A Novel Newton-Type Algorithm for Nonnegative Matrix Factorization with Alpha-Divergence

Satoshi Nakatsu(✉) and Norikazu Takahashi

Okayama University, Okayama 700-8530, Japan
nakatsu@momo.cs.okayama-u.ac.jp, takahashi@cs.okayama-u.ac.jp

Abstract. We propose a novel iterative algorithm for nonnegative matrix factorization with the alpha-divergence. The proposed algorithm is based on the coordinate descent and the Newton method. We show that the proposed algorithm has the global convergence property in the sense that the 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. We also show through numerical experiments that the proposed algorithm is much faster than the multiplicative update rule.

Keywords: Nonnegative Matrix Factorization · Alpha-divergence · Newton method · Global convergence

1 Introduction

Nonnegative Matrix Factorization (NMF) [1–3] is a mathematical operation that decomposes a given nonnegative matrix $\boldsymbol{X}$ into two nonnegative matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ such that $\boldsymbol{X} \approx \boldsymbol{W}\boldsymbol{H}$. NMF has found many applications in various fields such as image processing, acoustic signal processing, data analysis and text mining because it can find nonnegative basis for a given set of nonnegative data.

NMF is formulated as a constrained optimization problem in which an error between $\boldsymbol{X}$ and $\boldsymbol{W}\boldsymbol{H}$ has to be minimized under the nonnegativity constraints on $\boldsymbol{W}$ and $\boldsymbol{H}$. Multiplicative update rules [3] are widely used as simple and easy-to-implement methods for solving the NMF optimization problems. This approach can be easily applied to a wide class of error measures [4–6], and the obtained update rules have the global convergence property [7,8]. However, they are often slow. Hence many studies have been done to develop faster algorithms for solving the NMF optimization problems (see, for example, [9,10] and references therein).

In this paper, we focus our attention on NMF with the alpha-divergence [11]. The alpha-divergence includes Pearson divergence, Hellinger divergence, and chi-square divergence as its special cases [12], and has been frequently used for NMF (see [13] and references therein). As a simple and fast method for solving the optimization problem for NMF with the alpha-divergence, we propose a novel

D. Liu et al. (Eds.): ICONIP 2017, Part I, LNCS 10634, pp. 335–344, 2017.
https://doi.org/10.1007/978-3-319-70087-8_36

iterative algorithm based on the coordinate descent and the Newton method. We show that the proposed algorithm has the global convergence property [7,14–16] in the sense that the 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. We also show through numerical experiments that the proposed algorithm is much faster than the multiplicative update rule.

Notation: $\mathbb{R}$ denotes the set of real numbers. $\mathbb{R}_+$ and $\mathbb{R}_{++}$ denote the set of nonnegative and positive real numbers, respectively. For any subset S of $\mathbb{R}$, $S^{I\times J}$ denotes the set of all $I \times J$ matrices such that each entry belongs to S. For example, $\mathbb{R}_+^{I\times J}$ is the set of all $I \times J$ real nonnegative matrices. $\mathbb{N}$ denotes the set of natural numbers or the set of positive integers. $\mathbf{0}_{I\times J}$ and $\mathbf{1}_{I\times J}$ denote the $I \times J$ matrix of all zeros and all ones, respectively. For two matrices $\boldsymbol{A} = (A_{ij})$ and $\boldsymbol{B} = (B_{ij})$ with the same size, the inequality $\boldsymbol{A} \geq \boldsymbol{B}$ means that $A_{ij} \geq B_{ij}$ for all i and j, and $(\boldsymbol{AB})_{ij}$ denotes the (i, j)-th entry of the matrix $\boldsymbol{AB}$, that is, $(\boldsymbol{AB})_{ij} = \sum_k A_{ik}B_{kj}$.

2 Alpha-Divergence Based Nonnegative Matrix Factorization

2.1 Optimization Problem

Suppose we are given an $M \times N$ nonnegative matrix $\boldsymbol{X} = (X_{ij})$. The alpha-divergence based NMF is formulated as the constrained optimization problem:

$$\begin{array}{ll}\text{minimize} & D_\alpha(\boldsymbol{X} \,\|\, \boldsymbol{W}\boldsymbol{H}^{\mathrm{T}}) \\ \text{subject to} & \boldsymbol{W} \in \mathbb{R}_+^{M\times K}, \quad \boldsymbol{H} \in \mathbb{R}_+^{N\times K}\end{array} \tag{1}$$

where

$$D_\alpha(\boldsymbol{X} \,\|\, \boldsymbol{W}\boldsymbol{H}^{\mathrm{T}}) = \frac{1}{\alpha(1-\alpha)} \sum_{i=1}^{M}\sum_{j=1}^{N}\Big[\alpha X_{ij} + (1-\alpha)(\boldsymbol{W}\boldsymbol{H}^{\mathrm{T}})_{ij} - X_{ij}^{\alpha}(\boldsymbol{W}\boldsymbol{H}^{\mathrm{T}})_{ij}^{1-\alpha}\Big] \quad (\alpha \neq 0, 1). \tag{2}$$

