix Factorization (NMF) with sparseness and smoothness constraints has attracted increasing attention. When these properties are considered, NMF is usually formulated as an optimization problem in which a linear combination of an approximation error term and some regularization terms must be minimized under the constraint that the factor matrices are nonnegative. In this paper, we focus our attention on the error measure based on the Euclidean distance and propose a new iterative method for solving those optimization problems. The proposed method is based on the Hierarchical Alternating Least Squares (HALS) algorithm developed by Cichocki et al. We first present an example to show that the original HALS algorithm can increase the objective value. We then propose a new algorithm called the Gauss-Seidel HALS algorithm that decreases the objective value monotonically. We also prove that it has the global convergence property in the sense of Zangwill. We finally verify the effectiveness of the proposed algorithm through numerical experiments using synthetic and real data.

key words: *nonnegative matrix factorization, hierarchical alternating least squares algorithm, Euclidean distance, global convergence, sparseness, smoothness*

1. Introduction

Nonnegative Matrix Factorization (NMF) [1]–[3] is an operation that decomposes a given $M \times N$ nonnegative matrix $\mathbf{X} = [X_{mn}]$ into an $M \times K$ nonnegative matrix $\mathbf{W} = [W_{mk}]$ and a $K \times N$ nonnegative matrix $\mathbf{H}^{\mathrm{T}} = [H_{nk}]^{\mathrm{T}}$ (see Fig. 1). Because NMF is useful for extraction of nonnegative bases and dimensionality reduction, it has found many applications in various fields such as face image processing [2], [4], text mining [5], recommender systems [6], [7], and so on.

NMF is formulated as a constrained optimization problem in which an error between $\mathbf{X}$ and $\mathbf{W}\mathbf{H}^{\mathrm{T}}$ must be minimized under the constraint that all entries of $\mathbf{W}$ and $\mathbf{H}$ are nonnegative. The Euclidean distance and various types of divergences have been used as the error criterion. The Euclidean distance-based NMF is formulated as the optimization problem:

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\left\|\mathbf{X} - \mathbf{W}\mathbf{H}^{\mathrm{T}}\right\|_{\mathrm{F}}^2 \\ &\text{subject to} && \mathbf{W} \geq \mathbf{0}_{M\times K},\ \mathbf{H} \geq \mathbf{0}_{N\times K} \end{aligned}$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm, that is,

$$\left\|\mathbf{X} - \mathbf{W}\mathbf{H}^{\mathrm{T}}\right\|_{\mathrm{F}}^2 = \sum_{m=1}^{M}\sum_{n=1}^{N}\left(X_{mn} - (\mathbf{W}\mathbf{H}^{\mathrm{T}})_{mn}\right)^2,$$

Manuscript received July 5, 2017.

[†]The authors are with Okayama University, Okayama-shi, 700-8530 Japan.

a) E-mail: kimu@momo.cs.okayama-u.ac.jp

b) E-mail: takahashi@cs.okayama-u.ac.jp

DOI: 10.1587/transfun.E100.A.2925

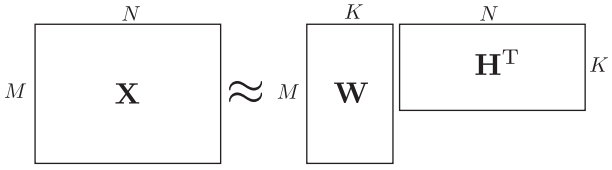

Fig. 1 Nonnegative matrix factorization.

$\mathbf{0}_{M\times K}$ ($\mathbf{0}_{N\times K}$, resp.) is the $M \times K$ ($N \times K$, resp.) matrix of all zeros and the inequality holds componentwise. In general, it is hard to find a global optimal solution of the NMF optimization problem because the objective function is not convex. So our goal is to find a local optimal solution.

The most popular algorithm for solving NMF optimization problems is the multiplicative update rules (MURs) developed by Lee and Seung [3]. They considered the Euclidean distance and the I-divergence as the error measure and derived update rules based on the idea of minimizing a strictly convex function called the auxiliary function instead of the objective function itself. Later on, this idea was generalized and applied to various types of error measures (see [8] for example). However, because the MURs are expressed in the form of a fraction, they are not defined for all pairs of nonnegative matrices $\mathbf{W}$ and $\mathbf{H}$. Also, the global convergence is not guaranteed because of this problem. By the global convergence, we mean that any sequence of solutions has at least one convergent subsequence and the limit of any convergent subsequence is a stationary point of the corresponding optimization problem [9]. In order to avoid the problem mentioned above, Gillis and Glineur [10] devised a modified version of the MUR for the Euclidean distance by using the idea of Cichocki et al. [11], which prevents variables from being less than a small positive constant. Furthermore, it was proved by Takahashi et al. that this modification guarantees the global convergence of many MURs [12]–[14].

The MURs have many good properties. They are simple and thus easy to implement. They are applicable to various types of error measures. Also, they have the global convergence property as mentioned above. However, the MURs are very slow in general. Therefore, during the last decades, many authors have developed faster algorithms for NMF [11], [15]–[17] which require less number of iterations than the MURs. Among them, the hierarchical alternating least squares (HALS) algorithm proposed by Cichocki et al. [11] is widely known as a simple and fast method for the

Euclidean distance based NMF. In this algorithm, $\mathbf{W}$ and $\mathbf{H}$ are partitioned into $2K$ blocks as $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K]$ and $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_K]$, and these $2K$ blocks are updated one by one in a cyclic manner. A similar method is rank-one residue iteration method proposed by Ho [16]. Also, Kim et al. [18] recently showed that some algorithms including the HALS algorithm can be derived using one common framework of the block coordinate descent method. Because the update rule of the HALS algorithm is expressed in the form of a fraction as in the case of the MURs, they are not defined for all pairs of nonnegative matrices $\mathbf{W}$ and $\mathbf{H}$. In order to solve this problem, Cichocki et al. [11] proposed a modified HALS algorithm, and recently this algorithm was proved to have the global convergence property [19].

In many applications of NMF, it is preferable that the obtained factor matrices are sparse or/and smooth. A matrix is said to be sparse if it has a small number of nonzero entries, while it is said to be smooth if neighboring entries take similar values. A simple way to control the sparseness and the smoothness is to add regularization terms representing the L^1 norm and the Frobenius norm of $\mathbf{W}$ and $\mathbf{H}$ to the objective function [20]–[24]. In this paper, we focus on the NMF optimization problem considered by Cichocki et al. [23], [24] which is described as

$$\begin{array}{ll} \text{minimize} & f(\mathbf{W}, \mathbf{H}) \\ \text{subject to} & \mathbf{W} \geq \mathbf{0}_{M\times K},\ \mathbf{H} \geq \mathbf{0}_{N\times K} \end{array} \tag{1}$$

where $f(\mathbf{W}, \mathbf{H})$ is given by

$$f(\mathbf{W}, \mathbf{H}) = \frac{1}{2}\left\|\mathbf{X} - \mathbf{W}\mathbf{H}^{\mathrm{T}}\right\|_{\mathrm{F}}^2 + \alpha_{\mathrm{sp}} \|\mathbf{H}\|_1 + \frac{\alpha_{\mathrm{sm}}}{2} \|\mathbf{L}\mathbf{H}\|_{\mathrm{F}}^2,$$

$\|\cdot\|_1$ denotes L^1 norm, that is,

$$\|\mathbf{H}\|_1 = \sum_{n=1}^{N}\sum_{k=1}^{K} |H_{nk}|.$$

Also, $\alpha_{\mathrm{sp}} > 0$ and $\alpha_{\mathrm{sm}} > 0$ are regularization parameters controlling the levels of sparseness and smoothness, respectively. As examples of $\mathbf{L}$, Cichocki et al. [24] presented

$$\mathbf{L}_1 = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & & \\ \vdots & & \ddots & \ddots & \\ 0 & & & 1 & -1 \end{bmatrix},$$

$$\mathbf{L}_2 = \begin{bmatrix} -1 & 2 & -1 & 0 & \cdots & 0 \\ 0 & -1 & 2 & -1 & & \\ \vdots & & \ddots & \ddots & \ddots & \\ 0 & & & -1 & 2 & -1 \end{bmatrix}$$

where $\mathbf{L}_1$ is a $(N-1)\times N$ matrix and $\mathbf{L}_2$ is a $(N-2)\times N$ matrix. However, $\mathbf{L}$ is not restricted to these specific matrices, but can be any $T \times N$ real matrix with $1 \leq T \leq N$.

The objective of this paper is to develop a global convergence guaranteed algorithm for solving (1) based on the HALS algorithm. We first introduce the HALS algorithm [23], [24] to solve (1). We then present an example to show that the original HALS algorithm can increase the objective value. We then propose a new algorithm called the Gauss-Seidel HALS (GSHALS) algorithm with which the objective value decreases monotonically, and prove that it has the global convergence property in the sense mentioned above. Finally, we verify the effectiveness of the proposed algorithm through numerical experiments using synthetic and real data.

Recently, various constraints have been considered for NMF (see [25] and references therein). Although we focus our attention only on the sparseness and smoothness constraints mentioned above, the idea behind the GSHALS algorithm may be useful for some other constraints. For example, the algorithm proposed by Liao and Zhang [26] for the graph regularized NMF [27], which has the same problem as the HALS algorithm, can be easily modified by using the same idea to guarantee the global convergence.

