sensors

MDPI

Article

Block-Active ADMM to Minimize NMF with Bregman Divergences

Xinyao Li and Akhilesh Tyagi *

Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50010, USA; xli@iastate.edu
* Correspondence: tyagi@iastate.edu

Abstract: Over the last ten years, there has been a significant interest in employing *nonnegative matrix factorization* (NMF) to reduce dimensionality to enable a more efficient clustering analysis in machine learning. This technique has been applied in various image processing applications within the fields of computer vision and sensor-based systems. Many algorithms exist to solve the NMF problem. Among these algorithms, the *alternating direction method of multipliers* (ADMM) and its variants are one of the most popular methods used in practice. In this paper, we propose a block-active ADMM method to minimize the NMF problem with general Bregman divergences. The subproblems in the ADMM are solved iteratively by a *block-coordinate-descent-type* (BCD-type) method. In particular, each block is chosen directly based on the *stationary condition*. As a result, we are able to use much fewer auxiliary variables and the proposed algorithm converges faster than the previously proposed algorithms. From the theoretical point of view, the proposed algorithm is proved to converge to a stationary point sublinearly. We also conduct a series of numerical experiments to demonstrate the superiority of the proposed algorithm.

Keywords: NMF; ADMM; Bregman divergence; block active; imaging sensor

Citation: Li, X.; Tyagi, A. Block-Active ADMM to Minimize NMF with Bregman Divergences. *Sensors* **2023**, *23*, 7229. https://doi.org/10.3390/s23167229

Academic Editor: Loris Nanni

Received: 26 May 2023
Revised: 10 August 2023
Accepted: 16 August 2023
Published: 17 August 2023

1. Introduction

1.1. Overview of the Matrix Factorization Algorithms

Unsupervised learning is a form of machine learning in which models are trained on unlabeled data to classify patterns or make inferences without any external guidance or supervision. The key advantage of unsupervised learning is its ability to uncover hidden structures and relationships within datasets. This may not be readily apparent through manual inspection, enabling the automatic discovery of insights and patterns in large datasets. However, working with large datasets can be challenging because they are often noisy and high-dimensional, which makes their processing and analysis difficult. To address this challenge, researchers often use dimensionality reduction techniques to extract meaningful features from the data. By reducing the dimensionality of the data, the computation cost of training and classification algorithms can be improved. Dimensionality reduction can be achieved by reducing the size of the feature vector, which is the input data used to train and test machine learning models. Unsupervised learning methods factor the data matrix subject to various constraints for such dimensionality reduction. Depending on the constraints, the resulting factors have significantly different data representations. *Principal component analysis* (PCA) [1] enforces no constraints on the factorization. Consequently, PCA achieves an optimal low-dimension approximation to the data matrix while retaining as much of the original variation as possible. For this reason, PCA has been widely applied in various applications such as face recognition [2–4] and document representation [5–7].

In parallel, previous studies have shown that there is some psychological and physiological evidence for parts-based representation in the human brain [8–10]. *Nonnegative matrix factorization* (NMF) [8] has been proposed to learn the parts of objects by enforcing the nonnegative constraints. In particular, NMF approximates the data matrix by a

product of two nonnegative matrices. The nonnegative constraints are useful to learn a parts-based representation of the data because it only allows additive combinations. It has been shown nonnegative matrix factorization is superior to PCA in fields that use nonnegative datasets such as face recognition [11–14], document clustering [15,16], audio signal processing [17,18], and recommendation systems [19,20].

1.2. Nonnegative Matrix Factorization

Nonnegative Matrix Factorization (NMF) is a technique for factorizing a matrix into two nonnegative matrices, denoted as W and H. This method is distinct because it constrains all elements in W and H to be nonnegative. To grasp the concept of NMF, it is essential to comprehend the underlying intuition behind matrix factorization.

As shown in Figure 1, suppose we have a matrix V of size $m \times n$, where each element is greater than or equal to zero. With NMF, we can decompose V into two matrices: W of size $m \times k$ and H of size $k \times n$, where k is a chosen rank. Notably, both W and H have only nonnegative elements. Here, V is defined as:

$$V_{m \times n} = W_{m \times k} H_{k \times n} \tag{1}$$

where

- V is the original input matrix (Linear combination of W and H);
- W is the feature matrix;
- H is the coefficient matrix;
- k is the low-rank approximation of V ($k \leq \min(m, n)$).

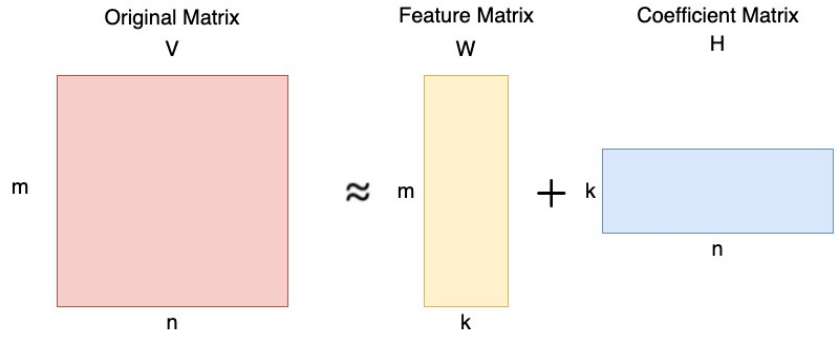

Figure 1. NMF intuition.

The primary goal of NMF is to perform dimensional reduction and feature extraction. By specifying a lower dimension k, the main objective of NMF is to identify two matrices, $W \in R_{m \times k}$ and $H \in R_{k \times n}$, containing only nonnegative elements, as illustrated in Figure 1.

Specifically, by setting $k \leq \min(m, n)$, the factorization process breaks down the original matrix V into two matrices. Therefore, the dimensional reduction occurs as the original matrix V of size $m \times n$ is represented as the product of a smaller matrix W of size $m \times k$ and a smaller matrix H of size $k \times n$, resulting in a dimensional reduction from $m \times n$ to $m \times k + k \times n$. Note that $(m \times k + k \times n) \leq (m \times n)$ since $k \leq \min(m, n)$. Typical machine learning algorithms, including statistical machine learning and deep learning methods such as convolutional neural networks (CNN), take a time superlinear in the input size. This training time and classification time also depend heavily on the feature space size. NMF likely reduces both the input size and the feature vector size. With the training time and classification time being superlinear, the efficiency of both improves significantly with NMF.

The underlying assumption of NMF is that the input comprises a set of latent features, each of which is represented by a column in the W matrix. Moreover, each column in the H matrix represents the "coordinates of a data point" in the W matrix, essentially holding the weights related to matrix W. In essence, each data point represented by a column in V can be approximated by a summation of nonnegative vectors represented by columns in W weighted by a row in H.

1.2.1. Image Processing—Facial Feature Extraction

In order to better understand the intuition behind the NMF algorithm, we consider real-world scenarios, specifically the application of the algorithm to image processing. Suppose we have an input image consisting of pixels that form matrix X. NMF produces two factors (W, H) such that each image $X(:, j)$ is approximated as a linear combination of the columns in W. As shown in Figure 2, for facial images, the columns of W can be interpreted as basic images consisting of features such as eyes, noses, mustaches, and lips. The columns of H indicate the presence of these features in the corresponding X image.