When $\alpha > 1$, the right-hand side of (2) is not defined for all nonnegative matrices $\boldsymbol{W}$ and $\boldsymbol{H}$. A simple way to make the problem well-defined is to modify (1) as follows:

$$\begin{array}{ll}\text{minimize} & D_\alpha(\boldsymbol{X} \,\|\, \boldsymbol{W}\boldsymbol{H}^{\mathrm{T}}) \\ \text{subject to} & \boldsymbol{W} \in [\epsilon, \infty)^{M\times K}, \quad \boldsymbol{H} \in [\epsilon, \infty)^{N\times K}\end{array} \tag{3}$$

where ϵ is a positive constant, which is usually set to a small number so that (3) is close to (1). In what follows, we consider (3) instead of (1).

Note that sparse factor matrices can never be obtained from the modified optimization problem (3). However, if we replace all ϵ in the obtained solution with zero, the resulting factor matrices are expected to be sparse because local optimal solutions of (3) are often located at the boundary of the feasible region. In addition, if ϵ is sufficiently small, it is expected that the pair of the resulting factor matrices is close to the original local optimal solution.

When $\alpha < 0$ and $\boldsymbol{X}$ has a zero entry, the right-hand of (2) is not determined. We thus impose throughout this paper the following assumption on $\boldsymbol{X}$.

Assumption 1. All entries of $\boldsymbol{X}$ are positive.

2.2 Properties of the Objective Function

The partial derivatives of $D_\alpha(\boldsymbol{X} \,\|\, \boldsymbol{W}\boldsymbol{H}^{\mathrm{T}})$ with respect to W_{ik} and H_{jk} are given by

$$\frac{\partial D_\alpha}{\partial W_{ik}} = \frac{1}{\alpha}\left(\sum_j H_{jk} - \sum_j X_{ij}^{\alpha} H_{jk} (\boldsymbol{W}\boldsymbol{H}^{\mathrm{T}})_{ij}^{-\alpha}\right),$$

$$\frac{\partial D_\alpha}{\partial H_{jk}} = \frac{1}{\alpha}\left(\sum_i W_{ik} - \sum_i X_{ij}^{\alpha} W_{ik} (\boldsymbol{W}\boldsymbol{H}^{\mathrm{T}})_{ij}^{-\alpha}\right),$$

and the second and third partial derivatives are given by

$$\frac{\partial^2 D_\alpha}{\partial W_{ik}^2} = \sum_j X_{ij}^{\alpha} H_{jk}^2 (\boldsymbol{W}\boldsymbol{H}^{\mathrm{T}})_{ij}^{-\alpha-1},$$

$$\frac{\partial^2 D_\alpha}{\partial H_{jk}^2} = \sum_i X_{ij}^{\alpha} W_{ik}^2 (\boldsymbol{W}\boldsymbol{H}^{\mathrm{T}})_{ij}^{-\alpha-1},$$

$$\frac{\partial^3 D_\alpha}{\partial W_{ik}^3} = -(\alpha+1)\sum_j X_{ij}^{\alpha} H_{jk}^3 (\boldsymbol{W}\boldsymbol{H}^{\mathrm{T}})_{ij}^{-\alpha-2},$$

$$\frac{\partial^3 D_\alpha}{\partial H_{jk}^3} = -(\alpha+1)\sum_i X_{ij}^{\alpha} W_{ik}^3 (\boldsymbol{W}\boldsymbol{H}^{\mathrm{T}})_{ij}^{-\alpha-2}.$$

Under Assumption 1, the second partial derivatives are positive for all $\alpha\ (\neq 0, 1)$ and all pairs of positive matrices $\boldsymbol{W}$ and $\boldsymbol{H}$. Therefore, if we fix all entries of $\boldsymbol{W}$ and $\boldsymbol{H}$ except W_{ik} (H_{jk}, resp.) to constants not less than ϵ then we obtain a function of W_{ik} (H_{jk}, resp.) which is strictly convex on $[\epsilon, \infty)$. In what follows, we express these functions as $f_{ik}(W_{ik})$ and $g_{jk}(H_{jk})$. Then $f'_{ik}(W_{ik})$ and $g'_{jk}(H_{jk})$ are monotone increasing functions, and convex if $\alpha \leq -1$ and concave otherwise.

