NON-NEGATIVE MATRIX FACTORIZATION USING BREGMAN MONOTONE OPERATOR SPLITTING

Kenta Niwa and Noboru Harada

NTT Media Intelligence Laboratories, Japan

ABSTRACT

A non-negative matrix factorization (NMF) algorithm based on Bregman monotone operator splitting (B-MOS) is proposed. Several commonly used NMF algorithms, such as the multiplicative update method, are often used in source separation for speech and image signals. To improve the convergence rate in the tail, applying the alternating direction method of multipliers (ADMM) is reported to be effective. However, a fixed step-size parameter has to be carefully chosen for fast and stable convergence. Our main idea to overcome this issue is to adaptively modify the variable space metric so that it matches the cost convexity. Besides this, selecting an appropriate MOS (e.g., Peaceman-Rachford splitting) instead of the Douglas-Rachford splitting used in the ADMM may effectively improve the convergence rate further. To realize these ideas w.r.t. adaptive metric modification and appropriate operator splitting selection, we apply B-MOS to the NMF problem and obtain a new NMF solver in this paper. Results of numerical experiments demonstrate that the proposed NMF solver with B-MOS improved the convergence rate in the tail.

***Index Terms*—** Non-negative matrix factorization (NMF), convex optimization, Bregman divergence, multiplicative iterative method, alternating direction method of multipliers (ADMM)

1. INTRODUCTION

Non-negative matrix factorization (NMF) [1, 2] is used to find relatively few bases and their activations from noisy observation data. It is practically used in many applications, such as source separation in speech and image signals. In some applications, it is demanded to shorten the processing time, including real-time processing [3, 4, 5, 6]. Therefore, the goal of this study is to construct an NMF algorithm with fast and stable convergence.

Many studies have used the multiplicative update method [1]. The cost function to be minimized is designed by a *bi-convex* function, i.e., it is convex with respect to one variable (e.g., bases) while fixing another variable (e.g., activations). When variables are alternately updated in accordance with the multiplicative update method, the gradient step-size is selected to make the update procedure simple while taking its majorization function into account [1]. Although the multiplicative update method is easy to implement, issues remain such as a low convergence rate in the tail and difficulties in constructing solvers when a combination of loss and normalization terms is complicated.

To overcome these issues, Sun and Fevotte reported that applying the alternating direction method of multipliers (ADMM) [7] to NMF problems was effective [8]. In this approach, the cost formulation is modified to be a linearly constrained minimization problem by replacing the NMF output, which is the product of two matrix variables, with an auxiliary variable. The ADMM is an applicable solver that is equivalent to applying the Douglas-Rachford splitting [9, 10] to the dual problem. However, a gradient step-size needs to be carefully chosen to obtain fast and stable convergence in the tail. Moreover, the convergence rate will be further improved by using another monotone operator splitting (MOS), such as the Peaceman-Rachford splitting [11]. However, also in this case, a gradient step-size needs to be adjusted.

As an advanced MOS, the Bregman-MOS (B-MOS) [12] is applied to the NMF problem in this paper. As used in the ADMM-based conventional method [8], the cost formulation follows a linearly constrained minimization, and we will solve its dual problem. In the traditional Douglas-Rachford splitting and Peaceman Rachford splitting, the variable space has been latently defined by a scaled Euclidean metric and its scale parameter corresponds to the gradient step-size to be adjusted. B-MOS generalizes the variable space metric by using Bregman divergence [13] instead of a scaled Euclidean. By designing Bregman divergence such that it adaptively matches the cost convexity and applying the B-MOS to the NMF problem, we will obtain a fast and stable convergence NMF algorithm without careful step-size selection. Through numerical experiments, we will compare its convergence rate with those of several conventional NMF solvers.

The rest of this paper is organized as follows. The problem formulation and conventional NMF solvers are explained in Sec. 2. The proposed solver based on B-MOS is provided in Sec. 3. After investigating the convergence rates through several experiments in Sec. 4, we conclude this paper in Sec. 5.

2. CONVENTIONAL METHODS

The NMF cost form is defined in Sec. 2.1. The conventional solvers for NMF problems are explained in Sec. 2.2.

2.1. Basic cost formulation

Let us suppose that a non-negative observed signal in a matrix form $\mathbf{Y} \in \mathbb{R}^{I \times J}$ is given. It is composed of a non-negative matrix $\mathbf{Z} \in \mathbb{R}^{I \times J}$ and noise $\mathbf{E} \in \mathbb{R}^{I \times J}$, where $\mathbf{Z}$ is decomposed into non-negative factor matrices $\mathbf{A} \in \mathbb{R}^{I \times K}$ and $\mathbf{B} \in \mathbb{R}^{J \times K}$ as

$$\mathbf{Y} = \mathbf{Z} + \mathbf{E} = \mathbf{A}\mathbf{B}^{\mathrm{T}} + \mathbf{E} \tag{1}$$

where K is assumed to be relatively small and $^{\mathrm{T}}$ denotes the transposition. As a basic formulation, the cost function $\mathcal{J}(\mathbf{A}, \mathbf{B})$ is composed of (i) a loss term G that measures differences between $\mathbf{Z} = \mathbf{A}\mathbf{B}^{\mathrm{T}}$ and $\mathbf{Y}$ and (ii) regularization terms H_A and H_B, which represent the probabilistic assumption for each variable:

$$\mathcal{J}(\mathbf{A}, \mathbf{B}) = G(\mathbf{Y} \| \mathbf{A}\mathbf{B}^{\mathrm{T}}) + H_A(\mathbf{A}) + H_B(\mathbf{B}) \tag{2}$$

where regularization terms are designed so that they strictly satisfy non-negative definitions of variables. The aim of the optimization procedure is to find a set of variables $\{\mathbf{A}, \mathbf{B}\}$ that minimizes the cost while satisfying the non-negative definition as

$$\inf_{\mathbf{A} \geq \mathbf{0}, \mathbf{B} \geq \mathbf{0}} \mathcal{J}(\mathbf{A}, \mathbf{B}). \tag{3}$$

For a loss term G, any convex, closed, and proper (CCP) functions with respect to $\mathbf{Z}$ are assumed to be used. Since two variables are multiplied as $\mathbf{Z}=\mathbf{A}\mathbf{B}^{\mathrm{T}}$, the cost function is in a bi-convex function of $\mathbf{A}$ and $\mathbf{B}$. Then, $\mathcal{J}$ is a CCP function of a variable (e.g., $\mathbf{A}$) while fixing another variable (e.g., $\mathbf{B}$). Since the element separable cost form is often utilized in the field of NMF, we implement it by using the Bregman divergence form [13] as

$$G_{\Phi}(\mathbf{Y}\|\mathbf{Z})=\sum_{ij}\left(\Phi(y_{ij})-\Phi(z_{ij})-\nabla\Phi(z_{ij})(y_{ij}-z_{ij})\right) \quad (4)$$

where any differentiable strictly convex function (e.g., [14]) can be used for Φ and ∇ denotes the differential operator. Since the β-divergence [15, 16] is a subclass of the Bregman divergence (e.g. [2]), it is derived by selecting Φ as follows:

$$\Phi^{(\beta)}(z)=\begin{cases}\frac{z^{\beta+1}}{\beta(\beta+1)}-\frac{z}{\beta}+\frac{1}{\beta+1} & (\beta>0)\\ z\log z-z+1 & (\beta=0)\\ z-\log z-1 & (\beta=-1).\end{cases} \quad (5)$$

Then, the β-divergence is obtained by

$$G_{\Phi^{(\beta)}}(\mathbf{Y}\|\mathbf{Z})=\begin{cases}\sum_{ij}\left(y_{ij}\frac{y_{ij}^{\beta}-z_{ij}^{\beta}}{\beta}-\frac{y_{ij}^{\beta+1}-z_{ij}^{\beta+1}}{\beta+1}\right) & (\beta>0)\\ \sum_{ij}\left(y_{ij}\log\left(\frac{y_{ij}}{z_{ij}}\right)-y_{ij}+z_{ij}\right) & (\beta=0)\\ \sum_{ij}\left(\log\left(\frac{z_{ij}}{y_{ij}}\right)+\frac{y_{ij}}{z_{ij}}-1\right) & (\beta=-1).\end{cases} \quad (6)$$

where we obtain the standard squared Euclidean distance when $\beta=1$, the generalized Kullback-Leibler divergence (I-divergence) when $\beta=0$, and the Itakura-Saito distance when $\beta=-1$, respectively. Other loss formulations are summarized in [2].

For the regularization terms, their design is required to make variable definition non-negative at least. The non-negative barrier function, which outputs zero when all matrix elements are non-negative, is a choice of $H_A(\mathbf{A})=\sum_{ik}\delta_{(a_{ik}\geq 0)}(a_{ik})$ where

$$\delta_{(a_{ik}\geq 0)}(a_{ik})=\begin{cases}0 & (a_{ik}\geq 0)\\ +\infty & (\text{otherwise}).\end{cases} \quad (7)$$

For a sparse representation of variables [17], adding L_1 norm to (7) will be effective. H_B is also designed to represent the statistical property of $\mathbf{B}$.

2.2. Conventional solvers

Several conventional optimization methods for the NMF problem are explained. First, a well-known multiplicative update method [1] is applied to (3). In accordance with the standard gradient descent, each variable is alternately updated as

$$a_{ik}\leftarrow a_{ik}-\eta_{ik}\,\partial_{a_{ik}}\mathcal{J}(\mathbf{A},\mathbf{B}),\ b_{jk}\leftarrow b_{jk}-\eta_{jk}\,\partial_{b_{jk}}\mathcal{J}(\mathbf{A},\mathbf{B}) \quad (8)$$

where η_{ik} is a gradient step-size and the subgradient is given by $\partial_{a_{ik}}\mathcal{J}=\partial\mathcal{J}/\partial a_{ik}$. As an example, the β-divergence (6) is used for G. Since G, H_A and H_B are assumed to be separable for each element, the subgradient of $\mathcal{J}$ is given by

$$\partial_{a_{ik}}\mathcal{J}(\mathbf{A},\mathbf{B})=\sum_{j=1}^{J}\left(y_{ij}/[\mathbf{A}\mathbf{B}^{\mathrm{T}}]_{ij}^{1-\beta}\right)b_{jk}-\partial_{a_{ik}}H_A(\mathbf{A}), \quad (9)$$

$$\partial_{b_{jk}}\mathcal{J}(\mathbf{A},\mathbf{B})=\sum_{i=1}^{I}\left(y_{ij}/[\mathbf{A}\mathbf{B}^{\mathrm{T}}]_{ij}^{1-\beta}\right)a_{ik}-\partial_{b_{jk}}H_B(\mathbf{B}). \quad (10)$$