2. Nonnegative Matrix Factorization with Sparseness and Smoothness Constraints

2.1 HALS Algorithm

Let us partition matrices $\mathbf{W}$ and $\mathbf{H}$ into $2K$ blocks as $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K]$ and $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_K]$. Then (1) can be rewritten as follows:

$$\begin{array}{ll} \text{minimize} & \dfrac{1}{2}\left\|\mathbf{X} - \displaystyle\sum_{k=1}^{K} \mathbf{w}_k\mathbf{h}_k^{\mathrm{T}}\right\|_{\mathrm{F}}^2 \\ & +\alpha_{\mathrm{sp}} \displaystyle\sum_{k=1}^{K} \|\mathbf{h}_k\|_1 + \frac{\alpha_{\mathrm{sm}}}{2} \sum_{k=1}^{K} \|\mathbf{L}\mathbf{h}_k\|_2^2 \\ \text{subject to} & \mathbf{w}_k \geq \mathbf{0}_{M\times 1},\ \mathbf{h}_k \geq \mathbf{0}_{N\times 1}, \\ & k = 1, 2, \ldots, K. \end{array}$$

The HALS algorithm [23], [24] tries to minimize the objective value by updating $2K$ blocks $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K$, $\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_K$ one by one in a cyclic manner. Typical update orders are

$$\mathbf{w}_1 \to \mathbf{h}_1 \to \mathbf{w}_2 \to \mathbf{h}_2 \to \cdots \to \mathbf{w}_K \to \mathbf{h}_K \tag{2}$$

and

$$\mathbf{w}_1 \to \mathbf{w}_2 \to \cdots \to \mathbf{w}_K \to \mathbf{h}_1 \to \mathbf{h}_2 \to \cdots \to \mathbf{h}_K. \tag{3}$$

When updating $\mathbf{w}_k$, other $2K-1$ blocks are considered as constants and the optimization problem:

$$\begin{array}{ll} \text{minimize} & \phi_k(\mathbf{w}_k) = \dfrac{1}{2}\left\|\mathbf{R}_k^{\mathrm{T}} - \mathbf{h}_k\mathbf{w}_k^{\mathrm{T}}\right\|_{\mathrm{F}}^2 \\ \text{subject to} & \mathbf{w}_k \geq \mathbf{0}_{M\times 1} \end{array} \tag{4}$$

is solved, where

$$\mathbf{R}_k = \mathbf{X} - \sum_{\tilde{k}=1,\tilde{k}\neq k}^{K} \mathbf{w}_{\tilde{k}}\mathbf{h}_{\tilde{k}}^{\mathrm{T}}.$$

Similarly, when updating $\mathbf{h}_k$, other $2K-1$ blocks are considered as constants and the optimization problem:

$$\begin{array}{ll}\text{minimize} & \psi_k(\mathbf{h}_k) \\ \text{subject to} & \mathbf{h}_k \geq \mathbf{0}_{N\times 1}\end{array} \tag{5}$$

is solved, where $\psi_k(\mathbf{h}_k)$ is given by

$$\psi_k(\mathbf{h}_k) = \frac{1}{2}\left\|\mathbf{R}_k - \mathbf{w}_k\mathbf{h}_k^{\mathrm{T}}\right\|_{\mathrm{F}}^2 + \alpha_{\mathrm{sp}}\|\mathbf{h}_k\|_1 + \frac{\alpha_{\mathrm{sm}}}{2}\|\mathbf{L}\mathbf{h}_k\|_2^2.$$

As shown in [18], the problem (4) has a unique optimal solution given by $\mathbf{w}_k = [\mathbf{R}_k\mathbf{h}_k]_+/(\mathbf{h}_k^T\mathbf{h}_k)$ if $\mathbf{h}_k \neq \mathbf{0}_{N\times 1}$, where $[\mathbf{v}]_+$ denotes the vector obtained from $\mathbf{v}$ by replacing all negative entries with zero. As for the problem (5), we cannot obtain the explicit formula for the optimal solution, but it is easy to find the minimum point of $\psi_k(\mathbf{h}_k)$ by solving $\nabla\psi_k(\mathbf{h}_k) = \mathbf{0}_{N\times 1}$, that is,

$$(\mathbf{w}_k^{\mathrm{T}}\mathbf{w}_k\mathbf{I}_{N\times N} + \alpha_{\mathrm{sm}}\mathbf{L}^{\mathrm{T}}\mathbf{L})\mathbf{h}_k = \mathbf{R}_k^{\mathrm{T}}\mathbf{w}_k - \alpha_{\mathrm{sp}}\mathbf{1}_{N\times 1}, \tag{6}$$

where $\mathbf{I}_{N\times N}$ is the $N\times N$ identity matrix and $\mathbf{1}_{N\times 1}$ is the N-dimensional column vector of all ones. Suppose that $\mathbf{w}_k \neq \mathbf{0}_{N\times 1}$. Then $\mathbf{w}_k^{\mathrm{T}}\mathbf{w}_k\mathbf{I}_{N\times N} + \alpha_{\mathrm{sm}}\mathbf{L}^{\mathrm{T}}\mathbf{L}$ is nonsingular and hence, multiplying both sides of (6) by $(\mathbf{w}_k^{\mathrm{T}}\mathbf{w}_k\mathbf{I}_{N\times N} + \alpha_{\mathrm{sm}}\mathbf{L}^{\mathrm{T}}\mathbf{L})^{-1}$ from the left, we have

$$\mathbf{h}_k = (\mathbf{w}_k^{\mathrm{T}}\mathbf{w}_k\mathbf{I}_{N\times N} + \alpha_{\mathrm{sm}}\mathbf{L}^{\mathrm{T}}\mathbf{L})^{-1}(\mathbf{R}_k^{\mathrm{T}}\mathbf{w}_k - \alpha_{\mathrm{sp}}\mathbf{1}_{N\times 1}). \tag{7}$$

The right-hand side may have one or more negative entries, but we can obtain a nonnegative vector from it by applying the operator $[\cdot]_+$.

The HALS algorithm is based on the above idea, and described by the following update rule:

$$\mathbf{w}_k \leftarrow \frac{[\mathbf{R}_k\mathbf{h}_k]_+}{\mathbf{h}_k^{\mathrm{T}}\mathbf{h}_k}, \tag{8}$$

$$\mathbf{h}_k \leftarrow [(\mathbf{w}_k^{\mathrm{T}}\mathbf{w}_k\mathbf{I}_{N\times N} + \alpha_{\mathrm{sm}}\mathbf{L}^{\mathrm{T}}\mathbf{L})^{-1} \times (\mathbf{R}_k^{\mathrm{T}}\mathbf{w}_k - \alpha_{\mathrm{sp}}\mathbf{1}_{N\times 1})]_+. \tag{9}$$

Let the solution after l (≥ 0) rounds of updates be denoted by $(\mathbf{W}^{(l)}, \mathbf{H}^{(l)})$. When using update rule (8) and (9), it could occur that one or more columns of $\mathbf{W}^{(l)}$ and $\mathbf{H}^{(l)}$ become zero for some l. If $\mathbf{w}_k = \mathbf{0}_{M\times 1}$ and $\mathbf{L} = \mathbf{L}_2$ for instance, $\mathbf{w}_k^{\mathrm{T}}\mathbf{w}_k\mathbf{I}_{N\times N} + \alpha_{\mathrm{sm}}\mathbf{L}^{\mathrm{T}}\mathbf{L}$ in the right-hand side of (9) is not invertible. If $\mathbf{h}_k = \mathbf{0}_{N\times 1}$, the denominator of the right-hand side of (8) becomes zero. In these cases, the algorithm has to be stopped before the solution is obtained. In order to avoid this situation, Cichocki et al. [24] used the update rule expressed by

$$\mathbf{w}_k \leftarrow \left[\frac{\mathbf{R}_k\mathbf{h}_k}{\mathbf{h}_k^{\mathrm{T}}\mathbf{h}_k}\right]_{\epsilon+}, \tag{10}$$

$$\mathbf{h}_k \leftarrow [(\mathbf{w}_k^{\mathrm{T}}\mathbf{w}_k\mathbf{I}_{N\times N} + \alpha_{\mathrm{sm}}\mathbf{L}^{\mathrm{T}}\mathbf{L})^{-1} \times (\mathbf{R}_k^{\mathrm{T}}\mathbf{w}_k - \alpha_{\mathrm{sp}}\mathbf{1}_{N\times 1})]_{\epsilon+} \tag{11}$$

instead of (8) and (9), where $[\mathbf{v}]_{\epsilon+}$ denotes the vector obtained from $\mathbf{v}$ by replacing all entries less than ϵ with ϵ.

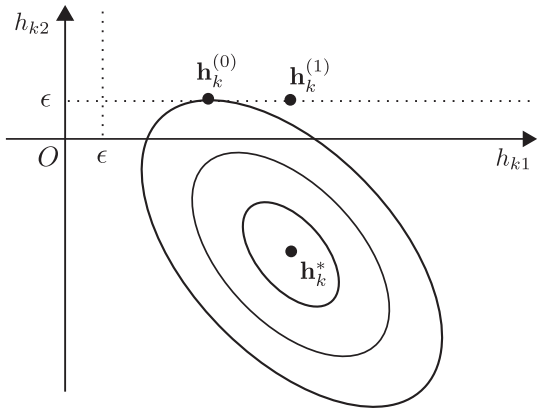

Fig. 2 Situation where $\psi_k(\mathbf{h}_k)$ is increased by the HALS algorithm.

This notation is also used for scalars and matrices in later discussion. When the update rule given by (10) and (11) is used, we have to consider a modified optimization problem:

$$\begin{array}{ll}\text{minimize} & f(\mathbf{W},\mathbf{H}) \\ \text{subject to} & \mathbf{W} \geq \epsilon\mathbf{1}_{M\times K},\ \mathbf{H} \geq \epsilon\mathbf{1}_{N\times K},\end{array} \tag{12}$$

where $\mathbf{1}_{M\times K}$ ($\mathbf{1}_{N\times K}$) denotes the $M\times K$ ($N\times K$) matrix of all ones. In the following sections, by the HALS algorithm, we mean the update rule given by (10) and (11).

2.2 Problem of HALS Algorithm

If $\mathbf{w}_k \geq \epsilon\mathbf{1}_{M\times 1}$ then the right-hand side of (7) is the minimum point of $\psi_k(\mathbf{h}_k)$. However, it is not always true that the right-hand side of (11) is the optimal solution of the optimization problem:

$$\begin{array}{ll}\text{minimize} & \psi_k(\mathbf{h}_k) \\ \text{subject to} & \mathbf{h}_k \geq \epsilon\mathbf{1}_{N\times 1}.\end{array} \tag{13}$$

To make matters worse, it may occur that the value of ψ_k is increased by the update (11). To see this, let us consider the situation shown in Fig. 2 where $\mathbf{h}_k^{(0)}$ represents the current solution and $\mathbf{h}_k^*$ is the minimum point of $\psi_k(\mathbf{h}_k)$. Because the second entry of $\mathbf{h}_k^*$ is less than ϵ, (11) returns $\mathbf{h}_k^{(1)}$ as the new solution. However, looking at contours of ψ_k, we see that $\psi_k(\mathbf{h}_k^{(1)})$ is greater than $\psi_k(\mathbf{h}_k^{(0)})$.

We now give a more concrete example to show that the update (11) can increase the value of ψ_k. Let $M = 1$, $N = 3$, $K = 1$, $\mathbf{X} = [3, 2, 1]$, $\mathbf{L} = [-1, 2, -1]$, $\alpha_{\mathrm{sp}} = 3/2$, $\alpha_{\mathrm{sm}} = 1$, $\epsilon = 1$, $\mathbf{w}_1^{(0)} = [1]$ and $\mathbf{h}_1^{(0)} = [6/5, 1, 1]^{\mathrm{T}}$. Under this setting, we have $\psi_k(\mathbf{h}_1^{(0)}) = 6.94$. If the value of $\mathbf{h}_1$ is updated by (11), we have $\mathbf{h}_1^{(1)} = [3/2, 1, 1]^{\mathrm{T}}$ and $\psi_k(\mathbf{h}_1^{(1)}) = 7$, which is greater than $\psi_k(\mathbf{h}_1^0)$.