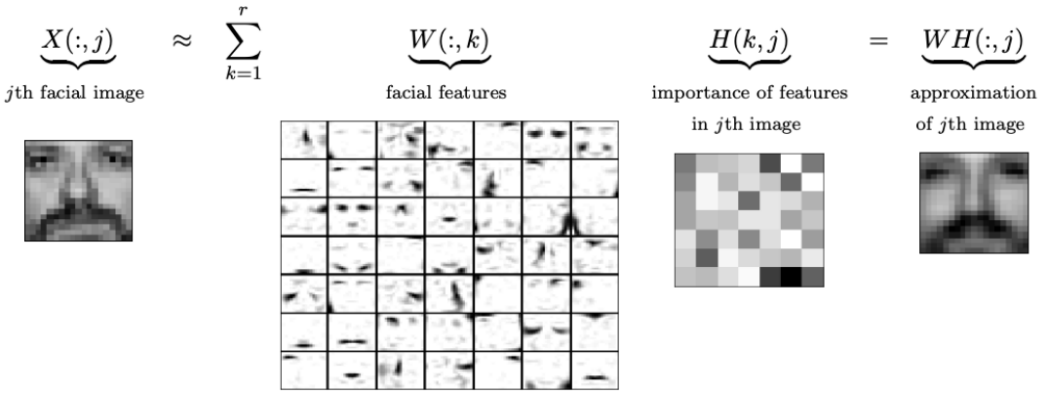

Figure 2. NMF face recognition [21].

1.2.2. Contributions

This paper makes the following innovative contributions revolving around a novel algorithm for tackling NMF challenges:

1. We present a coordinate descent approach coupled with an innovative strategy for selecting coordinates to address the ADMM subproblems.
2. In contrast to the classic ADMM and multiplicative update methods, our proposed algorithm attains a notably reduced error level while showcasing enhanced convergence characteristics, marked by an enhanced stability, smoother trajectories, and expedited convergence.
3. We establish the effectiveness of our approach through a rigorous theoretical analysis and substantiate our claims via an array of comprehensive experiments conducted on synthetic and real datasets in Section 6. These experiments collectively serve to underscore the superior performance and potential of our novel methodology.

1.2.3. Discussion

1. Comparing the proposed method in Algorithm 3 with the classical ADMM, we use much fewer primal and dual decision variables. Specifically, the ADMM in Algorithm 1 introduces new primal variables W_+ and H_+, and dual variables α_X, α_W, and α_H, while the proposed method in Algorithm 3 introduces no primal variables, and only one dual variable α_X. This helps with the efficiency of the algorithm.
2. In Section 4, we introduce a new approach termed the "block active method" designed to tackle the problems formulated in (14). Our central result, as established in Theorem 4, rigorously demonstrates that under reasonable assumptions, our proposed method converges towards a stationary point denoted as x^* in Equation (15) at a sublinear rate of convergence.

To expound on this, we demonstrate that the error, as defined by $f(x^k) - f(x^*)$ on the left-hand side of the equation within Theorem 4, consistently diminishes. This reduction is characterized by the relation $f(x^k) - f(x) = O(k^{-1})$, indicative of the error's gradual decline to zero with iteration count k approaching infinity. This type of convergence behavior is denoted as sublinear [22] due to its property of diminishing error reduction over iterations. This stands in contrast to the linear convergence typified by expressions such as γ^k for some constant $0 < \gamma < 1$, where the decline in error remains consistent.

3. NMF finds applications in tasks such as face recognition, document clustering, audio signal processing, and recommendation systems. When employing NMF to address analogous optimization problems, there should not be any difference in the theoretical results.
4. The image resolution may or may not affect the results. Given NMF is a nonconvex optimization problem, a global min cannot be guaranteed to be found in the general setup. The quality of the solution a method converges to depends on several factors, such as the initialization of W, H, and X, and the learning rate *rho*. Thus, improving the resolution of the data or quality of the data may or may not improve the result.

1.2.4. Paper Organization

This paper is organized as follows. Section 2 introduces the NMF problem and places it within the context of the related existing research. Section 3 provides a concise overview of the widely recognized ADMM. Section 4 presents the proposed *block-active method* as a coordinate descent approach coupled with an innovative strategy for selecting coordinates to address the ADMM subproblems. Section 5 introduces an innovative ADMM-style approach for addressing the NMF problem (2). Section 6 assesses the experimental outcomes across both synthetic and real datasets, while Section 7 serves as the concluding segment.

2. NMF Problem and Previous Work

The NMF problem can be formulated as follows:

$$\min f(W, H) \equiv D(V|WH), \tag{2}$$

$$\text{s.t. } W_{ik}, H_{kj} \geq 0, \tag{3}$$

where $D(V|\hat{V})$ represents some measure of divergence between V and its approximation $\hat{V}$. A general family of divergence functions is the β-divergence, denoted by D_β. The β-divergence between two matrices is defined as the sum of the elementwise divergence, i.e., $D_\beta(V, \hat{V}) = \sum_{i,j} d_\beta(V_{ij}|\hat{V}_{ij})$, where d_β is defined by

$$d_\beta(x|y) = \frac{x^\beta}{\beta(\beta-1)} + \frac{y^\beta}{\beta} - \frac{xy^{\beta-1}}{\beta-1}. \tag{4}$$

The three most commonly used β-divergence functions with NMF in practice are the Euclidean distance, Kullback–Leibler (KL) divergence, and Itakura–Saito (IS) divergence so as to model the Gaussian noise, Poisson noise, and multiplicative gamma noise, respectively. Particularly, we have

- $\beta = 2$ (Euclidean distance): $d(x|y) = \frac{1}{2}(x-y)^2$;
- $\beta = 1$ (Kullback–Leibler divergence): $d(x|y) = x \log \frac{x}{y} - x + y$;
- $\beta = 0$ (Itakura–Saito divergence): $d(x|y) = -\log \frac{x}{y} + \frac{x}{y} - 1$.

In the literature on NMF, many algorithms have been proposed to solve Problem (2) for $\beta = 2$, including *multiplicative updates* (MU) [23–26], projected gradient descent (PGD) [27], hierarchical alternating least square (HALS) [28–30], the alternating direction method of multipliers (ADMM) [31,32], and alternating nonnegative least square (ANLS) [33]. Unfortunately, there are few works proposed to solve the NMF problem with the general Bregman divergence. In this paper, we propose an ADMM that can be used to solve the

NMF problem with the general Bregman divergence. In fact, we are not the first one proposing an ADMM method to solve the NMF problem. For example, reference [31] also proposed an ADMM method to solve NMF problem and each subproblem had a closed-form solution. However, our method introduces a much fewer number of auxiliary variables, and we use an iterative method to solve each subproblem. By doing so, the proposed algorithm converges much faster than the previously proposed algorithms. We provide the theoretical analysis of the proposed algorithm and construct a line of numerical experiments to demonstrate the performance of the proposed method.