By selecting a step-size as $\eta_{ik}=a_{ik}/\sum_{j=1}^{J}[\mathbf{A}\mathbf{B}^{\mathrm{T}}]_{ij}^{\beta}b_{jk}$ and $\eta_{jk}=b_{jk}/\sum_{i=1}^{I}[\mathbf{A}\mathbf{B}^{\mathrm{T}}]_{ij}^{\beta}a_{ik}$, the update rule in a matrix form is given by

$$a_{ik}\leftarrow a_{ik}\frac{\left[\sum_{j=1}^{J}\left(y_{ij}/[\mathbf{A}\mathbf{B}^{\mathrm{T}}]_{ij}^{1-\beta}\right)b_{jk}-\partial_{a_{ik}}H_A(\mathbf{A})\right]_{+}}{\sum_{j=1}^{J}[\mathbf{A}\mathbf{B}^{\mathrm{T}}]_{ij}^{\beta}b_{jk}+\varepsilon}, \quad (11)$$

$$b_{jk}\leftarrow b_{jk}\frac{\left[\sum_{i=1}^{I}\left(y_{ij}/[\mathbf{A}\mathbf{B}^{\mathrm{T}}]_{ij}^{1-\beta}\right)a_{ik}-\partial_{b_{jk}}H_B(\mathbf{B})\right]_{+}}{\sum_{i=1}^{I}[\mathbf{A}\mathbf{B}^{\mathrm{T}}]_{ij}^{\beta}a_{ik}+\varepsilon} \quad (12)$$

where $[\mathbf{Z}]_{+}=\max\{\mathbf{Z},\mathbf{0}\}$. However, it has been reported that its convergence rate in the tail is relatively slow (e.g., [17, 18, 19]).

To overcome this issue, applying ADMM to the NMF problem may be effective, as reported in [8]. The main idea of this method is to replace the NMF output $\mathbf{A}\mathbf{B}^{\mathrm{T}}$ by $\mathbf{Z}$ and modify the cost function by a linearly constrained minimization form as:

$$\inf_{\substack{\mathbf{Z}\geq\mathbf{0},\\ \mathbf{A}\geq\mathbf{0},\mathbf{B}\geq\mathbf{0}}} G(\mathbf{Y}\|\mathbf{Z})+H_Z(\mathbf{Z})+H_A(\mathbf{A})+H_B(\mathbf{B}) \quad \text{s.t. } \mathbf{Z}=\mathbf{A}\mathbf{B}^{\mathrm{T}} \quad (13)$$

where H_Z is a normalization term for $\mathbf{Z}$. The formulation (13) is different from that of [8] because we want to use arbitrary normalization terms for H_Z, H_A, H_B. By using a dual variable $\mathbf{U}\in\mathbb{R}^{I\times J}$, the Lagrange function associated with (13) is given by

$$\mathcal{L}(\mathbf{Z},\mathbf{A},\mathbf{B},\mathbf{U})=G(\mathbf{Y}\|\mathbf{Z})+H_Z(\mathbf{Z})+H_A(\mathbf{A})+H_B(\mathbf{B})+\left\langle\mathbf{U},\mathbf{Z}-\mathbf{A}\mathbf{B}^{\mathrm{T}}\right\rangle \quad (14)$$

where $\langle\mathbf{U},\mathbf{Z}\rangle=\mathrm{tr}(\mathbf{U}^{\mathrm{T}}\mathbf{Z})$. The ADMM is a solver for the dual problem of a linearly constrained minimization, given

$$\sup_{\mathbf{U}}\inf_{\substack{\mathbf{Z}\geq\mathbf{0},\\ \mathbf{A}\geq\mathbf{0},\mathbf{B}\geq\mathbf{0}}}\mathcal{L}(\mathbf{Z},\mathbf{A},\mathbf{B},\mathbf{U}). \quad (15)$$

The update procedure is not described in detail here because the proposed method will provide a generalized solver including ADMM. Sun and Fevotte reported that applying ADMM effectively improved the tail convergence rate. However, a gradient step-size had to be carefully chosen to obtain a fast and stable convergence [8].

3. PROPOSED METHOD

For an NMF algorithm that improves the convergence rate without needing a step-size to be carefully chosen, B-MOS is applied to the dual problem (15). After the basic procedure is provided in Sec. 3.1, the algorithm of the proposed NMF solver is explained in Sec. 3.2.