It is easy to see that the objective function of (3) has the following property.

Lemma 1. For any $\alpha \in \mathbb{R} \setminus \{0, 1\}$, $\epsilon \in \mathbb{R}_{++}$, $\boldsymbol{X} \in \mathbb{R}_{++}^{M \times N}$, $\boldsymbol{W}^* \in [\epsilon, \infty)^{M \times K}$ and $\boldsymbol{H}^* \in [\epsilon, \infty)^{N \times K}$, the level set

$$\{(\boldsymbol{W}, \boldsymbol{H}) \in [\epsilon, \infty)^{M \times K} \times [\epsilon, \infty)^{N \times K} \,|D_\alpha(\boldsymbol{X} \,\|\, \boldsymbol{W}\boldsymbol{H}^{\mathrm{T}}) \leq D_\alpha(\boldsymbol{X} \,\|\, \boldsymbol{W}^*(\boldsymbol{H}^*)^{\mathrm{T}})\}$$

is bounded.

2.3 Optimality Conditions

Let $\mathcal{F}_\epsilon = [\epsilon, \infty)^{M \times K} \times [\epsilon, \infty)^{N \times K}$ be the feasible region of the problem (3). If $(\boldsymbol{W}, \boldsymbol{H}) \in \mathcal{F}_\epsilon$ is a local optimal solution of (3), it must satisfy the following conditions:

$$\forall i, k, \quad \frac{\partial D_\alpha}{\partial W_{ik}} \begin{cases} \geq 0, & \text{if} \ \ W_{ik} = \epsilon, \\ = 0, & \text{if} \ \ W_{ik} > \epsilon, \end{cases} \tag{4}$$

$$\forall j, k, \quad \frac{\partial D_\alpha}{\partial H_{jk}} \begin{cases} \geq 0, & \text{if} \ \ H_{jk} = \epsilon, \\ = 0, & \text{if} \ \ H_{jk} > \epsilon. \end{cases} \tag{5}$$

A point $(\boldsymbol{W}, \boldsymbol{H}) \in \mathcal{F}_\epsilon$ satisfying (4) and (5) is called a stationary point of (3).

3 Proposed Algorithm

The algorithm proposed here for solving the problem (3) is based on the coordinate descent and the Newton method. Let the current values of $\boldsymbol{W}$ and $\boldsymbol{H}$ be $\boldsymbol{W}^{\mathrm{c}} \in [\epsilon, \infty)^{M \times K}$ and $\boldsymbol{H}^{\mathrm{c}} \in [\epsilon, \infty)^{N \times K}$, respectively. We want to minimize the value of the objective function by updating only one variable, say W_{ik}. This problem is formulated as

$$\begin{aligned} &\text{minimize} \ \ f_{ik}(W_{ik}) \\ &\text{subject to} \ W_{ik} \geq \epsilon \end{aligned} \tag{6}$$

where $f_{ik}(W_{ik})$ is the function obtained from $D_\alpha(\boldsymbol{X} \,\|\, \boldsymbol{W}\boldsymbol{H}^{\mathrm{T}})$ by fixing all variables except W_{ik} to the current values. Because $f_{ik}(W_{ik})$ is strictly convex as stated in the previous section, (6) is a convex optimization problem. Therefore, if $f'_{ik}(W^{\mathrm{c}}_{ik}) = 0$ then W^{c}_{ik} is the optimal solution of (6). However, we cannot obtain the optimal solution in a closed form in general. So we apply the Newton method to obtain an approximate solution of $f'_{ik}(W_{ik}) = 0$, which is given by

$$W^{\mathrm{n}}_{ik} = W^{\mathrm{c}}_{ik} - \frac{f'_{ik}(W^{\mathrm{c}}_{ik})}{f''_{ik}(W^{\mathrm{c}}_{ik})}.$$