3. GSHALS Algorithm

3.1 Derivation of Update Rule

In order to solve the problem of the HALS algorithm pointed

out in the previous section, we propose to update entries of $\mathbf{h}_k$ one by one instead of using (11). In the following discussion, N entries of $\mathbf{h}_k$ are denoted by $h_{k1}, h_{k2} \ldots, h_{kN}$ instead of $H_{1k}, H_{2k}, \ldots, H_{Nk}$. In addition, let the n-th columns of $\mathbf{X}$ and $\mathbf{R}_k$ be denoted by $\mathbf{x}_n$ and $\mathbf{r}_{kn}$. Here, $\mathbf{r}_{kn}$ is given by

$$\mathbf{r}_{kn} = \mathbf{x}_n - \sum_{\tilde{k}=1,\tilde{k}\neq k}^{K} \mathbf{w}_{\tilde{k}} h_{\tilde{k}n}.$$

Note that we do not focus on some specific update order of entries of $\mathbf{h}_k$, but only assume that the update order is fixed during the execution of the proposed algorithm. When updating h_{kn}, other $N-1$ entries are considered as constants and the optimization problem:

$$\begin{array}{ll} \text{minimize} & \psi_{kn}(h_{kn}) \\ \text{subject to} & h_{kn} \geq \epsilon \end{array} \tag{14}$$

is solved, where

$$\psi_{kn}(h_{kn}) = \frac{1}{2}\|\mathbf{r}_{kn} - \mathbf{w}_k h_{kn}\|_2^2 + \alpha_{\mathrm{sp}} h_{kn} + \frac{\alpha_{\mathrm{sm}}}{2} \sum_{t=1}^{T} \Big(\sum_{\tilde{n}=1}^{N} L_{t\tilde{n}} h_{k\tilde{n}} \Big)^2.$$

Let h_{kn}^* be the minimum point of $\psi_{kn}(h_{kn})$. If h_{kn}^* is greater than or equal to ϵ then it is the optimal solution of (14) because $\psi_{kn}(h_{kn})$ is strictly convex (see Fig. 3(a)). Otherwise, the optimal solution of (14) is ϵ (see Fig. 3(b)). These observations can be summarized in the following lemma.

Lemma 1: If $\mathbf{w}_k$ is positive then the optimization problem (14) has a unique optimal solution given by

$$h_{kn} = \left[\frac{\mathbf{r}_{kn}^{\mathrm{T}} \mathbf{w}_k - \alpha_{\mathrm{sp}} - \alpha_{\mathrm{sm}} \sum_{t=1}^{T} L_{tn} \sum_{\tilde{n}=1,\tilde{n}\neq n}^{N} L_{t\tilde{n}} h_{k\tilde{n}}}{\mathbf{w}_k^{\mathrm{T}} \mathbf{w}_k + \alpha_{\mathrm{sm}} \sum_{t=1}^{T} L_{tn}^2} \right]_{\epsilon+}. \tag{15}$$

Proof : The solution of the equation

$$\psi'_{kn}(h_{kn}) = (\mathbf{w}_k^{\mathrm{T}} \mathbf{w}_k) h_{kn} - \mathbf{r}_{kn}^{\mathrm{T}} \mathbf{w}_k + \alpha_{\mathrm{sp}} + \alpha_{\mathrm{sm}} \sum_{t=1}^{T} L_{tn} \sum_{\tilde{n}=1}^{N} L_{t\tilde{n}} h_{k\tilde{n}} = 0$$

is given by

$$h_{kn} = \frac{\mathbf{r}_{kn}^{\mathrm{T}} \mathbf{w}_k - \alpha_{\mathrm{sp}} - \alpha_{\mathrm{sm}} \sum_{t=1}^{T} L_{tn} \sum_{\tilde{n}=1,\tilde{n}\neq n}^{N} L_{t\tilde{n}} h_{k\tilde{n}}}{\mathbf{w}_k^{\mathrm{T}} \mathbf{w}_k + \alpha_{\mathrm{sm}} \sum_{t=1}^{T} L_{tn}^2}$$

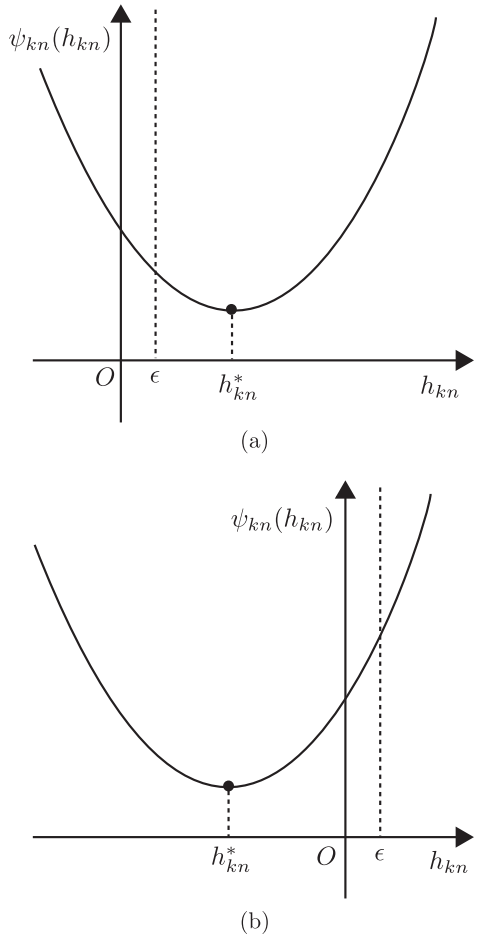

Fig. 3 The minimum point h_{kn}^* of $\psi_{kn}(h_{kn})$. (a) The case where $h_{kn}^* \geq \epsilon$. (b) The case where $h_{kn}^* < \epsilon$.

that minimizes $\psi_{kn}(h_{kn})$ because $\psi_{kn}(h_{kn})$ is strictly convex. If this is greater than or equal to ϵ then it is a unique optimal solution of (14). Otherwise, ϵ is a unique optimal solution of (14) because $\psi_{kn}(h_{kn})$ is strictly monotone increasing in $[\epsilon, \infty)$. Therefore, the optimal solution of (14) is given by (15). □

From Lemma 1, we obtain the update rule for entries of $\mathbf{h}_k$, which is expressed as

$$h_{kn} \leftarrow \left[\frac{\mathbf{r}_{kn}^{\mathrm{T}} \mathbf{w}_k - \alpha_{\mathrm{sp}} - \alpha_{\mathrm{sm}} \sum_{t=1}^{T} L_{tn} \sum_{\tilde{n}=1,\tilde{n}\neq n}^{N} L_{t\tilde{n}} h_{k\tilde{n}}}{\mathbf{w}_k^{\mathrm{T}} \mathbf{w}_k + \alpha_{\mathrm{sm}} \sum_{t=1}^{T} L_{tn}^2} \right]_{\epsilon+}. \tag{16}$$

It is clear from the above discussion that the objective value of (12) is not increased by the update (16).

Note that entries of $\mathbf{h}_k$ are updated one by one by (16) like the Gauss-Seidel method (see [28] for example) for solving linear equations. We thus call this algorithm the Gauss-Seidel HALS (GSHALS) algorithm.

3.2 Global Convergence of GSHALS Algorithm

In this section, we prove that the GSHALS algorithm has the global convergence property in the sense of Zangwill [9]. Let the feasible region and the set of stationary points of (12) be denoted by $\mathcal{F}_\epsilon$ and $\mathcal{S}_\epsilon$, respectively. A point $(\mathbf{W}, \mathbf{H}) \in \mathcal{F}_\epsilon$ is called a stationary point if it satisfies the following conditions:

$$\nabla_{\mathbf{W}} f(\mathbf{W}, \mathbf{H}) \geq \mathbf{0}_{M\times K}, \tag{17}$$
$$\nabla_{\mathbf{H}} f(\mathbf{W}, \mathbf{H}) \geq \mathbf{0}_{N\times K}, \tag{18}$$
$$\nabla_{\mathbf{W}} f(\mathbf{W}, \mathbf{H}) \odot (\epsilon \mathbf{1}_{M\times K} - \mathbf{W}) = \mathbf{0}_{M\times K}, \tag{19}$$
$$\nabla_{\mathbf{H}} f(\mathbf{W}, \mathbf{H}) \odot (\epsilon \mathbf{1}_{N\times K} - \mathbf{H}) = \mathbf{0}_{N\times K}, \tag{20}$$

where

$$\begin{aligned}\nabla_{\mathbf{W}} f(\mathbf{W}, \mathbf{H}) =& (\mathbf{W}\mathbf{H}^{\mathrm{T}} - \mathbf{X})\mathbf{H},\\ \nabla_{\mathbf{H}} f(\mathbf{W}, \mathbf{H}) =& (\mathbf{H}\mathbf{W}^{\mathrm{T}} - \mathbf{X}^{\mathrm{T}})\mathbf{W}\\ &+ \alpha_{\mathrm{sp}} \mathbf{1}_{N\times K} + \alpha_{\mathrm{sm}} \mathbf{L}^{\mathrm{T}}\mathbf{L}\mathbf{H}\end{aligned}$$

and $\odot$ represents componentwise multiplication.

In order to make discussions simple, we define one round of updates of $2K$ blocks by using (10) and (16) as a mapping $A : \mathcal{F}_\epsilon \to \mathcal{F}_\epsilon$. Then the solution after l rounds of updates is expressed as

$$\begin{aligned}(\mathbf{W}^{(l)}, \mathbf{H}^{(l)}) &= A(\mathbf{W}^{(l-1)}, \mathbf{H}^{(l-1)})\\ &= A^2(\mathbf{W}^{(l-2)}, \mathbf{H}^{(l-2)})\\ &\vdots\\ &= A^l(\mathbf{W}^{(0)}, \mathbf{H}^{(0)}).\end{aligned}$$

Note that we do not focus on some specific update order such as (2) and (3), but only assume that the update order is fixed during the execution of the algorithm.

The global convergence property of the GSHALS algorithm is stated as follows.