3.1. Proposed algorithm using B-MOS

B-MOS [12] is a generalization of MOS solvers that will be effective for fast convergence by appropriately modifying the variable space metric. The conventional ADMM is equivalent to applying Douglas-Rachford splitting to the dual problem as in (15). Then, it needs a gradient step-size to be carefully chosen because the variable space metric is defined by a scaled Euclidean. Since B-MOS generalizes MOS by using Bregman divergence, the step-size selection will be eliminated if Bregman divergence is designed so that it adaptively matches the cost convexity. (The details of Bregman divergence design are given in Sec. 3.2.) In addition to the adaptive metric modification, selecting an appropriate MOS, such as Peaceman-Rachford splitting [11], will also be effective for fast convergence. In our B-MOS based NMF algorithm, generalized Peaceman-Rachford splitting and Douglas-Rachford splitting are used.

To apply B-MOS to the dual problem, (15) is transformed into the minimization of the sum of CCP functions as

$$\begin{aligned}\sup_{\mathbf{U}}\inf_{\substack{\mathbf{Z}\geq\mathbf{0},\\ \mathbf{A}\geq\mathbf{0},\mathbf{B}\geq\mathbf{0}}}\mathcal{L}(\mathbf{Z},\mathbf{A},\mathbf{B},\mathbf{U})&=\sup_{\mathbf{U}}\Big[-\sup_{\mathbf{Z}\geq\mathbf{0}}\Big(-\langle\mathbf{U},\mathbf{Z}\rangle-G(\mathbf{Y}\|\mathbf{Z})\\ &\quad-H_Z(\mathbf{Z})\Big)-\sup_{\mathbf{A}\geq\mathbf{0},\mathbf{B}\geq\mathbf{0}}\Big(\left\langle\mathbf{U},\mathbf{A}\mathbf{B}^{\mathrm{T}}\right\rangle-H_A(\mathbf{A})-H_B(\mathbf{B})\Big)\Big]\\ &=-\inf_{\mathbf{U}}\left(G^{\star}(-\mathbf{U})+H^{\star}(\mathbf{U})\right)\end{aligned} \quad (16)$$

where the convex conjugate functions [20] w.r.t. $G(\mathbf{Y}\|\mathbf{Z})+H_Z(\mathbf{Z})$ and $H_A(\mathbf{A})+H_B(\mathbf{B})$ are respectively denoted by $G^\star$ and $H^\star$ as

$$G^\star(-\mathbf{U}) = \sup_{\mathbf{Z}\geq\mathbf{0}}\Big(-\langle\mathbf{U},\mathbf{Z}\rangle - G(\mathbf{Y}\|\mathbf{Z}) - H_Z(\mathbf{Z})\Big), \tag{17}$$

$$H^\star(\mathbf{U}) = \sup_{\mathbf{A}\geq\mathbf{0},\mathbf{B}\geq\mathbf{0}}\Big(\Big\langle\mathbf{U},\mathbf{A}\mathbf{B}^{\mathrm{T}}\Big\rangle - H_A(\mathbf{A}) - H_B(\mathbf{B})\Big). \tag{18}$$

For a simple notation, the subdifferential of each convex conjugate function is denoted by $T_1(\mathbf{U}) = -\partial G^\star(-\mathbf{U})$ and $T_2(\mathbf{U}) = \partial H^\star(\mathbf{U})$, where it maps from an input matrix to an output matrix, i.e., $T_i : \mathbb{R}^{I\times J}\rightarrow\mathbb{R}^{I\times J}$. For the problem (16), the fixed point is obtained when its subgradient includes a zero matrix as:

$$\mathbf{0} \in T_1(\mathbf{U}) + T_2(\mathbf{U}) \tag{19}$$

where $\in$ indicates that the output through T_i might be multivalued at discontinuous points. To modify its variable space metric, we introduce a differentiable strictly convex function $D\colon \mathbb{R}^{I\times J}\rightarrow\mathbb{R}\cup\{\infty\}$ whose gradient satisfies $\nabla D(\mathbf{0})=\mathbf{0}$. This is because applying the inverse operator of ∇D, i.e., $(\nabla D)^{-1}\colon \mathbb{R}^{I\times J}\rightarrow\mathbb{R}^{I\times J}$, to both sides of (19) will not affect the fixed point as

$$\mathbf{0} \in (\nabla D)^{-1}\circ T_1(\mathbf{U}) + (\nabla D)^{-1}\circ T_2(\mathbf{U}) \tag{20}$$

where $\circ$ synthesizes two operators, e.g., $(\nabla D)^{-1}$ and T_i.

Reformulation of (20) results in the B-MOS algorithms as summarized in [12]. As a result, we use the Bregman Peaceman-Rachford (B-P-R) splitting and the Bregman Douglas-Rachford (B-D-R) splitting in this paper. As a preliminary step, we introduce a dual auxiliary variable $\mathbf{X}$ that satisfies $\mathbf{U}\in R_1(\mathbf{X})$, where R_i is the D-resolvent operator [21] defined by $R_i=(I+(\nabla D)^{-1}\circ T_i)^{-1}$. The recursive variable update forms of B-P-R and B-D-R splitting are respectively given by reformulating (20) by

$$\mathbf{X} \in C_{T_2}\circ C_{T_1}(\mathbf{X}) \qquad \text{(B-P-R splitting)} \tag{21}$$

$$\mathbf{X} \in \xi C_{T_2}\circ C_{T_1}(\mathbf{X}) + (1-\xi)(\mathbf{X}) \qquad \text{(B-D-R splitting)} \tag{22}$$