If $f'_{ik}(W^{\mathrm{n}}_{ik}) = 0$ then W^{n}_{ik} is the minimum point of $f_{ik}(W_{ik})$, and hence $W^{\mathrm{new}}_{ik} = \max\{\epsilon, W^{\mathrm{n}}_{ik}\}$ is the optimal solution of (6). If $f'_{ik}(W^{\mathrm{n}}_{ik}) f'_{ik}(W^{\mathrm{c}}_{ik}) > 0$ then $f_{ik}(W_{ik})$ decreases monotonically as the value of W_{ik} varies from W^{c}_{ik} to W^{n}_{ik}. Hence, letting $W^{\mathrm{new}}_{ik} = \max\{\epsilon, W^{\mathrm{n}}_{ik}\}$, we have $f_{ik}(W^{\mathrm{new}}_{ik}) < f_{ik}(W^{\mathrm{c}}_{ik})$. On the other hand, in the case where $f'_{ik}(W^{\mathrm{n}}_{ik}) f'_{ik}(W^{\mathrm{c}}_{ik}) < 0$, it can occur that $f_{ik}(W^{\mathrm{n}}_{ik}) > f_{ik}(W^{\mathrm{c}}_{ik})$ (see Fig. 1). In order to avoid this situation, we use a linear interpolation of the curve $Y = f'_{ik}(W_{ik})$. We first draw a line

$$Y - f'_{ik}(W^{\mathrm{c}}_{ik}) = \frac{f'_{ik}(W^{\mathrm{c}}_{ik}) - f'_{ik}(W^{\mathrm{n}}_{ik})}{W^{\mathrm{c}}_{ik} - W^{\mathrm{n}}_{ik}} (W_{ik} - W^{\mathrm{c}}_{ik}).$$

on W_{ik}-Y plane, which passes through the points $(W^{\mathrm{c}}_{ik}, f'(W^{\mathrm{c}}_{ik}))$ and $(W^{\mathrm{n}}_{ik}, f'(W^{\mathrm{n}}_{ik}))$ (see the red line in Fig. 1). We then find the point at which the

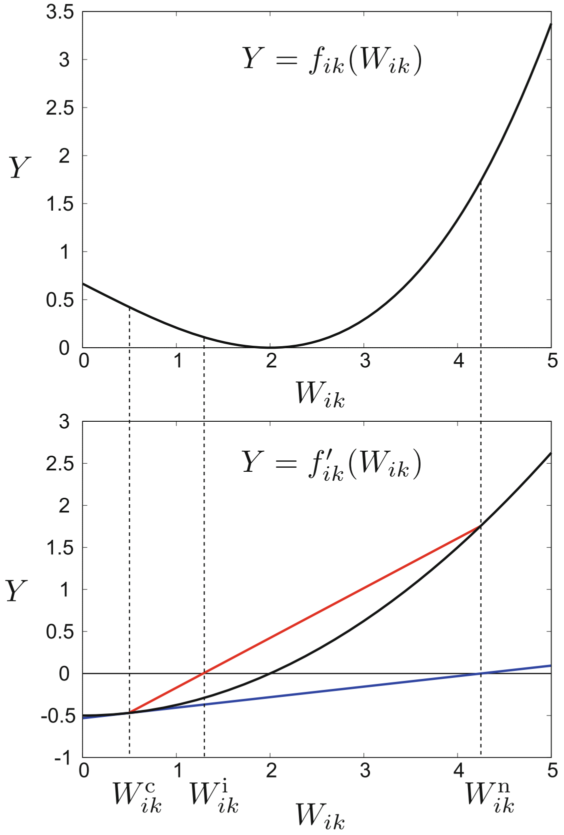

Fig. 1. Update rule for W_{ik} when $f'_{ik}(W_{ik}^{\mathrm{c}})f'_{ik}(W_{ik}^{\mathrm{n}}) < 0$. (Color figure online)

line intersects the W_{ik}-axis, and let the W_{ik}-coordinate of the point be a new approximate solution W_{ik}^{i} of $f'_{ik}(W_{ik}) = 0$, that is,

$$W_{ik}^{\mathrm{i}} = W_{ik}^{\mathrm{c}} - \frac{W_{ik}^{\mathrm{n}} - W_{ik}^{\mathrm{c}}}{f'_{ik}(W_{ik}^{\mathrm{n}}) - f'_{ik}(W_{ik}^{\mathrm{c}})} f'_{ik}(W_{ik}^{\mathrm{c}}).$$

Furthermore, if W_{ik}^{i} is less than ϵ, we replace it with ϵ. Then we have $f'_{ik}(W_{ik}^{\mathrm{i}})f'_{ik}(W_{ik}^{\mathrm{c}}) > 0$ and $f_{ik}(W_{ik}^{\mathrm{i}}) < f_{ik}(W_{ik}^{\mathrm{c}})$.

Figure 2 shows the proposed update rule for W_{ik}, which is based on the idea described above but slightly modified so that a better solution can be obtained.