Theorem 1: For any positive constant ϵ and initial solution $(\mathbf{W}^{(0)}, \mathbf{H}^{(0)}) \in \mathcal{F}_\epsilon$, the sequence $\{(\mathbf{W}^{(l)}, \mathbf{H}^{(l)})\}_{l=0}^{\infty}$ generated by the GSHALS algorithm expressed by (10) and (16) has at least one convergent subsequence and the limit of any convergent subsequence belongs to $\mathcal{S}_\epsilon$.

In the rest of this section, we prove Theorem 1 by using Zangwill's global convergence theorem [9]. Namely, we show that the following statements hold true.

1. For any initial solution $(\mathbf{W}^{(0)}, \mathbf{H}^{(0)}) \in \mathcal{F}_\epsilon$, the sequence $\{(\mathbf{W}^{(l)}, \mathbf{H}^{(l)})\}_{l=0}^{\infty} = \{A^l(\mathbf{W}^{(0)}, \mathbf{H}^{(0)})\}_{l=0}^{\infty}$ is contained in a closed bounded subset of $\mathcal{F}_\epsilon$.
2. The mapping A and the objective function f satisfy the following statements.
   a. If $(\mathbf{W}, \mathbf{H}) \in \mathcal{F}_\epsilon \setminus \mathcal{S}_\epsilon$ then $f(A(\mathbf{W}, \mathbf{H})) < f(\mathbf{W}, \mathbf{H})$.
   b. If $(\mathbf{W}, \mathbf{H}) \in \mathcal{S}_\epsilon$ then $f(A(\mathbf{W}, \mathbf{H})) \leq f(\mathbf{W}, \mathbf{H})$.
3. The mapping A is continuous in $\mathcal{F}_\epsilon \setminus \mathcal{S}_\epsilon$.

It is clear from the update rule given by (10) and (16) that the third statement is true. Therefore, we prove that the remaining two statements hold.

First, we prove the second statement by the following two lemmas. The first lemma shows the relation between $\mathcal{S}_\epsilon$ and the optimal solutions of the subproblems. The second one proves that f is strictly decreased by the mapping A if the current solution is not a stationary point.

Lemma 2: $(\mathbf{W}^*, \mathbf{H}^*) = ((\mathbf{w}_1^*, \ldots, \mathbf{w}_K^*), (\mathbf{h}_1^*, \ldots, \mathbf{h}_K^*)) \in \mathcal{F}_\epsilon$ is a stationary point of (12) if and only if $\mathbf{w}_k^*$ is a unique optimal solution of the optimization problem:

$$\begin{aligned}&\text{minimize} && \phi_k(\mathbf{w}_k)\\ &\text{subject to} && \mathbf{w}_k \geq \epsilon \mathbf{1}_{M\times 1}\end{aligned} \tag{21}$$

for $k = 1, 2, \ldots, K$ and h_{kn}^* is a unique optimal solution of the optimization problem:

$$\begin{aligned}&\text{minimize} && \psi_{kn}(h_{kn})\\ &\text{subject to} && h_{kn} \geq \epsilon\end{aligned} \tag{22}$$

for $k = 1, 2, \ldots, K$ and $n = 1, 2, \ldots, N$.

Proof : The Lagrangian function for the optimization problem (21) is given by

$$L(\mathbf{w}_k, \boldsymbol{\lambda}) = \phi_k(\mathbf{w}_k) + \boldsymbol{\lambda}^{\mathrm{T}}(\epsilon \mathbf{1}_{M\times 1} - \mathbf{w}_k)$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_M)^T$ is the Lagrange multiplier vector. Then $\mathbf{w}_k^* \geq \epsilon \mathbf{1}_{M\times 1}$ is a stationary point of (21) if and only if there exists a $\boldsymbol{\lambda}$ that satisfies the following conditions:

$$\frac{\partial L}{\partial \mathbf{w}_k}(\mathbf{w}_k^*, \boldsymbol{\lambda}) = \mathbf{0}_{M\times 1}, \tag{23}$$
$$\boldsymbol{\lambda} \odot (\epsilon \mathbf{1}_{M\times 1} - \mathbf{w}_k^*) = \mathbf{0}_{M\times 1}, \tag{24}$$
$$\boldsymbol{\lambda} \geq \mathbf{0}_{M\times 1}. \tag{25}$$

Here, the left-hand side of (23) is given by

$$\begin{aligned}\frac{\partial L}{\partial \mathbf{w}_k}(\mathbf{w}_k^*, \boldsymbol{\lambda}) &= \left(\mathbf{w}_k^* \mathbf{h}_k^{*\mathrm{T}} + \sum_{\tilde{k}\neq k} \mathbf{w}_{\tilde{k}}^* \mathbf{h}_{\tilde{k}}^{*\mathrm{T}} - \mathbf{X}\right)\mathbf{h}_k^* - \boldsymbol{\lambda}\\ &= (\mathbf{W}^*\mathbf{H}^{*\mathrm{T}} - \mathbf{X})\mathbf{h}_k^* - \boldsymbol{\lambda}.\end{aligned}$$

So the conditions (23)–(25) can be rewritten as

$$(\mathbf{W}^*\mathbf{H}^{*\mathrm{T}} - \mathbf{X})\mathbf{h}_k^* \geq \mathbf{0}_{M\times 1},$$

$$\left((\mathbf{W}^*\mathbf{H}^{*\mathrm{T}} - \mathbf{X})\mathbf{h}_k^*\right) \odot (\epsilon \mathbf{1}_{M\times 1} - \mathbf{w}_k^*) = \mathbf{0}_{M\times 1}.$$

Note that the problem (21) has a unique stationary point and it is also a unique optimal solution because $\phi_k(\mathbf{w}_k)$ is strictly convex. Therefore, $\mathbf{w}_k^* \geq \epsilon \mathbf{1}_{M\times 1}$ is a unique optimal solution of (21) for $k = 1, 2, \ldots, K$ if and only if

$$(\mathbf{W}^*\mathbf{H}^{*\mathrm{T}} - \mathbf{X})\mathbf{H}^* \geq \mathbf{0}_{M\times K}, \tag{26}$$

$$\left((\mathbf{W}^*\mathbf{H}^{*\mathrm{T}} - \mathbf{X})\mathbf{H}^*\right) \odot (\epsilon \mathbf{1}_{M\times K} - \mathbf{W}^*) = \mathbf{0}_{M\times K}. \tag{27}$$

The Lagrangian function for the optimization problem (22) is given by

$$L(h_{kn}, \mu) = \psi_{kn}(h_{kn}) + \mu(\epsilon - h_{kn})$$

where μ is the Lagrange multiplier. Then $h^*_{kn} \geq \epsilon$ is a stationary point of (22) if and only if there exists a μ that satisfies the following conditions:

$$\frac{\partial L}{\partial h_{kn}}(h^*_{kn}, \mu) = 0, \tag{28}$$

$$\mu \odot (\epsilon - h^*_{kn}) = 0, \tag{29}$$

$$\mu \geq 0. \tag{30}$$

Here, the left-hand side of (28) is given by

$$\begin{aligned}
&\frac{\partial L}{\partial h_{kn}}(h^*_{kn}, \mu) \\
&= \left(h^*_{kn} \mathbf{w}^{*\mathrm{T}}_k + \sum_{\tilde{k} \neq k} h^*_{\tilde{k}n} \mathbf{w}^{*\mathrm{T}}_{\tilde{k}} - \mathbf{x}^{\mathrm{T}}_n \right) \mathbf{w}^*_k + \alpha_{\mathrm{sp}} \\
&+ \alpha_{\mathrm{sm}} \sum_{t=1}^{T} L_{tn} \left(L_{tn} h^*_{kn} + \sum_{\tilde{n} \neq n} L_{t\tilde{n}} h^*_{k\tilde{n}} \right) - \mu \\
&= \left(\sum_{p=1}^{K} h^*_{pn} \mathbf{w}^{*\mathrm{T}}_p - \mathbf{x}^{\mathrm{T}}_n \right) \mathbf{w}^*_k + \alpha_{\mathrm{sp}} \\
&+ \alpha_{\mathrm{sm}} \sum_{t=1}^{T} L_{tn} \sum_{\tilde{n}=1}^{N} L_{t\tilde{n}} h^*_{k\tilde{n}} - \mu.
\end{aligned}$$

So the conditions (28)–(30) can be rewritten as

$$\begin{aligned}
&\left(\sum_{p=1}^{K} h^*_{pn} \mathbf{w}^{*\mathrm{T}}_p - \mathbf{x}^{\mathrm{T}}_n \right) \mathbf{w}^*_k + \alpha_{\mathrm{sp}} \\
&\qquad + \alpha_{\mathrm{sm}} \sum_{t=1}^{T} L_{tn} \sum_{\tilde{n}=1}^{N} L_{t\tilde{n}} h^*_{k\tilde{n}} \geq 0,
\end{aligned}$$

$$\begin{aligned}
&\left(\left(\sum_{p=1}^{K} h^*_{pn} \mathbf{w}^{*\mathrm{T}}_p - \mathbf{x}^{\mathrm{T}}_n \right) \mathbf{w}^*_k + \alpha_{\mathrm{sp}} \right. \\
&\qquad \left. + \alpha_{\mathrm{sm}} \sum_{t=1}^{T} L_{tn} \sum_{\tilde{n}=1}^{N} L_{t\tilde{n}} h^*_{k\tilde{n}} \right) (\epsilon - h^*_{kn}) = 0.
\end{aligned}$$

Note that the problem (22) has a unique stationary point and it is also a unique optimal solution because $\psi_{kn}(\mathbf{h}_{kn})$ is strictly convex. Therefore, $h_{kn} \geq \epsilon$ is a unique optimal solution of the problem (22) for $n = 1, 2, \ldots, N$ if and only if

$$\begin{aligned}
&(\mathbf{H}^* \mathbf{W}^{*\mathrm{T}} - \mathbf{X}^{\mathrm{T}}) \mathbf{w}^*_k + \alpha_{\mathrm{sp}} \mathbf{1}_{N \times 1} \\
&\qquad + \alpha_{\mathrm{sm}} \mathbf{L}^{\mathrm{T}} \mathbf{L} \mathbf{h}^*_k \geq \mathbf{0}_{N \times 1}, \quad k = 1, 2, \ldots, K,
\end{aligned}$$

$$\begin{aligned}
&\left((\mathbf{H}^* \mathbf{W}^{*\mathrm{T}} - \mathbf{X}^{\mathrm{T}}) \mathbf{w}^*_k + \alpha_{\mathrm{sp}} \mathbf{1}_{N \times 1} + \alpha_{\mathrm{sm}} \mathbf{L}^{\mathrm{T}} \mathbf{L} \mathbf{h}^*_k \right) \\
&\qquad \odot (\epsilon \mathbf{1}_{N \times 1} - \mathbf{h}^*_k) = \mathbf{0}_{N \times 1}, \quad k = 1, 2, \ldots, K.
\end{aligned}$$