3. Alternating Direction Method of Multipliers

In this section, we provide a brief review of the well-known *alternating direction method of multipliers* (ADMM). We consider an optimization problem formulated as follows:

$$\min \ f(x) \quad \text{s.t.} \quad Ax = b \tag{5}$$

where $f : \mathbf{R}^n \to \mathbf{R}$ is the objective function, $x \in \mathbf{R}^n$ is the decision variable, $A \in \mathbf{R}^{m\times n}$ and $b \in \mathbf{R}^m$ are a given matrix and vector. Since the objective function f could be nonconvex, searching for a global solution is not easy. Instead, the common pursuit is to find a *stationary point* of the problem. A stationary point of (5) is a vector x^* that satisfies

$$\nabla f(x^*) + A^T y = 0 \tag{6}$$

$$Ax^* = b \tag{7}$$

The ADMM method can help us find such vector x^*. In particular, the *augmented Lagrangian function* is given by

$$L_\rho(x, y) = f(x) + \langle y, Ax - b\rangle + \frac{\rho}{2}\|Ax - b\|^2$$

where $y \in \mathbf{R}^m$ is the *Lagrangian multiplier*. Then, the ADMM method updates x and y by optimizing $L_\rho(x, y)$ alternatively. In particular, suppose we are given feasible vectors (x^k, y^k) at the kth iteration. Then,

$$x^{k+1} = \text{argmin}_x \ L_\rho(x, y) \tag{8}$$

$$y^{k+1} = y + \rho(Ax^{k+1} - b) \tag{9}$$

where x^{k+1} is a global minimizer of $L_\rho(x, y^k)$ for a fixed y^k, and y^{k+1} is the result of a one-step gradient ascent with step size ρ. Here the step size for y^{k+1} could be different from the penalty parameter ρ so that $y^{k+1} = y^k + \alpha(Ax^{k+1} - b)$ for some $\alpha \neq \rho$. However, it is common to use $\alpha = \rho$. Then, we obtain a sequence $\{x^k, y^k\}$ of vectors and [34] shows this sequence converges to a stationary point (x^*, y^*) that satisfies (3).

The paper [31] adapts the ADMM framework to solve the NMF problem with the Bregman divergence. They firstly reformulate the problem (2) by introducing additional auxiliary variables as follows:

$$\underset{W,H,X,W_+,H_+}{\text{minimize}} \ D(V|X) \tag{10}$$

$$\text{s.t.} \quad X = WH \tag{11}$$

$$W = W_+, H = H_+ \tag{12}$$

$$W_+ \geq 0, H_+ \geq 0 \tag{13}$$

where X, W_+, and H_+ are additional auxiliary variables. The corresponding augmented Lagrangian function of (2) is given by

$$
\begin{aligned}
L\rho(X, W, H, W_+, H_+, \alpha_X, \alpha_W, \alpha_H) = \\
& D(V, X) + \langle \alpha_X, X - WH \rangle + \frac{\rho}{2}\|X - WH\|_F^2 \\
& + \langle \alpha_W, W - W_+ \rangle + \frac{\rho}{2}\|W - W_+\|_F^2 \\
& + \langle \alpha_H, H - H_+ \rangle + \frac{\rho}{2}\|H - H_+\|_F^2
\end{aligned}
$$

where α_X, α_W, and α_H are the Lagrange multipliers. The updates are taken by minimizing L_ρ alternatively with respect to each primal variable and taking a gradient ascent in each of the Lagrange multipliers. The algorithm is summarized in Algorithm 1.

Algorithm 1: ADMM for NMF [31].

Initialize $X, W, H, W_+, H_+, \alpha_X, \alpha_W, \alpha_H$
for *iteration* $= 1, 2, \cdots$ **do**
$W^T \leftarrow (HH^T + I) \setminus (HX^T + W_+^T + \frac{1}{\rho}(H\alpha_X^T - \alpha_W^T))$
$H \leftarrow (W^T W + I) \setminus (W^T X + H_+ + \frac{1}{\rho}(W^T \alpha_X - \alpha_H))$
$X \leftarrow \text{argmin}_{X \geq 0} D(V|X) + \langle \alpha_X, X \rangle + \frac{\rho}{2}\|X - WH\|_F^2$
$W_+ \leftarrow \max\left\{0, W + \frac{1}{\rho}\alpha_W\right\}$
$H_+ \leftarrow \max\left\{0, H + \frac{1}{\rho}\alpha_H\right\}$
$\alpha_X \leftarrow \alpha_X + \rho(X - WH)$
$\alpha_H \leftarrow \alpha_H + \rho(H - H_+)$
$\alpha_W \leftarrow \alpha_W + \rho(W - W+)$
end

Here, the notation $A \setminus b$ is an operator in Matlab that takes the inverse of A and multiplies it by b, that is, $A \setminus b := A^{-1}b$.

4. Block-Active Method

In this section, we consider a constrained optimization problem formulated as follows:

$$\min \ f(x) \quad \text{s.t.} \quad x \geq 0, \tag{14}$$

where f is a convex objective function, and $x \in \mathbf{R}^n$ is the decision variable. Section 5 proposes a new ADMM-type method where the problem (14) is an important subproblem. We propose to use a *block coordinate descent* (BCD) method to solve the problem (14). In general, a BCD method picks up a block of coordinates of the decision variable and minimizes the objective function only with respect to the selected block of coordinates. In particular, let x^k be the current feasible point at the kth iteration. Let i_k be the selected coordinate. Then, the update rule is given by

$$x^{k+1} = \text{argmin} \ f(x + e_{i_k} v), \quad \text{s.t.} \quad x + e_{i_k} v \geq 0$$

where e_{i_k} is a vector whose entries are all zeros, except the i_kth entry is equal to 1. How to select the block is significant in a BCD method. In general, there are three ways to select the block, that is, cyclic, random, and greedy. In the cyclic selection rule, each block is selected cyclically. Each block is selected randomly if the random selection rule is applied. A block is selected if it has the largest magnitude of the partial derivative, that is,

$$i_k = \text{argmax}_{i\in[n]} \left| \frac{\partial f(x^k)}{\partial x_i} \right|$$

where $[n] = \{1, 2, \cdots, n\}$.

4.1. Block-Active Method

Here, we propose a new block coordinate method called *block-active method* to solve the problem (14) where the block is selected based on the *stationary condition*. Note that a vector x^* is a stationary point of (14) if it satisfies

$$\begin{cases} \frac{\partial f(x^*)}{\partial x_i} = 0, & \text{if } x_i^* > 0 \\ \frac{\partial f(x^*)}{\partial x_i} \geq 0, & \text{if } x_i^* = 0. \end{cases} \tag{15}$$

Here, Equation (15) is called *stationary condition*. At each iteration, we collect the coordinates that do not satisfy the stationary condition. In particular, let $x \geq 0$ be a feasible point. We construct an index set $\mathcal{F}$ as follows:

$$\mathcal{F} = \{i \in [n] : x_i > 0 \vee (x_i = 0 \wedge \partial f(x)/\partial x_i < 0)\}. \tag{16}$$

Note that here, we include some extra coordinates in $\mathcal{F}$, that is, $x_i > 0$ and $\frac{\partial f(x)}{\partial x_i} = 0$. Later on, we show that if x is already a stationary point, including these extra coordinates does not make the block-active method move away from a stationary point. Instead, if x is not a stationary point, with the help of a scale matrix H, including these extra coordinates make the proposed method converge faster.

Given the index set $\mathcal{F}$, we define vectors g and d as follows:

$$g_i := \frac{\partial f(x)}{\partial x_i}, \quad \forall i \in \mathcal{F} \quad \text{and} \quad d := -H^{-1}g \tag{17}$$

where $g \in \mathbf{R}^{|\mathcal{F}|}$, $d \in \mathbf{R}^{|\mathcal{F}|}$, and $H \in \mathbf{R}^{|\mathcal{F}|\times|\mathcal{F}|}$ is a *strictly positive definite (p.d.) matrix*. Given a scalar $\alpha > 0$, we define a single-variable function $x(\alpha)$ as follows:

$$x(\alpha)_i = \begin{cases} \max\{0, x_i + \alpha d_i\}, & i \in \mathcal{F}, \\ x_i, & i \notin \mathcal{F}. \end{cases} \tag{18}$$

From Equation (18), we can see only part of vector x is selected and updated. The sub-vector of x is selected based on the stationary condition (16). The algorithm is summarized in Algorithm 2. As noted by [35], our method has the potential to be extended in a distributed manner.