The problem of minimizing the value of the objective function by updating only H_{jk} is formulated as

$$\begin{array}{ll} \text{minimize} & g_{jk}(H_{jk}) \\ \text{subject to} & H_{jk} \geq \epsilon \end{array} \tag{7}$$

Algorithm 1. Update rule for W_{ik}

```
Require: $\boldsymbol{X} \in \mathbb{R}_{++}^{M \times N}$, $\boldsymbol{W} \in [\epsilon, \infty)^{M \times K}$, $\boldsymbol{H} \in [\epsilon, \infty)^{N \times K}$, $\alpha \in \mathbb{R} \setminus \{0, 1\}$, $\epsilon \in \mathbb{R}_{++}$,
    $i \in \{1, 2, \ldots, M\}$, $k \in \{1, 2, \ldots, K\}$
Ensure: $W_{ik}^{\mathrm{new}} \in [\epsilon, \infty)$
1: If $f'_{ik}(W_{ik}) = 0$ then set $W_{ik}^{\mathrm{new}} \leftarrow W_{ik}$ and go to Step 5. Otherwise set
       $$W_{ik}^{\mathrm{n}} \leftarrow W_{ik} - \frac{f'_{ik}(W_{ik})}{f''_{ik}(W_{ik})}.$$
2: If $W_{ik}^{\mathrm{n}} < \epsilon$ then set $W_{ik}^{\mathrm{n}} \leftarrow \epsilon$.
3: If $f'_{ik}(W_{ik}^{\mathrm{n}}) f'_{ik}(W_{ik}) \geq 0$ then set $W_{ik}^{\mathrm{new}} \leftarrow W_{ik}^{\mathrm{n}}$ and go to Step 5. Otherwise set
       $$W_{ik}^{\mathrm{new}} \leftarrow W_{ik} - \frac{W_{ik}^{\mathrm{n}} - W_{ik}}{f'_{ik}(W_{ik}^{\mathrm{n}}) - f'_{ik}(W_{ik})} f'_{ik}(W_{ik}).$$
4: If $W_{ik}^{\mathrm{new}} < \epsilon$ then set $W_{ik}^{\mathrm{new}} \leftarrow \epsilon$.
5: Return $W_{ik}^{\mathrm{new}}$.
```

Fig. 2. Update rule for W_{ik}.

Algorithm 2. Update rule for H_{jk}.

```
Require: $\boldsymbol{X} \in \mathbb{R}_{++}^{M \times N}$, $\boldsymbol{W} \in [\epsilon, \infty)^{M \times K}$, $\boldsymbol{H} \in [\epsilon, \infty)^{N \times K}$, $\alpha \in \mathbb{R} \setminus \{0, 1\}$, $\epsilon \in \mathbb{R}_{++}$,
    $j \in \{1, 2, \ldots, N\}$, $k \in \{1, 2, \ldots, K\}$
Ensure: $H_{jk}^{\mathrm{new}} \in [\epsilon, \infty)$
1: If $g'_{jk}(H_{ik}) = 0$ then set $H_{jk}^{\mathrm{new}} \leftarrow H_{jk}$ and go to Step 5. Otherwise set
       $$H_{jk}^{\mathrm{n}} \leftarrow H_{jk} - \frac{g'_{jk}(H_{jk})}{g''_{jk}(H_{jk})}.$$
2: If $H_{jk}^{\mathrm{n}} < \epsilon$ then set $H_{jk}^{\mathrm{n}} \leftarrow \epsilon$.
3: If $g'_{jk}(H_{jk}^{\mathrm{n}}) g'_{ik}(H_{jk}) \geq 0$ then set $H_{jk}^{\mathrm{new}} \leftarrow H_{jk}^{\mathrm{n}}$ and go to Step 5. Otherwise set
       $$H_{jk}^{\mathrm{new}} \leftarrow H_{jk} - \frac{H_{jk}^{\mathrm{n}} - H_{jk}}{g'_{jk}(H_{jk}^{\mathrm{n}}) - g'_{jk}(H_{jk})} g'_{jk}(H_{jk}).$$
4: If $H_{jk}^{\mathrm{new}} < \epsilon$ then set $H_{jk}^{\mathrm{new}} \leftarrow \epsilon$.
5: Return $H_{jk}^{\mathrm{new}}$.
```

Fig. 3. Update rule for H_{jk}.

where $g_{jk}(H_{jk})$ is the function obtained from $D_\alpha(\boldsymbol{X} \parallel \boldsymbol{W}\boldsymbol{H}^{\mathrm{T}})$ by fixing all variables except H_{jk} to the current values. Using the same idea as above, we can derive an update rule for H_{jk} as shown in Fig. 3.

It is clear from the derivation of Algorithms 1 and 2 that the following two lemmas hold true.