where the D-Cayley operator [12] is defined by $C_i=\big(I+(\nabla D)^{-1}\circ T_i\big)^{-1}\circ\big(I-(\nabla D)^{-1}\circ T_i\big)=2R_i-I$ and $\xi\in(0,1)$ is an averaging coefficient. Since applying the averaged operator to (21) results in (22), the convergence rate with B-P-R will be faster than that with B-D-R. By decomposing the recursive updates (21) and (22) into simpler procedures by using other dual auxiliary variables $\mathbf{V},\mathbf{W}$, the recursive update rules are reorganized by

$$\mathbf{U}\leftarrow R_1(\mathbf{X}) = \big(I+(\nabla D)^{-1}\circ T_1\big)^{-1}(\mathbf{X}) \tag{23}$$

$$\mathbf{V}\leftarrow C_1(\mathbf{X}) = 2\mathbf{U}-\mathbf{X} \tag{24}$$

$$\mathbf{W}\leftarrow R_2(\mathbf{V}) = (I+(\nabla D)^{-1}\circ T_2)^{-1}(\mathbf{V}) \tag{25}$$

$$\mathbf{X}\leftarrow\begin{cases} C_2(\mathbf{V})=2\mathbf{W}-\mathbf{V} & \text{(B-P-R splitting)}\\ \xi C_2(\mathbf{V})+(1-\xi)\mathbf{X}=\xi(2\mathbf{W}-\mathbf{V})+(1-\xi)\mathbf{X} & \text{(B-D-R splitting).}\end{cases} \tag{26}$$

However, two issues remain: (i) how to update a primal-dual variable by using the D-resolvent operator in (23) and (25) and (ii) how to design a strictly convex function D to effectively modify the metric of variable space. When we select the Frobenius norm as $D(\mathbf{U})=\frac{1}{2\kappa}\|\mathbf{U}\|_F^2$ ($\kappa>0$), the algorithm (23)-(26) results in the traditional Peaceman-Rachford and Douglas-Rachford splitting. By selecting $D(\mathbf{U})$ in such a way that it adaptively matches the cost convexity, not only does careful step-size selection become unnecessary but also fast and stable convergence is obtained. Algorithm implementations associated with these issues are briefly explained in the next subsection.

Algorithm 1 Proposed algorithm based on B-P-R/B-D-R splitting

Initialization of $\mathbf{A},\mathbf{B},\mathbf{Z},\tilde{\mathbf{X}}$

for $t=1,\ldots,T$

$\mathbf{Z}\leftarrow\arg\min_{\mathbf{Z}\geq\mathbf{0}}\Big(G(\mathbf{Y}\|\mathbf{Z})+H_Z(\mathbf{Z})+B_{D^\dagger}(\mathbf{Z}\|\tilde{\mathbf{X}})\Big)$

$\tilde{\mathbf{V}}\leftarrow\tilde{\mathbf{X}}-2\mathbf{Z}$

$\mathbf{A}\leftarrow\arg\min_{\mathbf{A}\geq\mathbf{0}}\Big(H_A(\mathbf{A})+B_{D^\dagger}(\mathbf{A}\mathbf{B}^{\mathrm{T}}\|-\tilde{\mathbf{V}})\Big)$

$\mathbf{B}\leftarrow\arg\min_{\mathbf{B}\geq\mathbf{0}}\Big(H_B(\mathbf{B})+B_{D^\dagger}(\mathbf{A}\mathbf{B}^{\mathrm{T}}\|-\tilde{\mathbf{V}})\Big)$

$\tilde{\mathbf{X}}\leftarrow\begin{cases}\tilde{\mathbf{V}}+2\mathbf{A}\mathbf{B}^{\mathrm{T}} & \text{(B-P-R splitting)}\\ \xi\tilde{\mathbf{X}}+(1-\xi)(\tilde{\mathbf{V}}+2\mathbf{A}\mathbf{B}^{\mathrm{T}}) & \text{(B-D-R splitting)}\end{cases}$

end for

3.2. Update rule and D-design for fast convergence rate

A primal-dual variable update rule following (23)-(26) is given. As an implementation of (23), an alternate update rule for primal-dual variables $\{\mathbf{Z},\mathbf{U}\}$ is often used (e.g., [22, 23]). Although the derivation is shown in Appendix, procedures (23), (24) for the nonlinearly transformed dual auxiliary variables $\mathbf{V}=(\nabla D)^{-1}(\tilde{\mathbf{V}})$, $\mathbf{X}=(\nabla D)^{-1}(\tilde{\mathbf{X}})$ result in

$$\mathbf{Z}\leftarrow\arg\min_{\mathbf{Z}\geq\mathbf{0}}\Big(G(\mathbf{Y}\|\mathbf{Z})+H_Z(\mathbf{Z})+B_{D^\dagger}(\mathbf{Z}\|\tilde{\mathbf{X}})\Big) \tag{27}$$

$$\tilde{\mathbf{V}}\leftarrow\tilde{\mathbf{X}}-2\mathbf{Z} \tag{28}$$

where the Bregman divergence $B_{D^\dagger}$ is used to generalize a variable space metric and it includes a strictly convex function $D^\dagger$ that satisfies $\nabla D^\dagger=(\nabla D)^{-1}$ as

$$B_{D^\dagger}(\mathbf{Z}\|\tilde{\mathbf{X}}) = D^\dagger(\mathbf{Z}) - D^\dagger(\tilde{\mathbf{X}}) - \Big\langle(\nabla D)^{-1}(\tilde{\mathbf{X}}),\mathbf{Z}-\tilde{\mathbf{X}}\Big\rangle. \tag{29}$$

The update rule for other procedures in (25), (26) is also given in the same way. The overall update procedure that follows (23)-(26) is summarized in **Algorithm 1**.