Algorithm 2: Block-active method to minimize (14)

Initialize x_0
for *iteration* $= 1, 2, \cdots$ **do**
  Select index set $\mathcal{F}_k$ based on the stationary condition (16)
  Compute g and d according to (17)
  Choose an appropriate step size α_k
  $x^{k+1} = x^k(\alpha_k)$ by (18)
end

4.2. Convergence Analysis of the BCD Method

Given a feasible point x, we can show x is a stationary point if and only if the single variable function $x(\alpha) = x$ for all strictly positive $\alpha > 0$. On the other hand, if x is not a

stationary point, then we can show there exists a strictly positive scalar $\overline{\alpha}$ for which any $\alpha \leq \overline{\alpha}$ causes a descent in the objective value.

Theorem 1. *(1) $x \geq 0$ is a stationary point of (14) if and only if $x = x(\alpha)$ for all $\alpha > 0$.*
(2) If x is not a stationary point, then there exists $\overline{\alpha} > 0$ such that

$$f(x(\alpha)) < f(x), \quad \text{for all } 0 < \alpha \leq \overline{\alpha}. \tag{19}$$

Definition 1. *A function g is called L-smooth if ∇g is Lipschitz continuous with constant $L > 0$. In particular, there exists a strictly positive scalar $L > 0$ for which*

$$\|\nabla g(x) - \nabla g(y)\| \leq L\|x - y\|, \quad \text{for all } x, y. \tag{20}$$

Given a function g is L-smooth, we have the following well-known descent lemma.

Lemma 1 (Descent Lemma)**.** *If g is L-smooth, then*

$$g(y) \leq g(x) + \langle \nabla g(x), y - x \rangle, \quad \text{for all } x, y. \tag{21}$$

Suppose the objective function f in (14) is L-smooth. It follows from the descent lemma that the sequence $\{f(x^k)\}$ generated by the *block-active method* consistently decreases and it converges to $f(x^*)$ sublinearly.

Theorem 2 (Convergence result)**.** *Assume the objective function f is L-smooth and $\lambda_{\min}\{Q^k\} \geq \mu$ for all k, where $\mu > 0$ is a fixed constant. If $\alpha_k \leq \min\{\overline{\alpha}^k, \mu^2/L\}$, where $\overline{\alpha}^k$ is defined in Theorem 1, then*

$$f(x^k) - f(x^*) \leq \frac{\|x^0 - x^*\|_Q^2}{k}, \tag{22}$$

where $\|z\|_H^2 := \sum_{i,j \in \mathcal{F}} z_i Q_{ij} z_j$.

In the above theorem, we demonstrate that the error terms defined as $f(x^k) - f(x^*)$ are consistently diminished. This reduction is characterized by the relation $f(x^k) - f(x^*) = \mathcal{O}(k^{-1})$, indicating the gradual decline to zero as the iteration count k approaches ∞. This type of convergence behavior is so-called sublinear [22]. It stands in contrast to the linear convergence rate in the form of γ^k for some constant $\gamma \in (0, 1)$, where the reduction is constant. On the other hand, the smoothness assumption of the objective function L can be dropped but the convergence can still be ensured by using the arguments introduced in [36,37] for the non-Lipschitz optimization. The proof of Theorem 1 and 2 can be found within Appendices A and B.

5. Block-Active ADMM

In this section, we propose a new ADMM-type method to solve the NMF problem (2) by using the *block-active method* to solve the subproblems. Particularly, since the intermediate quantity $X = WH$ needs to be updated repeatedly once the matrices W and H are updated, we directly introduce this quantity as a new variable in the optimization problem. Thus, the NMF problem becomes

$$\begin{aligned} &\min\ D(V|X), \\ &\text{s.t. } X = WH, \\ &\quad H, W \geq 0. \end{aligned} \tag{23}$$

Since the ADMM framework is good at dealing with equality constraints, we propose a new algorithm based on the ADMM framework by introducing one dual variable α_X. The corresponding augmented Lagrange function is given by

$$L_\rho(X, W, H, \alpha_X) = D(V|X) + \langle \alpha_X, X - WH \rangle + \frac{\rho}{2}||X - WH||_F^2. \tag{24}$$

The updates alternately optimize L_ρ with respect to each of the three primal variables, followed by one update on the dual variable. The updates are summarized as follows.

$$W^+ = \text{argmin}_{W \geq 0} L_\rho(X, W, H, \alpha_X) \tag{25}$$
$$H^+ = \text{argmin}_{H \geq 0} L_\rho(X, W^+, H, \alpha_X) \tag{26}$$
$$X^+ = \text{argmin} L_\rho(X, W^+, H^+, \alpha_X) \tag{27}$$
$$\alpha_X^+ = \alpha_X + \rho(X^+ - W^+ H^+) \tag{28}$$

Since the optimization with respect to X does not have any constraint, X^+ has a closed-form solution by solving the equation $\frac{\partial L_\rho}{\partial X} = 0$. The closed-form solution is given in ([31], Theorems 1 and 2). In contrast, the updates for W and H can be reformulated in the form of *nonnegative least squares*. Taking the optimization of H as an example, we have

$$\begin{aligned} H^+ &= \text{argmin}_{H \geq 0} L_\rho(X, W^+, H, \alpha_X) \\ &= \text{argmin}_{H \geq 0} \langle \alpha_X, X - WH \rangle + \frac{\rho}{2}||X - WH||_F^2 \\ &= \text{argmin}_{H \geq 0} \frac{\rho}{2} \left[\frac{1}{\rho^2} \|\alpha_X\|^2 + 2\langle \alpha_X/\rho, X - WH \rangle + ||X - WH||_F^2 \right] \\ &= \text{argmin}_{H \geq 0} \frac{\rho}{2} ||X - WH + \alpha_X/\rho||_F^2 \\ &= \text{argmin}_{H \geq 0} ||(X + \alpha_X/\rho) - WH||_F^2 \end{aligned}$$

Thus, this subproblem can be solved by the method we proposed in the previous section, called *block-active method*. In particular, we can choose the scaling matrix Q as part of $W^T W$ based on the index set $\mathcal{F}$. Since $W \in \mathbf{R}^{M \times K}$ and $M \gg K$, $W^T W$ is highly likely strictly positive definite so that Q is a submatrix of $W^T W$. Moreover, we denote $nnls\_blockactive(A, B)$ as the procedure proposed in the previous section and used to solve the nonnegative constraint problem in the form of

$$\min \|AX - B\|^2 \quad \text{s.t. } X \geq 0. \tag{29}$$

Algorithm 3 is provided as an example of the proposed block-active ADMM method for the case where $\beta = 1$ in the β-divergence distance.

Algorithm 3: Block active ADMM.

Inputs V
Initialize X, W, H, α_X
for $iteration = 1, 2, \ldots$ **do**
  $W^T = nnls\_blockactive(H^T, (X + \alpha_X/\rho)^T)$
  $H = nnls\_blockactive(W, X + \alpha_X/\rho)$
  $X = \frac{\rho WH - \alpha - 1 + \sqrt{(1+\alpha-\rho WH)^2 + 4\rho V}}{2\rho}$
  $\alpha_X = \alpha_X + \rho(X - WH)$
end

Remark 1. *As established in ([32], Theorem 2), the alternating direction method of multipliers (ADMM) demonstrates convergence to a stationary point, when descents are achieved for subproblems within each iteration, despite the global problem being nonconvex. In our context, we are*

addressing a nonconvex optimization challenge, specifically the nonnegative matrix factorization (NMF) problem as defined in Equation (2) and subsequently reformulated in Equation (23). Through the utilization of the ADMM framework, we strategically partition the problem into a series of subproblems, each solvable within an iteration. Leveraging the convergence assurance provided by Theorem 2, we can confidently assert that a descent is guaranteed within each subproblem. Consequently, invoking the findings of ([32], Theorem 2) within our specific context, we secure a robust convergence result for the proposed method delineated in Algorithm 3, leading to the attainment of a stationary point.