Lemma 2. If W_{ik} is not the optimal solution of the subproblem (6) then W_{ik}^{new} obtained by Algorithm 1 satisfies $f_{ik}(W_{ik}^{\mathrm{new}}) < f_{ik}(W_{ik})$. Similarly, if H_{jk} is not the optimal solution of the subproblem (7) then H_{jk}^{new} obtained by Algorithm 2 satisfies $g_{jk}(H_{jk}^{\text{new}}) < g_{jk}(H_{jk})$.

Lemma 3. Suppose that $\boldsymbol{X} \in \mathbb{R}_{++}^{M \times N}$, $\alpha \in \mathbb{R} \setminus \{0,1\}$ and $\epsilon \in \mathbb{R}_{++}$ are fixed. Then W_{ik}^{new}, the output of Algorithm 1, depends continuously on $\boldsymbol{W}$ and $\boldsymbol{H}$ for any i and k. Similarly, H_{jk}^{new}, the output of Algorithm 2, depends continuously on $\boldsymbol{W}$ and $\boldsymbol{H}$ for any j and k.

Furthermore, using Zangwill's global convergence theorem [16], we obtain the following theorem.

Theorem 1. Given $\boldsymbol{X} \in \mathbb{R}_{++}^{M \times N}$, $K \in \mathbb{N}$, $\alpha \in \mathbb{R} \setminus \{0,1\}$, $\epsilon \in \mathbb{R}_{++}$, and an initial solution $(\boldsymbol{W}^{(0)}, \boldsymbol{H}^{(0)}) \in \mathcal{F}_\epsilon$, we apply the update rules described by Algorithms 1 and 2 to $MK + NK$ variables in a fixed cyclic order. Let $(\boldsymbol{W}^{(l)}, \boldsymbol{H}^{(l)}) \in \mathcal{F}_\epsilon$ be the solution after l rounds of updates. Then the sequence $\{\boldsymbol{W}^{(l)}, \boldsymbol{H}^{(l)}\}_{l=0}^{\infty}$ has at least one convergent subsequence and the limit of any convergent subsequence is a stationary point of the problem (3).

Proof. Let us express the relation between $(\boldsymbol{W}^{(l)}, \boldsymbol{H}^{(l)})$ and $(\boldsymbol{W}^{(l+1)}, \boldsymbol{H}^{(l+1)})$ by using a mapping $A : \mathcal{F}_\epsilon \to \mathcal{F}_\epsilon$ as follows:

$$(\boldsymbol{W}^{(l+1)}, \boldsymbol{H}^{(l+1)}) = A(\boldsymbol{W}^{(l)}, \boldsymbol{H}^{(l)}).$$

In view of Zangwill's global convergence theorem [16], it suffices to show that the following statements hold true.

1. (Boundedness) For any initial solution $(\boldsymbol{W}^{(0)}, \boldsymbol{H}^{(0)}) \in \mathcal{F}_\epsilon$, the sequence $\{(\boldsymbol{W}^{(l)}, \boldsymbol{H}^{(l)})\}_{l=0}^{\infty}$ belongs to a compact subset of $\mathcal{F}_\epsilon$.
2. (Monotoneness) The objective function $D_\alpha(\boldsymbol{X} \,\|\, \boldsymbol{W}\boldsymbol{H}^{\mathrm{T}})$ satisfies

$$\begin{aligned}(\boldsymbol{W}, \boldsymbol{H}) \notin \mathcal{S}_\epsilon &\Rightarrow D_\alpha(\boldsymbol{X} \,\|\, \boldsymbol{W}'(\boldsymbol{H}')^{\mathrm{T}}) < D_\alpha(\boldsymbol{X} \,\|\, \boldsymbol{W}\boldsymbol{H}^{\mathrm{T}}) \\ (\boldsymbol{W}, \boldsymbol{H}) \in \mathcal{S}_\epsilon &\Rightarrow D_\alpha(\boldsymbol{X} \,\|\, \boldsymbol{W}'(\boldsymbol{H}')^{\mathrm{T}}) \le D_\alpha(\boldsymbol{X} \,\|\, \boldsymbol{W}\boldsymbol{H}^{\mathrm{T}})\end{aligned}$$

   where $\mathcal{S}_\epsilon$ is the set of stationary points of (3) and $(\boldsymbol{W}', \boldsymbol{H}') = A(\boldsymbol{W}, \boldsymbol{H})$.
3. (Continuity) The mapping A is continuous in $\mathcal{F}_\epsilon \setminus \mathcal{S}_\epsilon$.