These equations can be rewritten as

$$\begin{aligned}
&(\mathbf{H}^* \mathbf{W}^{*\mathrm{T}} - \mathbf{X}^{\mathrm{T}}) \mathbf{W}^* + \alpha_{\mathrm{sp}} \mathbf{1}_{N \times K} \\
&\qquad + \alpha_{\mathrm{sm}} \mathbf{L}^{\mathrm{T}} \mathbf{L} \mathbf{H}^* \geq \mathbf{0}_{N \times K}, \quad (31)
\end{aligned}$$

$$\begin{aligned}
&\left((\mathbf{H}^* \mathbf{W}^{*\mathrm{T}} - \mathbf{X}^{\mathrm{T}}) \mathbf{W}^* + \alpha_{\mathrm{sp}} \mathbf{1}_{N \times K} + \alpha_{\mathrm{sm}} \mathbf{L}^{\mathrm{T}} \mathbf{L} \mathbf{H}^* \right) \\
&\qquad \odot (\epsilon \mathbf{1}_{N \times K} - \mathbf{H}^*) = \mathbf{0}_{N \times K}. \quad (32)
\end{aligned}$$

Note that the set of conditions (26)–(27) and (31)–(32) is equivalent to set of conditions (17)–(20) with $\mathbf{W} = \mathbf{W}^*$ and $\mathbf{H} = \mathbf{H}^*$, which is the necessary and sufficient condition for $(\mathbf{W}^*, \mathbf{H}^*) \in \mathcal{F}_\epsilon$ to be a stationary point of (12). □

Lemma 3: If $(\mathbf{W}, \mathbf{H}) \in \mathcal{S}_\epsilon$ then $A(\mathbf{W}, \mathbf{H}) = (\mathbf{W}, \mathbf{H})$. If $(\mathbf{W}, \mathbf{H}) \in \mathcal{F}_\epsilon \setminus \mathcal{S}_\epsilon$ then $f(A(\mathbf{W}, \mathbf{H})) < f(\mathbf{W}, \mathbf{H})$.

Proof : Suppose first that $(\mathbf{W}, \mathbf{H}) \in \mathcal{S}_\epsilon$. Then it follows from Lemma 2 that $\mathbf{w}_k$ and h_{kn} are unique optimal solutions of (21) and (22), respectively, for $k = 1, 2, \ldots, K$ and $n = 1, 2, \ldots, N$. In other words,

$$\mathbf{w}_k = \left[\frac{\mathbf{R}_k \mathbf{h}_k}{\mathbf{h}^{\mathrm{T}}_k \mathbf{h}_k} \right]_{\epsilon+}$$

holds for $k = 1, 2, \ldots, K$ and

$$h_{kn} = \left[\frac{\mathbf{r}^{\mathrm{T}}_{kn} \mathbf{w}_k - \alpha_{\mathrm{sp}} - \alpha_{\mathrm{sm}} \sum_{t=1}^{T} L_{tn} \sum_{\tilde{n}=1, \tilde{n} \neq n}^{N} L_{t\tilde{n}} h_{k\tilde{n}}}{\mathbf{w}^{\mathrm{T}}_k \mathbf{w}_k + \alpha_{\mathrm{sm}} \sum_{t=1}^{T} L^2_{tn}} \right]_{\epsilon+}$$

holds for $k = 1, 2, \ldots, K$ and $n = 1, 2, \ldots, N$. We thus have $A(\mathbf{W}, \mathbf{H}) = (\mathbf{W}, \mathbf{H})$. Suppose next that $(\mathbf{W}, \mathbf{H}) \in \mathcal{F}_\epsilon \setminus \mathcal{S}_\epsilon$. Let $\mathcal{K}_1 \subseteq \{1, 2, \ldots, K\}$ be the set of k such that $\mathbf{w}_k$ is not an optimal solution of (21), and $\mathcal{K}_2 \subseteq \{(1, 1), (1, 2), \ldots, (1, N), (2, 1), (2, 2), \ldots, (K, N)\}$ be the set of pairs (k, n) such that h_{kn} is not an optimal solution of (22). Then it is clear that $\mathcal{K}_1 \neq \emptyset$ or $\mathcal{K}_2 \neq \emptyset$ holds. If $\mathbf{w}_{\hat{k}}$ is the block first updated among all blocks in $\{\mathbf{w}_k \mid k \in \mathcal{K}_1\} \cup \{h_{kn} \mid (k, n) \in \mathcal{K}_2\}$, the objective value does not change before the update of $\mathbf{w}_{\hat{k}}$, strictly decreases through the update of $\mathbf{w}_{\hat{k}}$, and does not increase after the update of $\mathbf{w}_{\hat{k}}$. Similarly, if $h_{\hat{k}\hat{n}}$ is the block first updated among all blocks in $\{\mathbf{w}_k \mid k \in \mathcal{K}_1\} \cup \{h_{kn} \mid (k, n) \in \mathcal{K}_2\}$, the value of the objective function does not change before the update of $h_{\hat{k}\hat{n}}$, strictly decreases through the update of $h_{\hat{k}\hat{n}}$, and does not increase after the update of $h_{\hat{k}\hat{n}}$. From these observations, we conclude that $f(A(\mathbf{W}, \mathbf{H}))$ is strictly less than $f(\mathbf{W}, \mathbf{H})$. □

Finally, we prove the first statement.

Lemma 4: For any $\mathbf{W}^{(0)} \geq \epsilon \mathbf{1}_{M \times K}, \mathbf{H}^{(0)} \geq \epsilon \mathbf{1}_{N \times K}$, the sequence $\{(\mathbf{W}^{(l)}, \mathbf{H}^{(l)})\}_{l=0}^{\infty}$ generated by (10) and (16) is contained in a closed bounded set

$$\{(\mathbf{W},\mathbf{H}) \mid f(\mathbf{W},\mathbf{H}) \le f(\mathbf{W}^{(0)},\mathbf{H}^{(0)}), \quad \mathbf{W} \ge \epsilon\mathbf{1}_{M\times K}, \mathbf{H} \ge \epsilon\mathbf{1}_{N\times K}\}. \quad (33)$$

Proof : It is clear from Lemma 3 that $\{(\mathbf{W}^{(l)},\mathbf{H}^{(l)})\}_{l=0}^{\infty}$ is contained in (33). So it suffices for us to prove that (33) is bounded. Let $C = f(\mathbf{W}^{(0)},\mathbf{H}^{(0)})$. If $f(\mathbf{W},\mathbf{H}) \le C$ then we have the following inequality:

$$\frac{1}{2}\left\|\mathbf{X} - \mathbf{W}\mathbf{H}^{\mathrm{T}}\right\|_{\mathrm{F}}^2 \le C$$

which implies that

$$X_{mn} - \sqrt{2C} \le \sum_{k=1}^{K} W_{mk}H_{nk} \le X_{mn} + \sqrt{2C} \quad (34)$$

for all m and n. Moreover, if $(\mathbf{W},\mathbf{H}) \in \mathcal{F}_\epsilon$, it follows from (34) that

$$W_{mk} \le \frac{X_{mn} + \sqrt{2C}}{H_{nk}} \le \frac{X_{mn} + \sqrt{2C}}{\epsilon}, \quad (35)$$

$$H_{nk} \le \frac{X_{mn} + \sqrt{2C}}{W_{mk}} \le \frac{X_{mn} + \sqrt{2C}}{\epsilon} \quad (36)$$

for all m, n and k. This means that (33) is bounded. □

In the above analysis, we took the same approach as in the previous work [19], but the boundedness of solutions was proved in a different way. In [19], a compact subset of $\mathcal{F}_\epsilon$, which depends only on $\mathbf{X}$ and ϵ, was derived directly from the update rule, while in this paper we proved the boundedness of solutions by making use of the level set of the objective function which is determined not only from $\mathbf{X}$ and ϵ but also the initial solution $(\mathbf{W}^{(0)},\mathbf{H}^{(0)})$.

3.3 Finite Termination of GSHALS Algorithm

Note that Theorem 1 does not guarantee the convergence of the whole sequence but only the existence of a subsequence converging to a stationary point. This is, however, sufficient for us because, by introducing an appropriate stopping condition, we can obtain an algorithm that always stops within a finite number of iterations after reaching an approximate stationary point [12].

The necessary and sufficient condition described by (17)–(20) for $(\mathbf{W},\mathbf{H}) \in \mathcal{F}_\epsilon$ to be a stationary point can be relaxed by using any positive constants δ_1, δ_2 as follows:

$$\nabla_{\mathbf{W}} f(\mathbf{W},\mathbf{H}) \ge -\delta_1\mathbf{1}_{M\times K}, \quad (37)$$

$$\nabla_{\mathbf{H}} f(\mathbf{W},\mathbf{H}) \ge -\delta_1\mathbf{1}_{N\times K}, \quad (38)$$

$$W_{mk} - \epsilon \le \delta_2 \ \text{ if } \ (\nabla_{\mathbf{W}} f(\mathbf{W},\mathbf{H}))_{mk} > \delta_1, \quad (39)$$

$$H_{nk} - \epsilon \le \delta_2 \ \text{ if } \ (\nabla_{\mathbf{H}} f(\mathbf{W},\mathbf{H}))_{nk} > \delta_1. \quad (40)$$

Let the set of $(\mathbf{W},\mathbf{H}) \in \mathcal{F}_\epsilon$ which satisfies (37)–(40) be denoted by $\bar{\mathcal{S}}_\epsilon$. Also, let $\mathbb{R}$, $\mathbb{R}_+$ and $\mathbb{R}_{++}$ denote the set of real numbers, the set of nonnegative real numbers, and the set of positive real numbers, respectively. Furthermore, let $\mathbb{N}$ denote the set of natural numbers. Then the algorithm with a stopping condition is stated as follows.

Algorithm 1 GSHALS

Input: $\mathbf{X} \in \mathbb{R}_+^{M\times N}$, $\mathbf{L} \in \mathbb{R}^{T\times N}$, $K \in \mathbb{N}$, α_{sp}, α_{sm}, ϵ, δ_1, $\delta_2 \in \mathbb{R}_{++}$
Output: $(\mathbf{W},\mathbf{H}) \in \bar{\mathcal{S}}_\epsilon$
1: Choose $(\mathbf{W},\mathbf{H}) \in \mathcal{F}_\epsilon$.
2: Update $2K$ columns of $\mathbf{W}$ and $\mathbf{H}$ one by one in the fixed order by using (10) and (16).
3: If $(\mathbf{W},\mathbf{H})$ satisfies (37)–(40) then return $(\mathbf{W},\mathbf{H})$ and stop. Otherwise go to Step 2.