6. Numerical Experiments

6.1. Synthetic Datasets

We first tested the proposed algorithm on a moderately synthetic dataset with $m = 500$, $n = 500$, and $k = 150$. We generated the ground truth W_0 and H_0, and $V = W_0H_0$. We examined the performance of the proposed algorithm against the standard multiplicative update [38] and the ADMM [34]. We set $\rho = 1$ and the maximum iteration to be 1000. The performance results are shown in Figure 3. We can see that the proposed block method can achieve a much lower error level given the same amount of time.

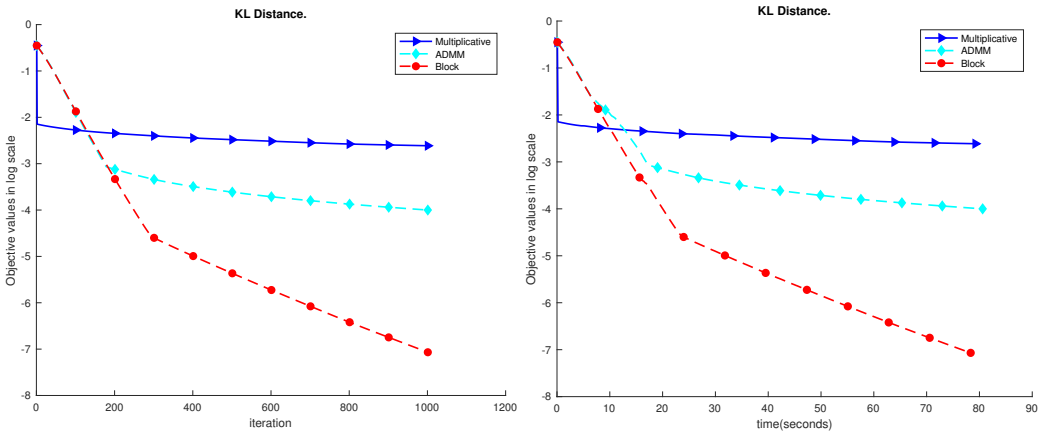

Figure 3. Performance comparison of the synthetic dataset. Here, we set $m = n = 500$ and $k = 150$. The maximum iteration is set to be 1000. We record the objective value on the log scale. We can see the proposed block method can achieve a much lower error level given the same amount of running time. In another word, the block method is faster to achieve the specified accuracy than the other two methods.

6.2. Real Datasets

We evaluated the proposed method against both the multiplicative update and ADMM algorithms using real datasets. These datasets were generated using either a 2D imaging sensor or a near-infrared (NIR) imaging sensor.

1. *UMist* (https://cs.nyu.edu/~roweis/data.html, accessed on 2 January 2022): This dataset is an image dataset containing 575 images of 20 people, which consist of images of individuals captured in various poses, ranging from profile to frontal views. All files in the dataset are in the PGM format, have a resolution of approximately 220×220 pixels, and are 256-bit grayscale images.
2. *ORL* (http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html, accessed on 2 January 2022): The dataset was generated by a 2D imaging sensor and includes 400 different images of each of 40 distinct individuals, where each image has 92×112 pixels and a depth of 256 levels of gray per pixel. The photographs were taken on different occasions, with variations in lighting, facial expressions, and facial features.
3. *COIL* (http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html, accessed on 2 January 2022): The dataset contains 7200 images in the form of 32×32 pixels

for 100 objects. The images were captured on a motorized turntable against a black background. The dataset was utilized in a real-time recognition system that employed a sensor to detect the objects and display their angular pose.

4. *YaleB* (http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html, accessed on 10 February 2023): The dataset consists of image data generated by a 2D imaging sensor. It comprises 2414 images of size 192 × 168 pixels from 38 individuals. The images were taken under different lighting conditions and a variety of facial expressions.
5. *NIR* (http://vcipl-okstate.org/pbvs/bench/Data/07/download.html, accessed on 10 February 2023): The dataset was created via a near-infrared (NIR) imaging sensor. It includes 3940 NIR face images of 197 persons. The images have a size of 480 × 640 pixels, 8-bit, and are not compressed.

The results are shown in Figures 4–8. Based on the results on the real datasets, we can see the objective value using multiplicative update decreases faster at the beginning, but later, the proposed block method can converge to a better solution which has a much lower error level. In addition, comparing to the ADMM, the proposed block method is much more stable. In Figure 7, the objective value using the ADMM does not consistently decrease. That is because $\rho = 1$ is too small for the YaleB dataset. However, using $\rho = 1$, the proposed block method does not diverge, and the objective value continuously decreases. The source code for the proposed algorithms has been added to GitHub, accessible at https://github.com/Xinyao90/Block-active-ADMM-to-Minimize-NMF-with-Bregman-Divergences.git, accessed on 10 August 2023.

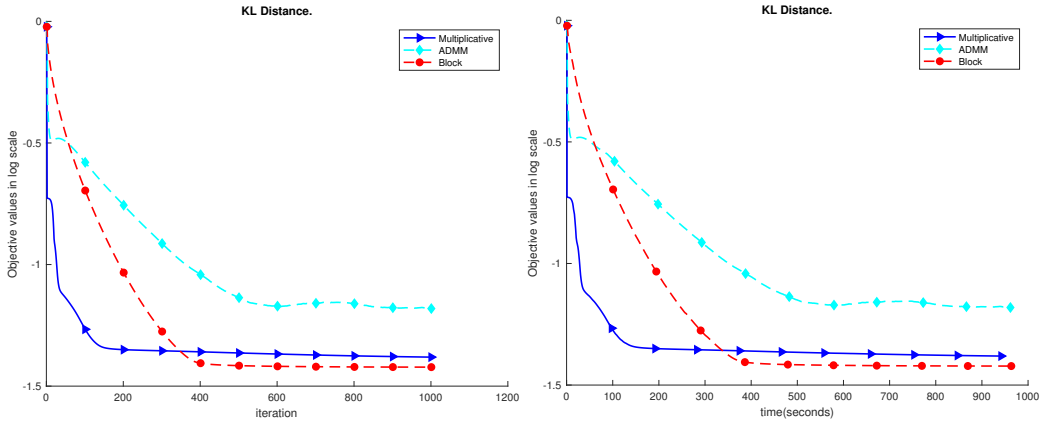

Figure 4. Performance comparison of the UMist dataset.

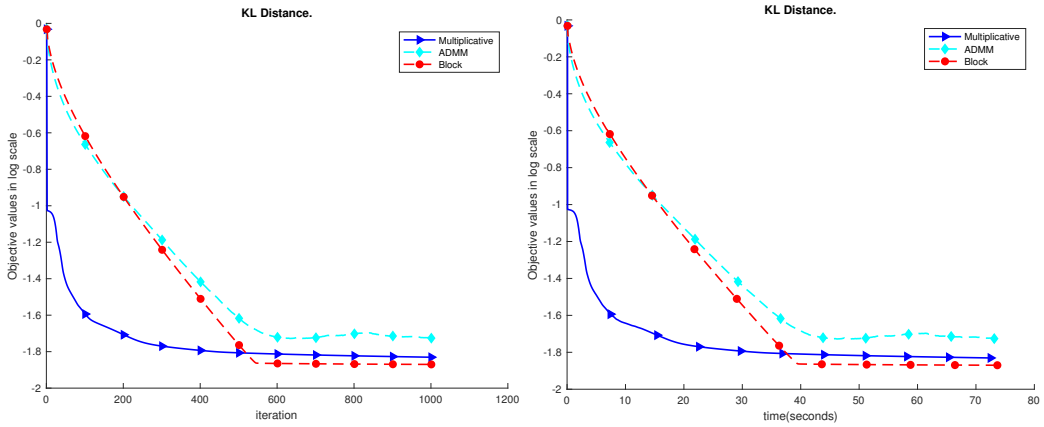

Figure 5. Performance comparison of the ORL dataset.