Next, an appropriate D-design to obtain fast convergence without a gradient step-size adjustment is explained. Since a Euclidean metric $D(\mathbf{Z})=\frac{1}{2\kappa}\|\mathbf{Z}\|_F^2$ is latently used in the conventional ADMM, it needs to choose κ to control the step-size. To make its tuning process unnecessary, D is designed such that it adaptively matches the cost convexity as the Newton or accelerated gradient descent (AGD) methods. However, since the number of matrix elements is huge, the second-order gradient for each element is heavy to calculate. For a simple D-design, we approximate the convexity property of G by a local quadratic function around the present variable z_{ij}^{old}. When the cost function is given by an element separable form as in (4), G is approximated by a sum of element separable quadratic functions:

$$D^\dagger(\mathbf{Z}) = \frac{1}{2}\sum_{ij}(h_{ij}+\epsilon)z_{ij}^2 \tag{30}$$

where the second-order convexity $h_{ij}=\nabla_{ij}^2 G(z_{ij}^{\text{old}})$ is updated for each iteration, $D(\mathbf{Z})=\frac{1}{2}\sum_{ij}\frac{1}{h_{ij}+\epsilon}z_{ij}^2$ that satisfies $\nabla D(\mathbf{0})=\mathbf{0}$, and a small coefficient $\epsilon(>0)$ is used to make D strictly convex.

To reduce updating time for the procedure, we further impose a restriction on h_{ij}. We noticed that $\{\mathbf{A},\mathbf{B}\}$-update in **Algorithm 1** is computationally heavy because the matrix product is included in the penalty term of the Bregman divergence. To make the matrix product simple, h_{ij} is imposed to be the (i,j)-th element of $\mathbf{H}=\mathbf{h}_A\mathbf{h}_B^{\mathrm{T}}$ where $\mathbf{h}_A\in\mathbb{R}^I$ and $\mathbf{h}_B\in\mathbb{R}^J$. As an implementation, K-element averaged second-order convexity was used for $\{\mathbf{h}_A,\mathbf{h}_B\}$.

Note that the update rule based on the conventional ADMM is also given by **Algorithm 1**, then the B-D-R splitting with $\xi=0.5$ and a Euclidean metric $D(\mathbf{Z})=\frac{1}{2\kappa}\|\mathbf{Z}\|_F^2=\frac{1}{2\kappa}\sum_{ij}z_{ij}^2$ are selected.

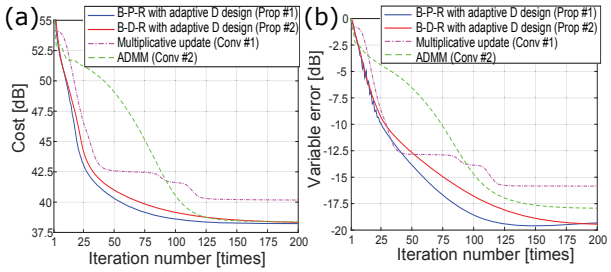

Fig. 1. Convergence curves measured with (a) cost function $\mathcal{J}$ and (b) variable error E

4. NUMERICAL EXPERIMENTS

4.1. Experimental conditions

To evaluate the proposed algorithms, we conducted experiments by using artificially generated data with a certain size $\{I, J, K\} = \{400, 1000, 50\}$. Assuming that ground-truth matrices $\{\mathbf{A}_{\mathrm{GT}}, \mathbf{B}_{\mathrm{GT}}\}$ are sparse, their elements were generated so that $10\,\%$ of their elements have non-zero values. After adding uniform noise $\mathbf{E}$, the prior SNR of $\mathbf{Y}$ was $10\log_{10}(\|\mathbf{A}_{\mathrm{GT}}\mathbf{B}_{\mathrm{GT}}^{\mathrm{T}}\|_F^2/\|\mathbf{E}\|_F^2)$=20.3 [dB]. In a problem to estimate $\{\mathbf{A}, \mathbf{B}\}$, the I-divergence in (6) (β=0) was used for a loss term and L_1 norm was used for a regularization term as $H_A(\mathbf{A}) = \sum_{ik}\big(\delta_{(a_{ik}\geq 0)}(a_{ik}) + \mu_A|a_{ik}|\big)$ where $\mu_A = 0.05$. The same form was used for another regularization term $H_B(\mathbf{B})$.