For Algorithm 1, we have the following theorem. The proof is omitted because it is similar to [12].

Theorem 2: For any positive constants $\epsilon, \delta_1, \delta_2$ and any initial solution $(\mathbf{W}^{(0)},\mathbf{H}^{(0)}) \in \mathcal{F}_\epsilon$, Algorithm 1 with the stopping condition given by (37)–(40) always stops within a finite number of iterations.

3.4 Fast GSHALS Algorithm

In the HALS algorithm, the computational cost per round is reduced by updating matrices $\mathbf{E} = \mathbf{X} - \mathbf{W}\mathbf{H}^{\mathrm{T}}$ and $\mathbf{R}_k$ $(k = 1, 2, \ldots, K)$ in a proper way [23]. This technique can be directly applied to the GSHALS algorithm. An important point is that $\mathbf{R}_k$ can be efficiently computed from the current values of $\mathbf{E}$, $\mathbf{w}_k$ and $\mathbf{h}_k$ as

$$\mathbf{R}_k = \mathbf{E} + \mathbf{w}_k\mathbf{h}_k^{\mathrm{T}}$$

and $\mathbf{E}$ can be efficiently computed from the current values of $\mathbf{R}_k$, $\mathbf{w}_k$ and $\mathbf{h}_k$ as

$$\mathbf{E} = \mathbf{R}_k - \mathbf{w}_k\mathbf{h}_k^{\mathrm{T}}.$$

Another important point is that $\nabla_{\mathbf{W}} f(\mathbf{W},\mathbf{H})$ and $\nabla_{\mathbf{H}} f(\mathbf{W},\mathbf{H})$ can be efficiently computed from the current values of $\mathbf{W}$, $\mathbf{H}$ and $\mathbf{E}$ as $\nabla_{\mathbf{W}} f(\mathbf{W},\mathbf{H}) = -\mathbf{E}\mathbf{H}$ and $\nabla_{\mathbf{H}} f(\mathbf{W},\mathbf{H}) = -\mathbf{E}^{\mathrm{T}}\mathbf{W}$. Therefore, we can determine whether the relaxed optimality condition given by (37)–(40) is satisfied or not, by checking the following conditions:

$$\mathbf{E}\mathbf{H} \le \delta_1\mathbf{1}_{M\times K}, \quad (41)$$

$$\mathbf{E}^{\mathrm{T}}\mathbf{W} \le \delta_1\mathbf{1}_{N\times K}, \quad (42)$$

$$W_{mk} - \epsilon \le \delta_2 \ \text{ if } \ (\mathbf{E}\mathbf{H})_{mk} < -\delta_1, \quad (43)$$

$$H_{nk} - \epsilon \le \delta_2 \ \text{ if } \ (\mathbf{E}^{\mathrm{T}}\mathbf{W})_{nk} < -\delta_1. \quad (44)$$

Making use of these ideas, we obtain two fast GSHALS algorithms depending on the update order as follows.

Algorithm 2 Fast GSHALS with update order (2)

Input: $\mathbf{X} \in \mathbb{R}_+^{M\times N}$, $\mathbf{L} \in \mathbb{R}^{T\times N}$, $K \in \mathbb{N}$, $\alpha_{\mathrm{sp}}, \alpha_{\mathrm{sm}}, \epsilon, \delta_1, \delta_2 \in \mathbb{R}_{++}$
Output: $(\mathbf{W}, \mathbf{H}) \in \bar{\mathcal{S}}_\epsilon$

1: Set $\mathbf{M} \leftarrow \mathbf{L}^{\mathrm{T}}\mathbf{L}$.
2: Choose $(\mathbf{W}, \mathbf{H}) \in \mathcal{F}_\epsilon$ and set $\mathbf{E} \leftarrow \mathbf{X} - \mathbf{W}\mathbf{H}^{\mathrm{T}}$.
3: Set $k \leftarrow 1$.
4: Set $\mathbf{R}_k \leftarrow \mathbf{E} + \mathbf{w}_k\mathbf{h}_k^{\mathrm{T}}$.
5: Set $\mathbf{w}_k \leftarrow \left[\mathbf{R}_k\mathbf{h}_k / \mathbf{h}_k^{\mathrm{T}}\mathbf{h}_k\right]_{\epsilon+}$.
6: Set $n \leftarrow 1$.
7: Set $h_{kn} \leftarrow \left[\dfrac{\mathbf{r}_{kn}^{\mathrm{T}}\mathbf{w}_k - \alpha_{\mathrm{sp}} - \alpha_{\mathrm{sm}}\sum_{\tilde{n}=1,\tilde{n}\neq n}^{N} M_{n\tilde{n}}h_{k\tilde{n}}}{\mathbf{w}_k^{\mathrm{T}}\mathbf{w}_k + \alpha_{\mathrm{sp}}M_{nn}}\right]_{\epsilon+}$.
8: If $n = N$ then go to Step 9. Otherwise add 1 to n and go to Step 7.
9: Set $\mathbf{E} \leftarrow \mathbf{R}_k - \mathbf{w}_k\mathbf{h}_k^{\mathrm{T}}$.
10: If $k = K$ then go to Step 11. Otherwise add 1 to k and go to Step 4.
11: If $(\mathbf{W}, \mathbf{H})$ satisfies (41)–(44) then return $(\mathbf{W}, \mathbf{H})$ and stop. Otherwise go to Step 3.

Algorithm 3 Fast GSHALS with update order (3)

Input: $\mathbf{X} \in \mathbb{R}_+^{M\times N}$, $\mathbf{L} \in \mathbb{R}^{T\times N}$, $K \in \mathbb{N}$, $\alpha_{\mathrm{sp}}, \alpha_{\mathrm{sm}}, \epsilon, \delta_1, \delta_2 \in \mathbb{R}_{++}$
Output: $(\mathbf{W}, \mathbf{H}) \in \bar{\mathcal{S}}_\epsilon$

1: Set $\mathbf{M} \leftarrow \mathbf{L}^{\mathrm{T}}\mathbf{L}$.
2: Choose $(\mathbf{W}, \mathbf{H}) \in \mathcal{F}_\epsilon$ and set $\mathbf{E} \leftarrow \mathbf{X} - \mathbf{W}\mathbf{H}^{\mathrm{T}}$.
3: Set $k \leftarrow 1$.
4: Set $\mathbf{R}_k \leftarrow \mathbf{E} + \mathbf{w}_k\mathbf{h}_k^{\mathrm{T}}$.
5: Set $\mathbf{w}_k \leftarrow \left[\mathbf{R}_k\mathbf{h}_k / \mathbf{h}_k^{\mathrm{T}}\mathbf{h}_k\right]_{\epsilon+}$.
6: Set $\mathbf{E} \leftarrow \mathbf{R}_k - \mathbf{w}_k\mathbf{h}_k^{\mathrm{T}}$.
7: If $k = K$ then go to Step 8. Otherwise add 1 to k and go to Step 4.
8: Set $k \leftarrow 1$.
9: Set $\mathbf{R}_k \leftarrow \mathbf{E} + \mathbf{w}_k\mathbf{h}_k^{\mathrm{T}}$.
10: Set $n \leftarrow 1$.
11: Set $h_{kn} \leftarrow \left[\dfrac{\mathbf{r}_{kn}^{\mathrm{T}}\mathbf{w}_k - \alpha_{\mathrm{sp}} - \alpha_{\mathrm{sm}}\sum_{\tilde{n}=1,\tilde{n}\neq n}^{N} M_{n\tilde{n}}h_{k\tilde{n}}}{\mathbf{w}_k^{\mathrm{T}}\mathbf{w}_k + \alpha_{\mathrm{sp}}M_{nn}}\right]_{\epsilon+}$.
12: If $n = N$ then go to Step 13. Otherwise add 1 to n and go to Step 11.
13: Set $\mathbf{E} \leftarrow \mathbf{R}_k - \mathbf{w}_k\mathbf{h}_k^{\mathrm{T}}$.
14: If $k = K$ then go to Step 15. Otherwise add 1 to k and go to Step 9.
15: If $(\mathbf{W}, \mathbf{H})$ satisfies (41)–(44) then return $(\mathbf{W}, \mathbf{H})$ and stop. Otherwise go to Step 3.

It is easy to see that the computational complexity per round of Algorithms 2 and 3 is $O(MNK + N^2K)$. This is equal to the computational complexity per round of the Fast HALS algorithm because, using the eigenvalue decomposition $\mathbf{L}^{\mathrm{T}}\mathbf{L} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{\mathrm{T}}$ where $\mathbf{Q}$ is an orthogonal matrix and $\mathbf{\Lambda}$ is a nonnegative diagonal matrix, we can rewrite (11) as

$$\mathbf{h}_k \leftarrow [\mathbf{Q}(\mathbf{w}_k^{\mathrm{T}}\mathbf{w}_k\mathbf{I}_{N\times N} + \alpha_{\mathrm{sm}}\mathbf{\Lambda})^{-1}\mathbf{Q}^{\mathrm{T}} \times (\mathbf{R}_k^{\mathrm{T}}\mathbf{w}_k - \alpha_{\mathrm{sp}}\mathbf{1}_{N\times 1})]_{\epsilon+}$$

which takes $O(MN + N^2)$ time. An important difference between Algorithms 2 and 3 is that the former updates $\mathbf{R}_k$ and $\mathbf{E}$ K times in each round, while the latter needs to do it $2K$ times. Therefore, the computation time of Algorithm 2 is shorter than that of Algorithm 3 for the same number of rounds. However, we cannot conclude from this observation that Algorithm 2 is faster than Algorithm 3, because the number of rounds required to reach an approximate stationary point depends on the update order.

4. Numerical Experiments

In order to examine the effectiveness of the GSHALS algorithm, we compare the performance of the MUR, the HALS algorithm and the GSHALS algorithm by using synthetic and real datasets. The MUR can be easily obtained by using the unified method proposed by Yang and Oja [8], and the resulting update rule is described by

$$\mathbf{W} \leftarrow \left[\mathbf{W} \odot \mathbf{X}\mathbf{H} \oslash \mathbf{W}\mathbf{H}^{\mathrm{T}}\mathbf{H}\right]_{\epsilon+}, \tag{45}$$

$$\mathbf{H} \leftarrow \left[\mathbf{H} \odot \mathbf{X}^{\mathrm{T}}\mathbf{W} \oslash \left(\mathbf{H}\mathbf{W}^{\mathrm{T}}\mathbf{W} + \alpha_{\mathrm{sp}}\mathbf{1}_{N\times K} + \alpha_{\mathrm{sm}}\mathbf{L}^{\mathrm{T}}\mathbf{L}\mathbf{H}\right)\right]_{\epsilon+}, \tag{46}$$

where $\oslash$ represents the componentwise division. The computational complexity per round of the MUR is $O(MNK + N^2K)$ like the HALS and GSHALS algorithms.