The monotoneness follows from Lemma 2. The boundedness follows from Lemmas 1 and 2. The continuity follows from Lemma 3. □

By Theorem 1, we can immediately obtain an algorithm that stops within a finite number of rounds by relaxing the optimality condition given by (4) and (5), as shown in Reference [7]. The resulting algorithm is shown in Fig. 4.

Theorem 2. For any input, Algorithm 3 stops within a finite number of rounds.

Algorithm 3. Newton-Type Algorithm for Solving (3)

```
Require: $\boldsymbol{X} \in \mathbb{R}_{++}^{M\times N}$, $K \in \mathbb{N}$, $\alpha \in \mathbb{R} \setminus \{0,1\}$, $\epsilon, \delta_1, \delta_2 \in \mathbb{R}_{++}$, $\delta_3 \in [0, \delta_1)$
Ensure: $\boldsymbol{W} \in [\epsilon, \infty)^{M\times K}$, $\boldsymbol{H} \in [\epsilon, \infty)^{N\times K}$
1: Choose $\boldsymbol{W} \in [\epsilon, \infty)^{M\times K}$ and $\boldsymbol{H} \in [\epsilon, \infty)^{N\times K}$.
2: Update $MK + NK$ variables one by one in a fixed order by using Algorithms 1
   and 2. However, $W_{ik}$ is not updated if the following inequality holds:
       $\begin{cases} \frac{\partial D_\alpha}{\partial W_{ik}} \geq -\delta_3, & \text{if } W_{ik} = \epsilon, \\ \left|\frac{\partial D_\alpha}{\partial W_{ik}}\right| \leq \delta_3, & \text{if } W_{ik} > \epsilon. \end{cases}$
   Similarly, $H_{jk}$ is not updated if the following inequality holds:
       $\begin{cases} \frac{\partial D_\alpha}{\partial H_{jk}} \geq -\delta_3, & \text{if } H_{jk} = \epsilon, \\ \left|\frac{\partial D_\alpha}{\partial H_{jk}}\right| \leq \delta_3, & \text{if } H_{jk} > \epsilon. \end{cases}$
3: If the following conditions are satisfied then return $\boldsymbol{W}$ and $\boldsymbol{H}$, and stop. Otherwise
   go to Step 2.
       $\forall i, j, \quad \begin{cases} \frac{\partial D_\alpha}{\partial W_{ik}} \geq -\delta_1, & \text{if } W_{ik} \in [\epsilon, \epsilon + \delta_2], \\ \left|\frac{\partial D_\alpha}{\partial W_{ik}}\right| \leq \delta_1, & \text{if } W_{ik} > \epsilon + \delta_2, \end{cases}$
       $\forall j, k, \quad \begin{cases} \frac{\partial D_\alpha}{\partial H_{jk}} \geq -\delta_1, & \text{if } H_{jk} \in [\epsilon, \epsilon + \delta_2], \\ \left|\frac{\partial D_\alpha}{\partial H_{jk}}\right| \leq \delta_1, & \text{if } H_{jk} > \epsilon + \delta_2. \end{cases}$
```

Fig. 4. Newton-type algorithm for solving (3).

4 Numerical Experiments

In order to evaluate the efficiency of the proposed algorithm, we applied it to a randomly generated matrix $\boldsymbol{X}$ and compared the results with those obtained using the multiplicative update rule [6,8] described by

$$W_{ik}^{\text{new}} \leftarrow \max\left(\epsilon, W_{ik}\left(\frac{\sum_{j=1}^{N} X_{ij}H_{jk}/(\boldsymbol{W}\boldsymbol{H}^{\mathrm{T}})_{ij}}{\sum_{j=1}^{N} H_{jk}}\right)^{\frac{1}{\alpha}}\right)$$

and the same stopping condition. Although some other methods have been proposed (see [13] for example), we do not consider them because the global convergence is not guaranteed.

In all experiments, $\boldsymbol{X}$ was set to the same 40×20 matrix of which each entry was drawn from an independent uniform distribution on the interval $[0, 1]$. The value of K was set to 5. The values of the parameters in Algorithm 3 were set to $\epsilon = 10^{-6}$, $\delta_1 = 10^{-4}$, $\delta_2 = 10^{-6}$, $u = 1.0$ and $\delta_3 = 0.5 \times \delta_1$. The value of the parameter α in the alpha-divergence was set to -1.5, 0.5 and 2.5. For each value of α, the multiplicative update rules and the proposed algorithms

Table 1. The number of rounds of the multiplicative update (MU) rule and Algorithm 3 for solving (3).