As summarized in **Algorithm 1**, our proposed algorithms are (Prop #1) the B-P-R splitting with adaptive D-design (30) and (Prop #2) the B-D-R splitting with adaptive D-design (30). They are both applied to a majorization of cost function $\mathcal{J}$. For the comparison methods, we selected (Conv #1) the multiplicative iterative method (11)-(12) and (Conv #2) ADMM, but the detailed update procedure of these methods is different from that of the conventional ADMM-based NMF [8] because the cost form was changed. As noted in Sec. 3.2, the update rule in ADMM is given by **Algorithm 1** where the B-D-R splitting with $\xi = 0.5$ and a Euclidean metric $D(\mathbf{Z}) = \frac{1}{2\kappa}\|\mathbf{Z}\|_F^2$ were selected. Since the convergence rate with ADMM was significantly changed with the parameter selection, $\kappa = 0.25$ was selected that works the most stably among manually adjusted values. For proposed methods, step-size parameter does not need to be selected unlike in ADMM.

For evaluation measures, (i) the cost value $\mathcal{J}(\mathbf{A}, \mathbf{B})$ and (ii) the variable error $E(\mathbf{A}, \mathbf{B})$ were calculated by using ground-truth matrices for each iteration:

$$\mathcal{J}(\mathbf{A}, \mathbf{B}) = G(\mathbf{A}_{\mathrm{GT}}\mathbf{B}_{\mathrm{GT}}^{\mathrm{T}}\|\mathbf{A}\mathbf{B}^{\mathrm{T}}) + H_A(\mathbf{A}) + H_B(\mathbf{B}),$$
$$E(\mathbf{A}, \mathbf{B}) = \frac{1}{2IJ}\|\mathbf{A}_{\mathrm{GT}}\mathbf{B}_{\mathrm{GT}}^{\mathrm{T}} - \mathbf{A}\mathbf{B}^{\mathrm{T}}\|_F^2.$$

Since the update procedure time for an iteration was different for each method, evaluation scores and processing time were recorded. The algorithms were implemented in Matlab and T=200 iterations were executed with a CPU (Intel Core i7 2.4 GHz).

4.2. Experimental results

Fig. 1 shows the convergence curves where evaluation measures were given by the cost function and the variable error. From these convergence curves, Prop #1 (B-P-R splitting) was the fastest and Prop #2 (B-D-R splitting) was the second fastest when the iteration number was used on the horizontal axis. Although the computation time was the shortest with Conv #1 (multiplicative update) as shown in **Table 1**, the convergence rate in the tail deteriorated as reported by Sun and Fevotte [8]. The results eventually obtained with the Conv #2 (ADMM) were better than those obtained with Conv #1. The proposed methods maintain their convergence rates even in the tail and obtained better estimation results than the conventional methods.

Table 1. Evaluation scores after T=200 iteration times

	Prop #1	Prop #2	Conv #1	Conv #2
Processing time [sec]	97.7	90.2	**74.9**	92.0
Cost $\mathcal{J}$ [dB]	**38.2**	38.3	40.2	38.4
Variable error E [dB]	-19.3	**-19.4**	-15.8	-17.9

5. CONCLUSION

An NMF algorithm based on Bregman monotone operator splitting (B-MOS) was constructed for fast and stable convergence. As used in the conventional ADMM-based NMF, the cost function is reformulated from a bi-convex form to a linearly constrained convex minimization form, and we solve the dual problem. By applying B-MOS algorithms with an appropriate Bregman divergence design, the variable space metric is adaptively modified such that it matches the cost convexity and results in fast convergence without careful step-size selection. Results of several experiments demonstrated that the proposed B-MOS based algorithms improved the convergence rate, especially in the tail.

For future work, we will apply this method to practical applications such as source enhancement for audio and speech sources. We will analytically predict the convergence rate and use our method to explain the convergence rate differences in the tail.

6. APPENDIX

The primal-dual variable update procedure in the D-resolvent operator is given. The procedure (23) includes an iterative update of primal-dual variables $\{\mathbf{Z}, \mathbf{U}\}$ as seen in (17). Let us consider the problem associated with the convex conjugate function $G^\star$ as

$$\inf_{\mathbf{Z}}\ G(\mathbf{Y}\|\mathbf{Z}) + H_Z(\mathbf{Z}) \qquad \text{s.t. } \mathbf{Z} = \mathbf{0}. \tag{31}$$

This is equivalent to an update $\mathbf{Z}$ that minimizes the associated Lagrangian $\mathcal{Q}(\mathbf{Z}, \mathbf{U}) = G(\mathbf{Y}\|\mathbf{Z}) + H_Z(\mathbf{Z}) - \langle \mathbf{U}, \mathbf{Z}\rangle$. When the subgradient of $\mathcal{Q}$ includes zero, the set of variables $\{\mathbf{Z}, \mathbf{U}\}$ satisfies

$$\mathbf{0} \in T_1^{-1}(\mathbf{Z}) - \mathbf{U} \ \Leftrightarrow\ \mathbf{Z} \in T_1(\mathbf{U}), \tag{32}$$

where the inverse of T_1 corresponds to the $\partial\,(G(\mathbf{Y}\|\mathbf{Z}) + H_Z(\mathbf{Z}))$ $= T_1^{-1}(\mathbf{Z})$ is used (e.g., [22, 23]). For the input/output pair of the D-resolvent operator $\mathbf{U} \in R_1(\mathbf{X})$, (23) is reorganized by using (32):

$$(I + (\nabla D)^{-1} \circ T_1)(\mathbf{U}) \in \mathbf{X},$$
$$\mathbf{U} + (\nabla D)^{-1}(\mathbf{Z}) = \mathbf{X}, \quad \mathbf{0} \in T_1^{-1}(\mathbf{Z}) - \mathbf{U}. \tag{33}$$