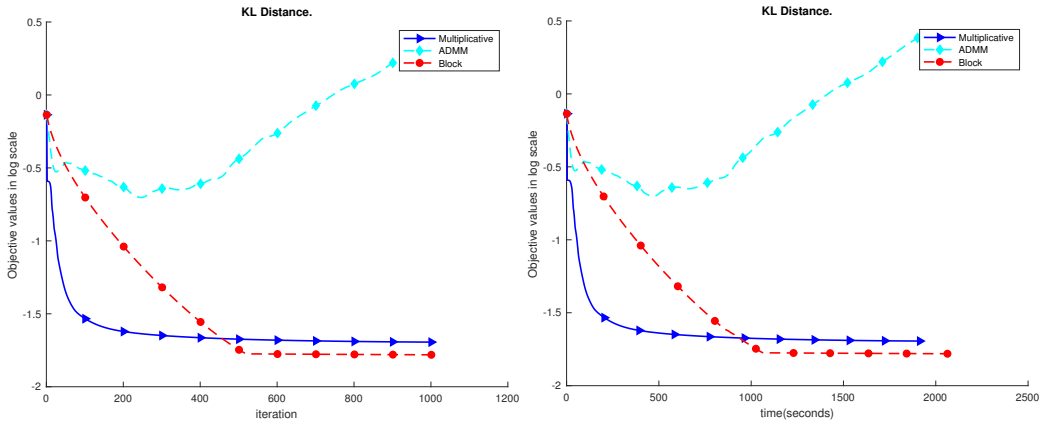

Figure 6. Performance comparison of the COIL dataset.

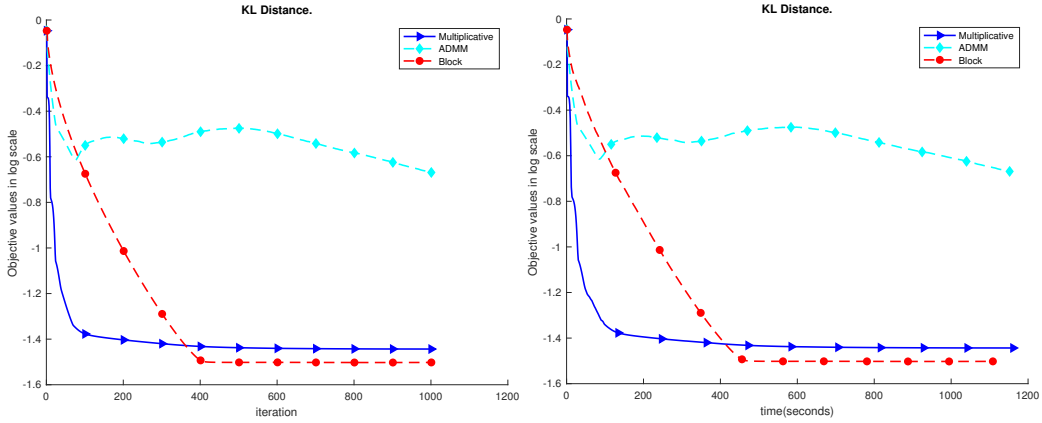

Figure 7. Performance comparison of the YaleB dataset.

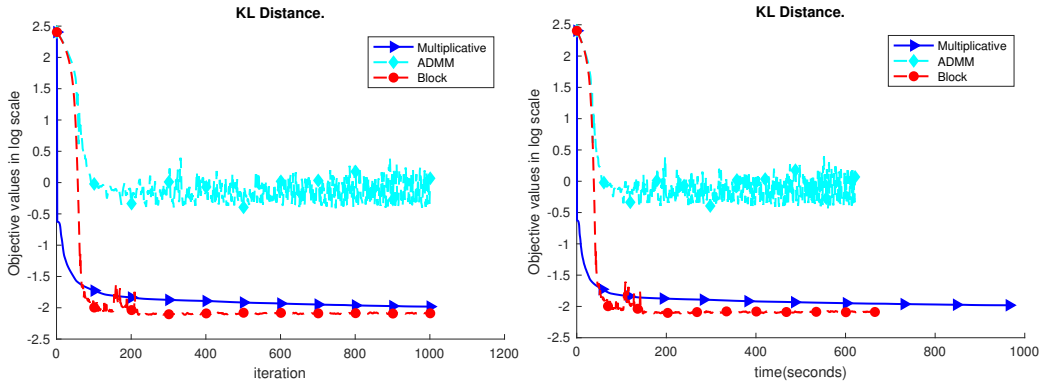

Figure 8. Performance comparison of the NIR dataset.

7. Conclusions

In this paper, we proposed a new block method that aimed to solve the nonnegative matrix factorization problem using the general Bregman divergence distance metric. Nonnegative matrix factorization is a widely used technique in various fields such as image processing, speech analysis, and bioinformatics. In particular, image processing systems heavily rely on databases generated by existing imaging sensors, and the efficiency and accuracy of these systems depend on the performance of the nonnegative matrix factorization method used to process these databases.

Our proposed block method was built on the framework of the alternating direction method of multipliers (ADMM), which is a popular algorithm used to solve optimization problems. However, instead of following the traditional approach of the ADMM to solve the subproblems, we introduced a new method that employed a block coordinate method. In this approach, we selected a block based on the stationary condition, which allowed

us to converge faster and to a solution with a lower error level compared to the previous ADMM method.

To demonstrate the effectiveness of our proposed method, we conducted a series of numerical experiments. The experiments included comparisons of our block method with the traditional ADMM method and other state-of-the-art methods in terms of runtime and overall accuracy. Our numerical results showed the dominance of our proposed block method over other methods, highlighting its effectiveness in solving the nonnegative matrix factorization problem using the general Bregman divergence distance metric.

In summary, our proposed block method provides an efficient and accurate solution to the nonnegative matrix factorization problem using the general Bregman divergence distance metric. Its unique approach to solving subproblems using a block coordinate method has proven to be faster and more accurate than traditional methods, as demonstrated by our numerical experiments. Our proposed method can help improve the performance of image processing systems and other applications that rely on nonnegative matrix factorization.

Author Contributions: Conceptualization, X.L.; Methodology, X.L.; Formal analysis, X.L.; Writing—original draft, X.L.; Writing—review & editing, A.T.; Project administration, A.T.; Funding acquisition, A.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable

Conflicts of Interest: The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A. Proof of Theorem 1

Proof. (i) $\Longrightarrow$ Since x is a stationary point, then $\mathcal{F} = \{i \in [n] : x_i > 0\}$. For each $i \in \mathcal{F}$, we have $x_i > 0$ and $\partial f/\partial x_i = 0$, so that $x_i(\alpha) = \max\{0, x_i\} = x_i$ for all $\alpha > 0$.