In all experiments, we set $\mathbf{L} = \mathbf{L}_2$ and use (37)–(40) as a stopping condition. The HALS and GSHALS algorithms are implemented by using the technique described in Sect. 3.4. In the GSHALS algorithm, $\mathbf{h}_k$ is updated in the order $h_{k1} \to h_{k2} \to \cdots \to h_{kN}$. All methods are implemented in C language with BLAS and LAPACK libraries, compiled with gcc 5.3.0, and executed on a PC with Intel Core i7-6700, 16 GB memory and Windows 10.

4.1 Experiment Using Synthetic Datasets

We first compare the performance of the five methods: MUR, HALS with (2), HALS with (3), GSHALS with (2), and GSHALS with (3) by using synthetic datasets. In this experiment, we set $M = 100$, $N = 50$, $K = 10$, $\alpha_{\mathrm{sm}} = 0.1, \alpha_{\mathrm{sp}} = 0.1, \epsilon = 0.001, \delta_1 \in \{0.1, 0.01, 0.001\}$ and $\delta_2 \in \{0.01, 0.001, 0.0001\}$. For each of the nine pairs of δ_1 and δ_2 values, we applied the five methods to 10 different triples $(\mathbf{X}, \mathbf{W}^{(0)}, \mathbf{H}^{(0)})$ which were generated in such a way that each entry of $\mathbf{X}$, $\mathbf{W}^{(0)}$ and $\mathbf{H}^{(0)}$ was drawn from an independent uniform distribution on the interval $[0, 1]$ and then all entries of $\mathbf{W}^{(0)}$ and $\mathbf{H}^{(0)}$ less than ϵ were replaced with ϵ so that $(\mathbf{W}^{(0)}, \mathbf{H}^{(0)}) \in \mathcal{F}_\epsilon$. The performance of the five methods are compared in terms of the number of rounds and the computation time. The maximum number of rounds were set to 60000, that is, each algorithm stops when the number of rounds reaches 60000 even if the stopping condition is not satisfied.

Results of the experiment are shown in Tables 1–3. We first see from these tables that the computational cost is not so sensitive to the value of δ_2 for all algorithms. From Tables 1 and 2 where the results for $\delta_1 = 0.1$ and $\delta_1 = 0.01$ are given respectively, we see that i) the HALS and the GSHALS algorithms are much faster than the MUR, ii) the update order (2) is faster than (3) for the same algorithm, and iii) HALS with (2) is faster than GSHALS with (3). However, some of these properties do not hold for a smaller value of δ_1. In fact, we see from Table 3, where the results

for $\delta_1 = 0.001$ are given, that the HALS algorithm did not satisfy the stopping condition before the number of rounds reached 60000 in all cases. This is because the objective value eventually stops decreasing as explained in Sect. 2.2. In contrast, the MUR and the GSHALS algorithm stopped within 60000 rounds in all cases. The GSHALS algorithm is much faster than the MUR. In particular, the GSHALS with (2) is the fastest.

It should be noted that the objective value when the stopping condition is satisfied differs depending on the algorithm, even if the same parameter values and the same initial condition are used. It often occurs that the final objective value reached by an algorithm is less than that by a faster algorithm. In fact, we see from Table 4 that the final objective value obtained by the GSHALS algorithm with the update order (2), which is the fastest among five methods as shown in Table 1, is not the smallest for all settings. This means that different algorithms may reach different approximate stationary points.

Table 1 Results for synthetic data ($\delta_1 = 0.1$).

δ_2	Method	Number of Rounds			Computation Time [s]		
		Ave.	Min.	Max.	Ave.	Min.	Max.
0.01	MUR	6173.3	2969	10985	2.191	1.032	3.891
	HALS (2)	**375.9**	210	**539**	**0.113**	**0.063**	**0.157**
	HALS (3)	1049.9	919	1178	0.433	0.375	0.469
	GSHALS (2)	379.5	**204**	739	0.121	**0.063**	0.235
	GSHALS (3)	1237.8	1126	1728	0.512	0.484	0.672
0.001	MUR	7618.8	4819	11179	2.704	1.719	3.985
	HALS (2)	495.8	319	752	0.155	0.093	0.235
	HALS (3)	1146.8	1030	1268	0.464	0.422	0.515
	GSHALS (2)	**354.7**	**220**	**585**	**0.107**	**0.063**	**0.172**
	GSHALS (3)	1165.5	1060	1214	0.478	0.438	0.516
0.0001	MUR	8203.5	4888	15777	2.977	1.708	5.678
	HALS (2)	433.8	267	772	0.131	0.078	0.235
	HALS (3)	1155.5	925	1521	0.469	0.375	0.609
	GSHALS (2)	**306.5**	**125**	**515**	**0.096**	**0.037**	**0.160**
	GSHALS (3)	1212.4	1096	1577	0.507	0.451	0.670

Table 2 Results for synthetic data ($\delta_1 = 0.01$).

δ_2	Method	Number of Rounds			Computation Time [s]		
		Ave.	Min.	Max.	Ave.	Min.	Max.
0.01	MUR	15793.2	10982	24186	6.018	4.104	9.168
	HALS (2)	1229.1	694	2296	0.374	0.204	0.687
	HALS (3)	2664.8	2271	3702	1.083	0.922	1.548
	GSHALS (2)	**962.8**	**342**	**1467**	**0.313**	**0.103**	**0.473**
	GSHALS (3)	2788.2	2564	2927	1.211	1.118	1.307
0.001	MUR	16336.0	10403	33477	6.141	3.642	12.894
	HALS (2)	**1051.0**	571	**2233**	**0.314**	0.172	**0.672**
	HALS (3)	2564.4	2295	2870	1.030	0.922	1.156
	GSHALS (2)	1163.6	**487**	2512	0.377	**0.152**	0.779
	GSHALS (3)	2790.7	2632	2919	1.207	1.110	1.312
0.0001	MUR	20129.8	10458	38751	7.268	3.657	14.140
	HALS (2)	**1092.5**	**571**	1601	**0.335**	**0.172**	0.485
	HALS (3)	2631.5	2470	3015	1.077	1.000	1.235
	GSHALS (2)	1150.8	641	**1583**	0.355	0.192	**0.481**
	GSHALS (3)	2945.8	2683	3848	1.229	1.107	1.584

Table 3 Results for synthetic data ($\delta_1 = 0.001$).

δ_2	Method	Number of Rounds			Computation Time [s]		
		Ave.	Min.	Max.	Ave.	Min.	Max.
0.01	MUR	27251.0	14184	46474	9.474	4.647	18.340
	HALS (2)	60000.0	60000	60000	18.522	18.174	18.799
	HALS (3)	60000.0	60000	60000	24.867	24.424	25.260
	GSHALS (2)	**1633.0**	**852**	**4352**	**0.526**	**0.249**	**1.377**
	GSHALS (3)	4554.1	4281	4795	1.930	1.754	2.162
0.001	MUR	24850.6	11136	53797	8.676	3.672	19.908
	HALS (2)	60000.0	60000	60000	18.496	18.189	18.690
	HALS (3)	60000.0	60000	60000	24.705	24.471	25.003
	GSHALS (2)	**2152.1**	**1232**	**3723**	**0.655**	**0.371**	**1.141**
	GSHALS (3)	4577.6	4426	4798	1.894	1.838	1.968
0.0001	MUR	27367.8	14684	41168	9.453	5.047	13.377
	HALS (2)	60000.0	60000	60000	18.636	18.361	18.955
	HALS (3)	60000.0	60000	60000	24.939	24.534	25.519
	GSHALS (2)	**1195.5**	**985**	**1995**	**0.370**	**0.296**	**0.609**
	GSHALS (3)	4554.4	4444	4921	1.930	1.812	2.172

4.2 Experiment Using Real Datasets

We next compare the performance of the three methods: MUR, GSHALS with (2), and GSHALS with (3) by using three kinds of real datasets. We do not consider the HALS algorithm in this experiment because we have observed in the previous experiment that the HALS algorithm always shows a similar or worse performance than the GSHALS algorithm and, more importantly, the HALS algorithm does not always reach an approximate stationary point.

The first dataset is the ORL Database of Faces[†] which is a facial image dataset offered by AT&T Laboratories Cambridge. This dataset contains 400 grayscale facial images and the size of each image is 92×112. Reducing the size of all images to 46×56, transforming them into column vectors, we obtain a 2576×400 nonnegative matrix $\mathbf{X}$. We set $\alpha_{\mathrm{sm}} = 0.1$, $\alpha_{\mathrm{sp}} = 0.1$, $\epsilon = 1.0$, $\delta_1 = 10.0$, $\delta_2 = 1.0$ and $K = 5$, and run the three algorithms for 10 different initial solutions $(\mathbf{W}^{(0)}, \mathbf{H}^{(0)})$ which are generated in such a way that each entry is drawn from an independent uniform distribution on the interval $[0, 10]$ and then all entries less than ϵ are replaced with ϵ so that $(\mathbf{W}^{(0)}, \mathbf{H}^{(0)}) \in \mathcal{F}_\epsilon$.

The second dataset is the Wisconsin Diagnostic Breast Cancer (WDBC) dataset which can be obtained from the UCI Machine Learning Repository[††]. This dataset contains 569 instances each of which consists of 32 features. Excluding the first two features (the first is ID number and the second is diagnosis), normalizing each of the remaining 30 features so that the values belong to the interval $[0, 1]$, we obtain a 30×569 nonnegative matrix $\mathbf{X}$. We set $\alpha_{\mathrm{sm}} = 0.1$, $\alpha_{\mathrm{sp}} = 0.1$, $\epsilon = 0.001$, $\delta_1 = 0.005$, $\delta_2 = 0.001$ and $K = 2$, and run the three algorithms for 10 different initial solutions $(\mathbf{W}^{(0)}, \mathbf{H}^{(0)})$ which are generated in the same way as in the case of the

[†]http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[††]http://archive.ics.uci.edu/ml/

Table 4 Final objective values for synthetic data ($\delta_1 = 0.1$, $\delta_2 = 0.0001$).