α	Method	Average	Minimum	Maximum
-1.5	MU	16,409.8	2,018	40,000
	Algorithm 3	275.5	143	458
0.5	MU	22,863.5	6,284	35,183
	Algorithm 3	499.6	251	821
2.5	MU	25,873.9	12,517	40,000
	Algorithm 3	706.4	343	1,283

Table 2. Computation time (in second) of the multiplicative update (MU) rule and Algorithm 3 for solving (3).

α	Method	Average	Minimum	Maximum
-1.5	MU	28.965	3.562	70.593
	Algorithm 3	1.406	0.672	2.297
0.5	MU	81.072	22.266	124.704
	Algorithm 3	3.739	1.859	6.109
2.5	MU	45.679	22.094	70.640
	Algorithm 3	3.508	1.703	6.094

were run for 10 times with 10 different initial solutions, which were generated in the same way as $\boldsymbol{X}$ but all entries less than ϵ were replaced with ϵ so that the initial solution belongs to the feasible region of the optimization problem. The maximum number of rounds was set to 40,000, that is, if the solution does not satisfy the stopping condition within 40,000 rounds then the algorithm was forcedly stopped. All algorithms were implemented in C language, compiled with gcc 5.3.0 and tested on a PC with Intel Core i5-4590 and 8 GB RAM.

The results are shown in Tables 1 and 2. It is easily seen from those tables that the proposed algorithm is much faster than the multiplicative update rule.

5 Conclusion

We have proposed a novel iterative algorithm, which is based on the coordinate descent and the Newton method, for NMF with the alpha-divergence. The proposed algorithm not only has the global convergence property like the multiplicative update rule but also is much faster than the multiplicative update rule, as shown in the experimental results in the previous section. Further experiments with various real data should be performed in the near future to evaluate the efficiency of the proposed algorithm.

Acknowledgments. This work was partially supported by JSPS KAKENHI Grant Number JP15K00035.

References

1. Paatero, P., Tapper, U.: Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. Environmetrics **5**(2), 111–126 (1994)
2. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature **401**, 788–792 (1999)
3. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Leen, T.K., Dietterich, T.G., Tresp, V. (eds.) Advances in Neural Information Processing Systems. vol. 13, pp. 556–562 (2001)
4. Févotte, C., Bertin, N., Durrieu, J.L.: Nonnegative matrix factorization with the Itakura-Saito divergence: with application to music analysis. Neural Comput. **21**(3), 793–830 (2009)
5. Févotte, C., Idier, J.: Algorithms for nonnegative matrix factorization with the β-divergence. Neural Comput. **23**(9), 2421–2456 (2011)
6. Yang, Z., Oja, E.: Unified development of multiplicative algorithm for linear and quadratic nonnegative matrix factorization. IEEE Trans. Neural Networks **22**(12), 1878–1891 (2011)
7. Takahashi, N., Hibi, R.: Global convergence of modified multiplicative updates for nonnegative matrix factorization. Comput. Optim. Appl. **57**, 417–440 (2014)
8. Takahashi, N., Katayama, J., Takeuchi, J.: A generalized sufficient condition for global convergence of modified multiplicative updates for NMF. In: Proceedings of 2014 International Symposium on Nonlinear Theory and Its Applications. pp. 44–47 (2014)
9. Kim, J., He, Y., Park, H.: Algorithms for nonnegative matrix and tensor factorization: a unified view based on block coordinate descent framework. J. Global Optim. **58**(2), 285–319 (2014)
10. Hansen, S., Plantenga, T., Kolda, T.G.: Newton-based optimization for Kullback-Leibler nonnegative tensor factorizations. Optim. Methods Softw. **30**(5), 1002–1029 (2015)
11. Amari, S.I.: Differential-Geometrical Methods in Statistics. Springer, New York (1985)
12. Cichocki, A., Zdunek, R., Amari, S.I.: Csiszar's divergences for non-negative matrix factorization: family of new algorithms. In: Proceedings of the 6th International Conference on Independent Component Analysis and Signal Separation, pp. 32–39 (2006)
13. Cichocki, A., Zdunek, R., Phan, A.H., Amari, S.I.: Nonnegative Matrix and Tensor Factorizations. Wiley, West Sussex (2009)
14. Kimura, T., Takahashi, N.: Global convergence of a modified HALS algorithm for nonnegative matrix factorization. In: Proceedings of 2015 IEEE 6th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, pp. 21–24 (2015)
15. Takahashi, N., Seki, M.: Multiplicative update for a class of constrained optimization problems related to NMF and its global convergence. In: Proceedings of 2016 European Signal Processing Conference, pp. 438–442 (2016)
16. Zangwill, W.: Nonlinear Programming: A Unified Approach. Prentice-Hall, Englewood Cliffs (1969)