$\Longleftarrow$ Suppose $x(\alpha)_i = x_i$ for all $i \in [n]$. Then,

$$\begin{cases} d_i = 0, & x_i > 0 \\ d_i \leq 0, & x_i = 0 \end{cases}$$

By the definition of $\mathcal{F}$, we have

(a) If $x_i > 0$, then $d_i = 0$, so that $\frac{\partial f}{\partial x_i} d_i = 0$.

(b) If $x_i = 0$, then $\frac{\partial f}{\partial x_i} < 0$ and $d_i \leq 0$, so that $\frac{\partial f}{\partial x_i} d_i \geq 0$.

Therefore, we have $\frac{\partial f}{\partial x_i} d_i \geq 0$ for all $i \in \mathcal{F}$ and so

$$\sum_{i \in \mathcal{F}} \frac{\partial f}{\partial x_i} d_i \geq 0.$$

On the other hand, since H is a positive definite matrix, then the definition of d implies that

$$\sum_{i \in \mathcal{F}} \frac{\partial f}{\partial x_i} d_i = -g^T H^{-1} g \leq 0.$$

Therefore, we have $\sum_{i\in\mathcal{F}} \frac{\partial f}{\partial x_i} d_i = 0$. Moreover, from (a)–(b), we know $\frac{\partial f}{\partial x_i} d_i = 0$ for all $i \in \mathcal{F}$. If $d_i = 0$ for all $i \in \mathcal{F}$, then $\frac{\partial f}{\partial x_i} = 0$ so that x is a stationary point. If $d_i < 0$ for $i \in \mathcal{F}$ and $x_i = 0$, then $\frac{\partial f}{\partial x_i} d_i = 0$ implies $\frac{\partial f}{\partial x_i} = 0$, so that x is a stationary point.

(ii) Suppose x is not a stationary point. Consider two index sets

$$W_1 = \{i \in \mathcal{F} : (x_i > 0 \wedge d_i \neq 0) \vee (x_i = 0 \wedge d_i > 0)\} \tag{A1}$$

$$W_2 = \{i \in \mathcal{F} : (x_i > 0 \wedge d_i = 0) \vee (x_i = 0 \wedge d_i \leq 0)\} \tag{A2}$$

Then, $\mathcal{F} = W_1 \uplus W_2$. Moreover, if $i \in W_2$, then $x(\alpha)_i = x_i$ for all $\alpha > 0$. Since x is not a stationary point, then W_1 is nonempty.
Let $i \in W_1$. If $x_i = 0$ and $d_i > 0$, then

$$x_i(\alpha) = \max\{0, x_i + \alpha d_i\} = x_i + \alpha d_i, \quad \text{for all } \alpha > 0.$$

If $x_i > 0$ and $d_i \neq 0$, then define

$$\overline{\alpha} = \sup\{\alpha : x_i + \alpha d_i \geq 0, \forall i \in W_1\}.$$

Note that here, $\overline{\alpha}$ is either ∞ or a positive number. Then, we define the direction $\overline{d} \in \mathbf{R}^{|\mathcal{F}|}$ as follows:

$$\overline{d}_i = \begin{cases} d_i, & i \in W_1 \\ 0, & i \in W_2 \end{cases} \tag{A3}$$

Therefore, for all $0 < \alpha \leq \overline{\alpha}$, we have

$$x(\alpha)_i = x_i + \alpha \overline{d}_i, \quad \text{for all } i \in \mathcal{F}.$$

Moreover, we have

$$\sum_{i\in W_2} \frac{\partial f}{\partial x_i} d_i \geq 0$$

so that

$$\sum_{i\in W_1} \frac{\partial f}{\partial x_i} \overline{d}_i = \sum_{i\in W_1} \frac{\partial f}{\partial x_i} d_i \leq \sum_{i\in\mathcal{F}} \frac{\partial f}{\partial x_i} d_i = -g^T H^{-1} g < 0.$$

As a result, $\overline{d}$ is a feasible descent direction, so that for any $\alpha \leq \overline{\alpha}$, we have $f(x(\alpha)) < f(x)$.
□

Appendix B. Proof of Theorem 2

Proof. Let x be the feasible point at the kth iteration. Let $G(x) = \frac{x - x^+}{\alpha}$. Then, the descent lemma implies that

$$\begin{aligned} f(x^+) &\leq f(x) + \langle \nabla f(x), x^+ - x \rangle + \frac{L}{2}\|x^+ - x\|^2 \\ &\overset{(a)}{\leq} f(z) + \langle \nabla f(x), x^+ - z \rangle + \frac{L}{2}\|x^+ - x\|^2 \end{aligned}$$

where (a) is due to the convexity of f. The update can be considered as the following optimization

$$\begin{aligned} x^+ =& \text{argmin } f(x) + \langle \nabla f(x), z - x \rangle + h(z) + \frac{1}{2\alpha}\|z - x\|_H^2 \\ =& \text{argmin } h(z) + \frac{1}{2\alpha}\|z - (x - \alpha H^{-1}\nabla f(x))\|_H^2 \end{aligned}$$

where $h(x) = \chi_+(x)$. Therefore, we have

$$\frac{1}{\alpha}H(x^+ - x) + \nabla f(x) + \partial h(x^+) \ni 0.$$

so that we have

$$\begin{aligned} h(z) \geq& h(x^+) + \langle -\frac{1}{\alpha}H(x^+ - x) - \nabla f(x), z - x^+ \rangle \\ =& h(x^+) + \langle HG(x) - \nabla f(x), z - x^+ \rangle \end{aligned}$$

which further implies

$$0 \geq \langle HG(x) - \nabla f(x), z - x^+ \rangle$$

Consequently, we obtain

$$\begin{aligned} f(x^+) \leq& f(z) + \langle \nabla f(x), x^+ - z \rangle + \frac{L}{2}\|x^+ - x\|^2 \\ \leq& f(z) + \langle HG(x), x^+ - z \rangle + \frac{L\alpha^2}{2}\|G(x)\|^2 \\ =& f(z) + \langle G(x), x^+ - z \rangle_H + \frac{L\alpha^2}{2}\|G(x)\|^2 \\ =& f(z) + \frac{1}{2\alpha}\Big(\|x^+ + \alpha G(x) - z\|_H^2 - \alpha^2\|G(x)\|_H^2 - \|x^+ - z\|_H^2\Big) + \frac{L\alpha^2}{2}\|G(x)\|^2 \\ =& f(z) + \frac{1}{2\alpha}\Big(\|x - z\|_H^2 - \|x^+ - z\|_H^2\Big) - \frac{\alpha}{2}\|G(x)\|_H^2 + \frac{L\alpha^2}{2}\|G(x)\|^2 \\ \overset{(b)}{\leq}& f(z) + \frac{1}{2\alpha}\Big(\|x - z\|_H^2 - \|x^+ - z\|_H^2\Big) - \left(\frac{\alpha\mu^2}{2} - \frac{L\alpha^2}{2}\right)\|G(x)\|^2 \\ \overset{(c)}{\leq}& f(z) + \frac{1}{2\alpha}\Big(\|x - z\|_H^2 - \|x^+ - z\|_H^2\Big) \end{aligned}$$

where (b) is because $\lambda_{\min}\{H\} \geq \mu$, and (c) is due to $\alpha \leq \frac{\mu^2}{L}$.