By reorganizing (33), $\{\mathbf{Z}, \mathbf{U}\}$ are found to be associated with

$$\mathbf{0} \in T_1^{-1}(\mathbf{Z}) + (\nabla D)^{-1}(\mathbf{Z}) - \mathbf{X}. \tag{34}$$

Integral of (34) gives the $\mathbf{Z}$-update procedure as

$$\mathbf{Z} \leftarrow \arg\min_{\mathbf{Z}\geq\mathbf{0}} \Big(G(\mathbf{Y}\|\mathbf{Z}) + H_Z(\mathbf{Z}) - \langle \mathbf{X}, \mathbf{Z}\rangle + D^{\dagger}(\mathbf{Z})\Big). \tag{35}$$

From (33), the $\mathbf{U}$-update procedure using $\mathbf{Z}$ is given by

$$\mathbf{U} \leftarrow \mathbf{X} - (\nabla D)^{-1}(\mathbf{Z}). \tag{36}$$

Integration of procedures (36) and (24) removes a dual auxiliary variable $\mathbf{U}$ as

$$\mathbf{V} \leftarrow \mathbf{X} - 2(\nabla D)^{-1}(\mathbf{Z}). \tag{37}$$

For further simplified notation, we put additional dual auxiliary variables by $\mathbf{V} = (\nabla D)^{-1}(\tilde{\mathbf{V}})$, $\mathbf{X} = (\nabla D)^{-1}(\tilde{\mathbf{X}})$. Substituting these variables into (35) and (37) resulted in (27) and (28), respectively.

7. REFERENCES

[1] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788, 1999.

[2] 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.

[3] F. Sha and L. K. Saul, "Real-time pitch determination of one or more voices by nonnegative matrix factorization," in *Advances in Neural Information Processing Systems*, 2005, pp. 1233–1240.

[4] A. Dessein, A. Cont, and G. Lemaitre, "Real-time polyphonic music transcription with non-negative matrix factorization and beta-divergence," in *ISMIR-11th International Society for Music Information Retrieval Conference*, 2010, pp. 489–494.

[5] C. Joder, F. Weninger, F. Eyben, D. Virette, and B. Schuller, "Real-time speech separation by semi-supervised nonnegative matrix factorization," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2012, pp. 322–329.

[6] Z. Duan, G. J. Mysore, and P. Smaragdis, "Online PLCA for real-time semi-supervised source separation," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2012, pp. 34–41.

[7] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximations," *Computers and Mathematics with Applications*, vol. 2, pp. 17–40, 1976.

[8] D. L. Sun and C. Fevotte, "Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2014*. IEEE, 2014, pp. 6201–6205.

[9] J. Douglas and H. H. Rachford, "On the numerical solution of heat conduction problems in two and three space variables," *Transactions of the American mathematical Society*, vol. 82, no. 2, pp. 421–439, 1956.

[10] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1-3, pp. 293–318, 1992.

[11] D. W. Peaceman and H. H. Rachford, "The numerical solution of parabolic and elliptic differential equations," *Journal of the Society for industrial and Applied Mathematics*, vol. 3, no. 1, pp. 28–41, 1955.

[12] K. Niwa and W. B. Kleijn, "Bregman monotone operator splitting," *arXiv preprint arXiv:1807.04871*, 2018.

[13] L. M. Bregman, "The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming," *USSR computational mathematics and mathematical physics*, vol. 7, no. 3, pp. 200–217, 1967.

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge university press, 2004.

[15] A. Basu, I. R. Harris, N. L. Hjort, and MC. Jones, "Robust and efficient estimation by minimising a density power divergence," *Biometrika*, vol. 85, no. 3, pp. 549–559, 1998.

[16] S. Eguchi and Y. Kano, "Robustifing maximum likelihood estimation by psi-divergence," *ISM Research Memorandam*, vol. 802, 2001.

[17] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of machine learning research*, vol. 5, no. Nov, pp. 1457–1469, 2004.

[18] 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.

[19] A. Lefèvre, *Dictionary learning methods for single-channel source separation*, Ph.D. thesis, École normale supérieure de Cachan-ENS Cachan, 2012.

[20] W. Fenchel, "On conjugate convex functions," *Canad. J. Math*, vol. 1, no. 73–77, 1949.

[21] H. H. Bauschke, J. M. Borwein, and P. L. Combettes, "Bregman monotone optimization algorithms," *SIAM Journal on control and optimization*, vol. 42, no. 2, pp. 596–636, 2003.

[22] R. T. Rockafellar, *Convex Analysis*, Princeton university press, 1970.

[23] E. K. Ryu and S. Boyd, "Primer on monotone operator methods," *Appl. Comput. Math*, vol. 15, no. 1, pp. 3–43, 2016.