Method	Final Objective Value									
	Setting 1	Setting 2	Setting 3	Setting 4	Setting 5	Setting 6	Setting 7	Setting 8	Setting 9	Setting 10
MUR	127.268	124.773	125.814	130.235	127.546	126.461	**127.147**	126.615	**124.520**	130.190
HALS (2)	127.285	124.595	126.137	**129.923**	**127.276**	126.836	127.428	126.627	125.085	130.319
HALS (3)	127.107	**124.482**	**125.736**	130.007	127.714	127.004	127.449	126.471	124.609	**130.028**
GSHALS (2)	127.607	124.732	126.211	130.167	127.889	126.309	128.165	126.610	125.574	130.470
GSHALS (3)	**127.080**	124.599	125.854	130.128	127.625	**126.216**	127.182	**126.397**	124.541	130.049

Table 5 Results for ORL dataset.

Method	Number of Rounds			Computation Time [s]		
	Ave.	Min.	Max.	Ave.	Min.	Max.
MUR	7197.3	6366	8909	466.319	420.269	560.454
GSHALS (2)	**1509.0**	**1121**	**1587**	**64.733**	**46.488**	**75.769**
GSHALS (3)	3662.3	3264	3834	222.477	207.611	237.239

Table 6 Results for WDBC dataset.

Method	Number of Rounds			Computation Time [s]		
	Ave.	Min.	Max.	Ave.	Min.	Max.
MUR	27032.4	24760	27801	20.924	19.033	23.362
GSHALS (2)	14780.8	10457	17660	13.608	10.376	**15.752**
GSHALS (3)	**11109.1**	**8289**	**15332**	**11.123**	**8.500**	15.987

Table 7 Results for CLUTO (tr23) dataset.

Method	Number of Rounds			Computation Time [s]		
	Ave.	Min.	Max.	Ave.	Min.	Max.
MUR	3928.9	2063	6152	306.357	162.234	480.680
GSHALS (2)	830.0	**742**	887	**40.009**	**35.865**	**43.442**
GSHALS (3)	**784.5**	764	**804**	56.870	54.368	61.082

ORL dataset except that the interval [0, 1] is used instead of [0, 10].

The third dataset is "tr23" in the CLUTO datasets. The CLUTO datasets have been used in evaluating the performance of document clustering algorithms and can be obtained from the CLUTO web site[†]. The dataset "tr23" contains 204 instances corresponding to 204 documents from six categories, and each instance contains 5832 features. Hence a 5832×204 nonnegative matrix $\mathbf{X}$ can be obtained from this dataset. We set $\alpha_{\mathrm{sm}} = 0.1$, $\alpha_{\mathrm{sp}} = 0.1$, $\epsilon = 0.1$, $\delta_1 = 1.0$, $\delta_2 = 0.1$ and $K = 6$, and run the three algorithms for 10 different initial solutions $(\mathbf{W}^{(0)}, \mathbf{H}^{(0)})$ which are generated in the same way as in the case of the ORL dataset.

Results of the experiment are shown in Tables 5–7. It is seen from these tables that the GSHALS algorithm is faster than the MUR for all datasets, as in the case of synthetic data. As for the update order of the GSHALS algorithm, which is faster depends on the dataset. The computation time of the GSHALS algorithm with (2) is shorter than that with (3) for the ORL and the CLUTO datasets, but the latter is faster for the WDBC dataset.

5. Conclusion

The HALS algorithm for NMF with sparseness and smoothness constraints has been studied in this paper. We have proposed the GSHALS algorithm based on the HALS algorithm, and proved that it has the global convergence property. We have also showed experimentally that the GSHALS algorithm outperforms the HALS algorithm and the MUR in terms of computation time. The GSHALS algorithm can be easily applied to the case where regularization terms for $\mathbf{W}$ added to the objective function, though it is not considered in this paper. A future problem is the global convergence analysis of the GSHALS algorithm when the update order is not fixed.

[†]http://glaros.dtc.umn.edu/gkhome/cluto/cluto/overview/

Acknowledgments

The authors would like to thank anonymous reviewers for their valuable comments to improve the quality of this paper. This work was supported by JSPS KAKENHI Grant Number JP15K00035.

References

[1] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," Environmetrics, vol.5, no.2, pp.111–126, June 1994. DOI: 10.1002/env.3170050203

[2] D.D. Lee and H.S. Seung, "Learning the parts of objects by non-negative matrix factorization," Nature, vol.401, pp.788–792, 1999. DOI: 10.1038/44565

[3] D.D. Lee and H.S. Seung, "Algorithms for non-negative matrix factorization," in Advances in Neural Information Processing Systems, T.K. Leen, T.G. Dietterich, and V. Tresp, Eds., vol.13, pp.556–562, 2001.

[4] P.O. Hoyer, "Non-negative matrix factorization with sparseness constraints," J. Machine Learning Research, vol.5, pp.1457–1469, 2004.

[5] V.P. Pauca, F. Shahnaz, M.W. Berry, and R.J. Plemmons, "Text mining using non-negative matrix factorizations," Proc. 4th SIAM International Conference on Data Mining, pp.452–456, 2004. DOI: 10.1137/1.9781611972740.45

[6] S. Zhang, W. Wang, J. Ford, and F. Makedon, "Learning from incomplete ratings using non-negative matrix factorization," Proc. 6th SIAM International Conference on Data Mining, pp.548–552, 2006. DOI: 10.1137/1.9781611972764.58

[7] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," IEEE Trans. Ind. Informat., vol.10, no.2, pp.1273–1284, 2014. DOI: 10.1109/TII.2014.2308433

[8] Z. Yang and E. Oja, "Unified development of multiplicative algorithms for linear and quadratic nonnegative matrix factorization," IEEE Trans. Neural Netw., vol.22, no.12, pp.1878–1891, Dec. 2011. DOI: 10.1109/TNN.2011.2170094

[9] W.I. Zangwill, Nonlinear Programming: A Unified Approach, Prentice-Hall, 1969.

[10] N. Gillis and F. Glineur, "Nonnegative matrix factorization and the maximum edge biclique problem," arXiv0810.4225, 2008.

[11] A. Cichocki, R. Zdunek, and S. Amari, "Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization," Springer Lecture Notes in Computer Science, vol.4666, pp.169–176, 2007. DOI: 10.1007/978-3-540-74494-8_22

[12] N. Takahashi and R. Hibi, "Global convergence of modified multiplicative updates for nonnegative matrix factorization," Computational Optimization and Applications, vol.57, no.2, pp.417–440, 2014. DOI: 10.1007/s10589-013-9593-0

[13] N. Takahashi, J. Katayama, and J. Takeuchi, "A generalized sufficient condition for global convergence of modified multiplicative updates for NMF," Proc. 2014 International Symposium on Nonlinear Theory and its Applications, pp.44–47, 2014.

[14] N. Takahashi and M. Seki, "Multiplicative update for a class of constrained optimization problems related to NMF and its global convergence," Proc. 2016 European Signal Processing Conference, pp.438–442, 2016. DOI: 10.1109/EUSIPCO.2016.7760286

[15] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," Neural Computation, vol.19, no.10, pp.2756–2779, 2007. DOI: 10.1162/neco.2007.19.10.2756

[16] N.D. Ho, Nonnegative Matrix Factorization Algorithms and Applications, Ph.D thesis, Univ. Catholique de Louvain, 2008.

[17] N. Guan, D. Tao, Z. Luo, and B. Yuan, "NeNMF: An optimal gradient method for nonnegative matrix factorization," IEEE

Trans. Signal Process., vol.60, no.6, pp.2882–2898, 2012. DOI: 10.1109/TSP.2012.2190406
[18] J. Kim, Y. He, and H. Park, "Algorithms for nonnegative matrix and tensor factorization: A unified view based on block coordinate descent framework," J. Global Optimization, vol.58, no.2, pp.285–319, 2014. DOI: 0.1007/s10898-013-0035-4
[19] T. Kimura and N. Takahashi, "Global convergence of a modified HALS algorithm for nonnegative matrix factorization," Proc. 2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, pp.21–24, 2015. DOI: 10.1109/CAMSAP.2015.7383726
[20] P.O. Hoyer, "Non-negative sparse coding," Proc. 2002 IEEE 12th International Workshop on Neural Networks for Signal Processing, pp.557–565, 2002. DOI: 10.1109/NNSP.2002.1030067
[21] M.W. Berry, M. Browne, A.N. Langville, V.P. Pauca, and R.J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," Computational Statistics & Data Analysis, vol.52, no.1, pp.155–173, 2007. DOI: 10.1016/j.csda.2006.11.006
[22] A. Cichocki, S. Amari, R. Zdunek, R. Kompass, G. Hori, and Z. He, "Extended SMART algorithms for non-negative matrix factorization," Lecture Notes in Computer Science, Springer, vol.4029, pp.548–562, 2006. DOI:10.1007/11785231_58
[23] A. Cichocki and A.H. Phan, "Fast local algorithms for large scale nonnegative matrix and tensor factorization," IEICE Trans. Fundamentals, vol.E92-A, no.3, pp.708–721, March 2009.
[24] A. Cichocki, R. Zdunek, A.H. Phan, and S. Amari, Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation, John Wiley & Sons, 2009.
[25] Y. Wang and Y. Zhang, "Nonnegative matrix factorization: A comprehensive review," IEEE Trans. Knowl. Data Eng., vol.25, no.6, pp.1336–1353, June 2013. DOI: 10.1109/TKDE.2012.51
[26] Q. Liao and Q. Zhang, "Efficient rank-one residue approximation method for graph regularized non-negative matrix factorization," Proc. European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, Part II, pp.242–255, 2013. DOI: 10.1007/978-3-642-40991-2_16
[27] D. Cai, X. He, J. Han, and T.S. Huang, "Graph regularized nonnegative matrix factorization for data representation," IEEE Trans. Pattern Analysis and Machine Intelligence, vol.33, no.8, pp.1548–1560, Aug. 2011. DOI: 10.1109/TPAMI.2010.231
[28] O. Axelsson, Iterative Solution Methods, Cambridge University Press, 1996.

Takumi Kimura received the B.E. degree in information technology and M.E. degree in electronic and information systems engineering from Okayama University, Japan in 2015 and 2017, respectively. He is now with ASTEC Co., Ltd., Osaka, Japan. His research interests include nonnegative matrix factorization and its applications.

Norikazu Takahashi received the B.E., M.E. and D.E. degrees from Kyushu University, Japan, in 1991, 1993 and 1996, respectively. He is currently a Professor in the Department of Computer Science, Okayama University, Japan. His research interests include optimization theory, nonlinear systems, multiagent systems, graph theory, and neural networks. He is a member of IEEE and Japanese Neural Network Society.