Now let $z = x$; then, we have

$$f(x^+) \leq f(x) - \frac{1}{2\alpha}\|x^+ - x\|_H^2$$

so that the objective f is a descent. Furthermore, let $z = x^*$; then, we have

$$\sum_{i=1}^{k} f(x^i) - f(x^*) \leq \sum_{i=1}^{k}\Big(\|x^{i-1} - x^*\|_H^2 - \|x^i - x^*\|_H^2\Big) \leq \|x^0 - x^*\|_H^2$$

Since f is a descent, we have

$$k(f(x^k) - f(x^*)) \leq \sum_{i=1}^{k} f(x^i) - f(x^*) \leq \|x^0 - x^*\|_H^2$$

□

References

1. Maćkiewicz, A.; Ratajczak, W. Principal components analysis (PCA). *Comput. Geosci.* **1993**, *19*, 303–342. [CrossRef]
2. Gottumukkal, R.; Asari, V.K. An improved face recognition technique based on modular PCA approach. *Pattern Recognit. Lett.* **2004**, *25*, 429–436. [CrossRef]
3. Moon, H.; Phillips, P.J. Computational and performance aspects of PCA-based face-recognition algorithms. *Perception* **2001**, *30*, 303–321. [CrossRef] [PubMed]
4. Perlibakas, V. Distance measures for PCA-based face recognition. *Pattern Recognit. Lett.* **2004**, *25*, 711–724. [CrossRef]
5. Platt, J.C.; Toutanova, K.; Yih, W.T. Translingual document representations from discriminative projections. In Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Cambridge, MA, USA, 9–11 October 2010; pp. 251–261.
6. Gomez, J.C.; Moens, M.F. PCA document reconstruction for email classification. *Comput. Stat. Data Anal.* **2012**, *56*, 741–751. [CrossRef]
7. He, X.; Cai, D.; Liu, H.; Ma, W.Y. Locality preserving indexing for document representation. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, 25–29 July 2004; pp. 96–103.
8. Lee, D.D.; Seung, H.S. Learning the parts of objects by non-negative matrix factorization. *Nature* **1999**, *401*, 788–791. [CrossRef] [PubMed]
9. Cai, D.; He, X.; Han, J.; Huang, T.S. Graph regularized nonnegative matrix factorization for data representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *33*, 1548–1560. [PubMed]
10. Cai, D.; He, X.; Wang, X.; Bao, H.; Han, J. Locality preserving nonnegative matrix factorization. In Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence, Pasadena, CA, USA, 11–17 July 2009.
11. Wang, Y.; Jia, Y.; Hu, C.; Turk, M. Non-negative matrix factorization framework for face recognition. *Int. J. Pattern Recognit. Artif. Intell.* **2005**, *19*, 495–511. [CrossRef]
12. Guillamet, D.; Vitria, J. Non-negative matrix factorization for face recognition. In Proceedings of the Topics in Artificial Intelligence: 5th Catalonian Conference on AI, CCIA 2002, Castellón, Spain, 24–25 October 2002; pp. 336–344.
13. Rajapakse, M.; Wyse, L. NMF vs. ICA for face recognition. In Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis, 2003, ISPA 2003, Rome, Italy, 18–20 September 2003; Volume 2, pp. 605–610.
14. Chen, W.S.; Pan, B.; Fang, B.; Li, M.; Tang, J. Incremental nonnegative matrix factorization for face recognition. *Math. Probl. Eng.* **2008**, *2008*, 410674. [CrossRef]
15. Allab, K.; Labiod, L.; Nadif, M. A semi-NMF-PCA unified framework for data clustering. *IEEE Trans. Knowl. Data Eng.* **2016**, *29*, 2–16. [CrossRef]
16. Gaussier, E.; Goutte, C. Relation between PLSA and NMF and implications. In Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Salvador, Brazil, 15–19 August 2005; pp. 601–602.
17. Hassan, N.; Ramli, D.A. A comparative study of blind source separation for bioacoustics sounds based on FastICA, PCA and NMF. *Procedia Comput. Sci.* **2018**, *126*, 363–372. [CrossRef]
18. Févotte, C.; Vincent, E.; Ozerov, A. Single-channel audio source separation with NMF: Divergences, constraints and algorithms. In *Audio Source Separation*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 1–24.
19. Javed, M.A.; Younis, M.S.; Latif, S.; Qadir, J.; Baig, A. Community detection in networks: A multidisciplinary review. *J. Netw. Comput. Appl.* **2018**, *108*, 87–111. [CrossRef]
20. Gao, T.; Olofsson, S.; Lu, S. Minimum-volume-regularized weighted symmetric nonnegative matrix factorization for clustering. In Proceedings of the 2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Washington, DC, USA, 7–9 December 2016; pp. 247–251.
21. Gillis, N. The why and how of nonnegative matrix factorization. In *Regularization, Optimization, Kernels, and Support Vector Machines*; Chapman & Hall: London, UK, 2014.
22. Bertsekas, D.P. Nonlinear programming. *J. Oper. Res. Soc.* **1997**, *48*, 334. [CrossRef]
23. Lee, D.D.; Seung, H.S. Algorithms for non-negative matrix factorization. In Proceedings of the NIPS 2001 Conference (Advances in Neural Information Processing Systems 14), Vancouver, BC, Canada, 3–8 December 2001; pp. 556–562.
24. Févotte, C.; Idier, J. Algorithms for nonnegative matrix factorization with the β-divergence. *Neural Comput.* **2011**, *23*, 2421–2456. [CrossRef]
25. Sra, S.; Dhillon, I.S. Generalized nonnegative matrix approximations with Bregman divergences. In Proceedings of the NIPS 2005 Conference (Advances in Neural Information Processing Systems 18 (NIPS 2005), Vancouver, BC, Canada December 5–8 2005; pp. 283–290.
26. Yang, Z.; Oja, E. Unified development of multiplicative algorithms for linear and quadratic nonnegative matrix factorization. *IEEE Trans. Neural Netw.* **2011**, *22*, 1878–1891. [CrossRef] [PubMed]
27. Lin, C.J. Projected gradient methods for nonnegative matrix factorization. *Neural Comput.* **2007**, *19*, 2756–2779. [CrossRef] [PubMed]
28. Cichocki, A.; Zdunek, R.; Amari, S.I. Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization. In Proceedings of the International Conference on Independent Component Analysis and Signal Separation, London, UK, 9–12 September 2007; pp. 169–176.

29. Cichocki, A.; Anh-Huy, P. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **2009**, *E92-A*, 708–721.
30. Hsieh, C.J.; Dhillon, I.S. Fast coordinate descent methods with variable selection for non-negative matrix factorization. In Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, 21–24 August 2011; pp. 1064–1072.
31. Sun, D.L.; Fevotte, C. Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence. In Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 6201–6205.
32. Hong, M.; Luo, Z.Q.; Razaviyayn, M. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM J. Optim.* **2016**, *26*, 337–364. [CrossRef]
33. Kim, J.; Park, H. Fast nonnegative matrix factorization: An active-set-like method and comparisons. *SIAM J. Sci. Comput.* **2011**, *33*, 3261–3281. [CrossRef]
34. Boyd, S.; Parikh, N.; Chu, E. *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*; Now Publishers Inc.: Norwell, MA, USA, 2011.
35. Gao, T.; Chu, C. Did: Distributed incremental block coordinate descent for nonnegative matrix factorization. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
36. Gao, T.; Lu, S.; Liu, J.; Chu, C. On the Convergence of Randomized Bregman Coordinate Descent for Non-Lipschitz Composite Problems. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 5549–5553.
37. Gao, T.; Lu, S.; Liu, J.; Chu, C. Randomized bregman coordinate descent methods for non-lipschitz optimization. *arXiv* **2020**, arXiv:2001.05202.
38. Lin, C.J. On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *IEEE Trans. Neural Netw.* **2007**, *18*, 1589–1596.